Utilizing, Producing, and Contributing Design Knowledge in DSR Projects

Andreas Drechsler[1] (✉) (ORCID 0000-0002-7026-0241), Alan R. Hevner[2] (ORCID 0000-0003-4953-3900)

[1]Victoria University of Wellington, Wellington, New Zealand
andreas.drechsler@vuw.ac.nz
[2]University of South Florida, Tampa, USA
ahevner@usf.edu

Abstract. We distinguish several design knowledge types in IS research and examine different modes of utilizing and contributing design knowledge that can take place during design science research (DSR) projects. DSR projects produce *project design knowledge*, which is project-specific, possibly untested, conjectural, and temporary; thus, distinct from the more stable contributions to the propositional and prescriptive human knowledge bases. We also identify *solution design knowledge* as distinct from solution design entities in the prescriptive knowledge base. Each of the six modes of utilizing or contributing knowledge (i.e. *design theorizing modes*) we examine draws on different knowledge types in a different way to inform the production of project design knowledge (including artifact design) in a DSR project or to grow the human knowledge bases in return. Design science researchers can draw on our design theorizing modes and design knowledge perspectives to utilize the different extant knowledge types more consciously and explicitly to inform their build and evaluation activities, and to better identify and explicate their research's contribution potential to the human knowledge bases.

Keywords: design knowledge, design theorizing, knowledge bases, knowledge contribution, knowledge for action, knowledge for entity realization

1 Introduction

Two major genres of inquiry in the IS discipline contribute to knowledge growth [9, 23]: science-oriented research activities primarily grow propositional knowledge or Ω-knowledge (comprising descriptive and explanatory knowledge), while design-oriented research activities primarily grow applicable (or prescriptive) knowledge or λ-knowledge. Contributions to λ typically comprise knowledge about technological innovations that are or can be useful for individuals, organizations, or society – and also to develop future technological innovations. Contributions to Ω enhance our understanding of the world and the phenomena our technologies harness (or cause). Research projects may combine both genres of inquiry and contribute to both knowledge bases.

Many IS literature sources highlight the importance of utilizing the knowledge in both knowledge bases together in design-oriented IS research projects [12, 23, 39, 42].

adfa, p. 1, 2011.

However, these sources usually do not cover the particular ways of doing so in greater detail. This not only leads to challenges for the researchers involved in DSR projects to ground their design decisions explicitly on extant knowledge and transparently document the decisions and their justification. It also challenges these researchers to systematically articulate contributions to Ω and λ in ways that allow a cumulative accumulation and evolution in both knowledge bases.

As a first remedy for this issue, we illustrate how researchers in DSR projects can grow and utilize all knowledge types through different modes of design theorizing. We also distinguish two distinct design knowledge types in the process, which allows us to provide a more comprehensive perspective on knowledge utilization, production, and contribution in DSR. We integrate unconnected positions in the IS literature concerning design knowledge and related concepts, as well as approaches to grow both knowledge bases. We employ the term *design theorizing* for all activities utilizing or contributing knowledge in the context of design. In doing so, we follow Weick's [55, 56] distinction between theory and theorizing. He particularly emphasizes the important roles of pre-theoretical knowledge in the process towards developing a more fully articulate (explanatory) theory and we do the same to illustrate how all forms of knowledge can contribute to informing artifact realization processes or vice-versa.

2 Distinguishing Project Design Knowledge and Solution Design Knowledge

The first distinction we wish to make concerns (1) design knowledge that is produced and remains within a single DSR project, and (2) design knowledge that is part of the λ-knowledge base. We call the former *project design knowledge* and the latter *solution design knowledge* in the remainder of the paper.

Within a DSR project, researchers draw on the existing knowledge bases as well as other sources (their own and others' experience or creativity, for instance) to produce a plethora of temporary, tentative, and highly project-specific design knowledge. Such knowledge comprises knowledge regarding the project's problem space (including the specific (class of) contexts, the problem diagnosis, and the related goodness criteria for the resulting artifact) and the solution space (including the (meta-)artifacts or artifact components as the actual solution entities, but also the corresponding search criteria, or the build and evaluation activities) [23, 26, 27]. The search and goodness criteria should not only address the artifacts' immediate goodness of fit (utility), but also the artifacts' potential for evolution in order to stay sustainably useful [20].

Over the course of a project (and possibly several iterations), the involved researchers may produce, test, and discard several instances of the afore-mentioned knowledge entities until they have reached a reasonably accurate understanding of problem and context, and a reasonably well-tested and useful solution. We see only the final results of the project to be candidates for an addition or contribution to the existing human knowledge bases through distinct and rigorous processes or *modes of design theorizing*. In particular, the refined understanding of context and problem primarily contributes to Ω, whereas the artifacts or other solution entities primarily contribute to λ. However, we see also another knowledge type with contribution potential to λ: artifact or entity-independent design knowledge, which we term *solution design knowledge*. Several authors have examined how such artifact-independent design knowledge could bridge Ω- and λ-knowledge [13, 14, 16, 46]. Here, the emphasis has frequently been on linking theory and IS design science research (DSR) artifacts [18, 23, 31]. However, the used terminologies, conceptualizations, representations, roles, and implications vary widely among the sources cited above. In the upcoming sections, we therefore take a closer look to find common ground across solution design knowledge and its roles in design theorizing.

3 Positioning Solution Design Knowledge in Relation to Artifacts and Design Theories

In this and the next section we first lay the foundation for illustrating how separating solution design knowledge from artifacts and design theories on one hand and Ω-knowledge (including theory) on the other hand contributes to a clearer picture on how these knowledge types can enhance artifact design and corresponding knowledge contributions in different design theorizing modes.

Traditionally, IS artifacts as design entities are said to constitute concepts, models, methods, and instantiations [35]. In our conceptualization, it makes sense, however, to understand abstract concepts, models, and methods as components of meta-artifacts, while the corresponding instantiations are components of artifact instances that have a physical existence in the real world [24]. Meta-artifacts are artifacts that lead to the development of other artifacts [28, 30] and constitute more abstract (nomothetic) knowledge about technology. In contrast, knowledge about artifact instances constitutes local (idiographic) knowledge [9]. Drawing on the most recent contributions to the ‘artifact debate’ [4, 10, 15, 29, 32, 43], we conceptualize IS artifacts as 1) consisting of any number of technical, social and/or informational components, 2) collectively supplying one or more functionalities, and 3) thus fulfilling an a) information-related or b) information technology-related purpose. Note that the informational components themselves may actually be comprised of knowledge. Due to space restrictions, we are unable to follow up with this ‘recursion’, however.

How then does solution design knowledge relate to artifacts within λ? In fact, some authors in the literature do not distinguish further between solution design knowledge (by the name of design propositions) and artifacts as an outcome of IS DSR [14]. Instead, they regard a body of design propositions as design theory that should be built on kernel theories, proposed, tested, and subsequently refined. In contrast, for others, solution design knowledge (by the name of techniques) can be artifacts as well – as long as one can formulate a means-end statement for the artifact – but does not necessarily have to be one [18]. In a third perspective, other authors conceptualize solution design knowledge as being distinctly separate from artifacts and theories – for instance, Kuechler and Vaishnavi’s design relevant explanatory and predictive theories [31].

To solve these contradictory conceptualizations, we go back to Simon [50], for whom an artifact exhibits the following key property, among others: An artifact is a human-made entity that constitutes an interface between its inner and an outer environment. As solution design knowledge exists in the knowledge base independently of a specific (class of) immediate application context(s), there is no class of or specific outer environment for solution design knowledge. For this reason, solution design knowledge in our understanding does not fit Simon's artifact definition. Solution design knowledge's nature is more abstract. We therefore argue solution design knowledge to be in a separate realm from knowledge about artifacts, but within λ.

Besides artifacts, design theories are also seen as common outputs of IS DSR [23]. Therefore, it is worthwhile to distinguish these two knowledge contribution types further. Gregor & Jones [24] list eight components of a full-fledged design theory: 1) its purpose and scope, 2) constructs of the entities of interest, 3) principles of form and function, 4) artifact mutability, 5) testable propositions, 6) justificatory knowledge, 7) implementation principles, and 8) an expository instantiation.

In the terminology we have used so far, the principles of form and function closely correspond to the *meta-artifact* design entity and the expository instantiation to the *artifact instance*. The implementation principles highlight the need to consider a complementary *implementation or instantiation 'design'* in addition to an artifact, which transforms a meta-artifact to an instance, and integrates it into a socio-technical system, possibly by means of *interventions* [1, 2]. Likewise, the artifact mutability highlights the need for regular artifact redesigns as part of an *artifact evolution* to retain the artifact's utility over time [20]. The remaining design theory elements provide further guidance during artifact design and evaluation (purpose and scope, testable propositions) or highlight the links to the knowledge bases (constructs, testable propositions, justificatory knowledge).

We therefore conclude that a design theory is more encompassing than a perspective on artifacts alone and provides links to specific knowledge types. For the purposes of this paper, we will, however, take an artifact-centric perspective and highlight more general ways of building on and linking back to knowledge for the different types of artifacts. In keeping with Weick's distinction between theory and theorizing [56], we call these more general ways design theorizing (see Section 7 for further elaboration). Nevertheless, design theories remain an established mode of communicating a DSR knowledge contribution in IS research papers.

4 Positioning Solution Design Knowledge in Relation to Propositional Knowledge

Having highlighted the difference between artifacts, design theories, and solution design knowledge, we now distinguish solution design knowledge from existing understandings of propositional knowledge, including theory.

As with artifacts, the precise definition of theory and its distinction from other forms of theoretical knowledge are contended concepts in the IS and neighboring disciplines [5, 6, 22, 33, 51, 54, 56, 57]. As described in the Introduction, we distinguish between two basic knowledge types: applicable human knowledge (λ-knowledge) and the corresponding foundation in descriptive or propositional knowledge (Ω-knowledge). This broad and high-level distinction allows us to side-step most of the debates around theory and be inclusive to different forms of Ω-knowledge. To do so makes particular sense in the context of utilizing knowledge in DSR projects because design is an inherently creative process where the designers can draw on a wide range of possible knowledge sources in Ω and λ, be it to directly inform their designs, to understand their problems and contexts, or only to spark their creativity.

With respect to Ω-knowledge, the consensus in the debates in the literature in the IS and neighboring disciplines is that potentially worthwhile theoretical research contributions can encompass explanatory and predictive theories in a rather focused understanding of what constitutes a theory [54, 57], but also other conceptual forms of knowledge [3, 36, 38, 58] that are produced by generalizing, specializing, or theorizing real-world phenomena or other existing knowledge about real-world phenomena [5, 25, 48, 49, 56]. For the IS discipline, Gregor [21] proposes five theory types: I) for description, II) for explanation, III) for prediction, IV) for explanation and prediction, V) for design and action. Only type V belongs to λ-knowledge (already covered in the previous section), whereas the other types belong to Ω-knowledge. Likewise, most other forms of theoretical knowledge discussed in the literature belong to Ω-knowledge. A notable exception is Markus' [36] distinction between theories of the problem and theories of the solution. The latter are also called theories of the intervention in Majchrzak & Markus [34] in the context of policy research. These theories of the solution or intervention belong to λ-knowledge, similarly to type V theories.

5 Solution Design Knowledge Types and Sub-types

In this section, we further refine our understanding of solution design knowledge based on solution design knowledge types that are commonly used in the IS literature. Note that the documentation of the underlying literature review and analysis process lies outside the scope of this paper, due to space restrictions. For the same reason, only a limited number of sources are referenced within each section, and only three major types are distinguished.

5.1 Technological Rules: Solution Design Knowledge for Action

In a philosophy of science perspective, Bunge proposes the technological rule concept to document solution design knowledge for action [13]. Note that Bunge understands technology in a wide interpretation that encompasses all knowledge and means to address practical problems. Technological rules ground instrumental knowledge on scientific knowledge. In particular, Bunge uses the term technological theory for theoretical statements that capture the effect of actions aimed at achieving real-world goals. Niiniluoto highlights that such technological theories – or norms, as he calls them – differ from nomological or law statements (which belong to Ω) in that they are instrumental and encompass an aspect of uncertainty [41]. Hence, they can be expressed as follows: "If you want A, and you believe you are in situation B, then you ought to / it is rational for you / it is profitable for you do X." The assumption is that actions based on technological theories – which, in turn, are based on well-founded hypotheses and reasonably precise data – are superior to actions based on unquestioned traditions [13]. Hypotheses and data can thus be used to justify these technological theories.

The concept of technological rules has been picked up in disciplines such as IS [7, 14], management [16], or sociology [45] to semi-formalize actionable knowledge. For instance, Carlsson proposes to extend the components of a technological rule in IS to a PIMCO format – (P)roblem situation, (I)S initiative, (M)echanism, (C)ontext, (O)utcome. Baskerville and Pries-Heje [8] propose to formalize them in the form "(~Z, ~Y) → ~X", meaning "if you want to achieve something like Y in a situation similar to Z, then something like action X will help". Note that each of their three components can, but does not have to be, ambiguous [7]. Their proposal thus highlights the ambiguous nature of technological rules even further than Niiniluoto's version mentioned above [41]. As technological rules present merely options for action and do not constitute a coherent design entity directed at addressing a particular real-world problem or problem class, we understand these rules to be different from IS artifacts as defined in Section 3. We discuss ways of utilizing knowledge for action for artifact design in Section 7.

5.2 Requirements, Principles, Features: Solution Design Knowledge for Entity Realization

A second common type of solution design knowledge in the IS literature concerns intermediate steps on the path towards (meta-)artifact (or solution entity) designs that address real-world problems (or problem classes). Even Walls et al. in 1992 emphasize the importance of deriving design meta-requirements before moving on to the meta-design [53] and requirements engineering is a well-established practice in software engineering [11]. More recently, Meth et al. propose to consider design requirements, principles, and features in this order before actually designing an artifact [37]. The benefits of doing so include a greater transparency with respect to design decisions and a possibility to re-use (meta-)requirements, (meta-)principles, and (meta-)features for the design of similar (meta-)artifacts to address similar problem (classes) in similar context (classes). In other words, these (meta-)requirements, (meta-)principles, and (meta-)features likewise do not represent IS artifacts, but re-usable solution design knowledge for solution entity realization within λ. Simultaneously, they are distinctly different from technological rules or other knowledge for action and thus represent a second distinct type of solution design knowledge. We will further discuss this type's roles in the design process in Section 7.

5.3 Solution Design Knowledge for Design Processes and Systems

Following Walls et al. [53], we also include knowledge for design processes into the realm of solution design knowledge. Further, these design processes often take place in the context of design systems [20]. Our inclusion is based on the rationale that solution design knowledge as actionable knowledge can not only inform the design of solution entities, but also the corresponding design processes and design systems. In research, these design processes largely correspond to the employed research methodologies. The corresponding solution design knowledge is usually presented as methodological contributions. By including design systems, our perspective, however, goes beyond research methodologies and includes, for instance, initiating and retaining engagement with practitioner clients [44] or established techniques such as design thinking [17].

5.4 Other Types and Forms of Solution Design Knowledge

We acknowledge that the solution design knowledge types identified above are not exhaustive and the same applies for the mentioned terminologies and representations. For instance, design patterns are another established type of solution design knowledge for technical artifact design (software [19]), social artifact design (change management [47]), or even for DSR itself [52]. Design patterns can be understood as a particular form of technological rules (representing options for design action) and simultaneously a form of codifying design principles or features as that can be drawn on or applied during artifact design. However, space restrictions prevent us from further exploring other solution design knowledge types that are somewhat less prevalent in the IS literature than the three types discussed in Sections 5.1 to 5.3.

6 A Unified Knowledge Utilization and Contribution Perspective for IS DSR

Based on the distinctions made in Sections 2 to 5, we now propose a conceptual framework (Figure 1) that places the covered realms among and within our two knowledge bases (λ and Ω) and shows six directions to utilize, produce, and contribute knowledge as modes of design theorizing. Figure 1 modifies and extends Figure 1 in Gregor and Hevner [23]. First, our Figure 1 separates project design knowledge from the two knowledge bases. The underlying rationale (cf. Section 2) is that a DSR project, in addition to drawing on and utilizing existing knowledge, produces untested, conjectural, and temporary knowledge and entities in a potentially unstructured, creative, and heuristic manner. This project design knowledge is shared only among the members (or a subset of them) of a single project. In the end, only selected knowledge may turn out to be suitable to be contributed back into the more widely disseminated human knowledge bases (for instance, in the form of publications). Second, our Figure 1 shows how we conceptualize solution design knowledge. Solution design knowledge is actionable or technological knowledge, and therefore belongs to the body of λ-knowledge. It differs from artifacts in λ in that solution design knowledge is independent from a particular manifestation in a distinguishable design entity.

Note that we included implementation / intervention / instantiation processes (shortened to *i-processes* in Figure 1) as a separate solution design entity besides artifacts (cf. Section 3). Likewise, we included processes to redesign existing artifacts as part of a necessary artifact evolution to retain their utility [20] as separate design entities. Moreover, we regard design processes and systems as further and distinct entities beyond the solution entities such as artifacts (cf. Section 5.3).

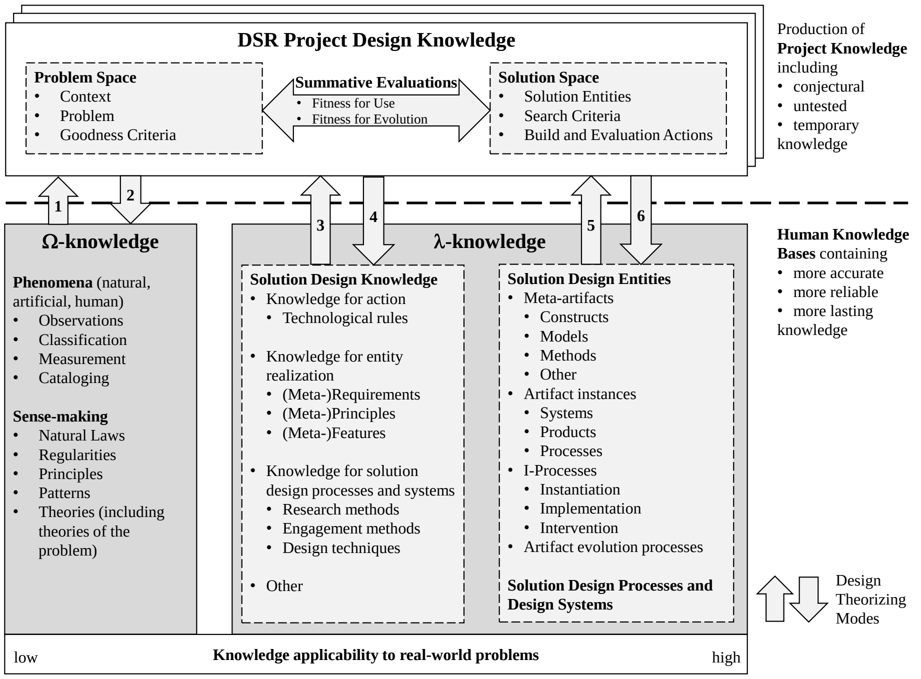

Fig. 1. A unified perspective on knowledge utilization, production, and contribution in IS DSR

7 Modes of Design Theorizing

Figure 1 contains six numbered block arrows that correspond to six design theorizing modes. They all draw on the different knowledge types either to utilize them in a DSR project for the purpose of producing project design knowledge, or to contribute selected knowledge back to a distinct part of the knowledge bases. Note that each mode merely represents an archetype; it is well possible that an actual knowledge contribution draws on more than one mode at a time. Note further that we, for simplicity's sake, focus on the design and evaluation of an IS artifact as solution entity.

7.1 Mode 1: Ω-knowledge Informs the Understanding of a Problem, its Context, or the Design of a Solution Entity

The design theorizing mode utilizes propositional or Ω-knowledge (which includes natural laws or behavioral theories that are formulated as law-like statements) to increase and substantiate the understanding of the context and the problem (mode 1A) or to inform the options for entity realization (mode 1B).

Mode 1A: Understanding the Context and Diagnosing the Problem. One way how Ω-knowledge informs a DSR project is the path towards deriving the (meta-)requirements for a subsequent (meta-)artifact or entity design. Here, the primary interest is the in-depth understanding of the (class of) context(s) and the diagnosis of the real-world problem (class). Both draw on our existing understanding of the real world (and also add to it in the process), so that the researchers can formulate or reformulate a clear goal statement for the design effort and, subsequently, develop specific (meta-)requirements and goodness criteria for evaluating if the solution satisfies the problem. In other words, mode 1A lays the foundation for and forms a key part in theorizing about the requirements for effective and context-specific solutions to a given problem or means to reach a given goal. This mode of design theorizing anchors and guides the subsequent design effort. Beyond Ω-knowledge, creativity and experience and also past requirements for similar design efforts (cf. mode 3B) can inform the requirements.

Mode 1B: Informing the Realization of a Solution Entity. Ω-knowledge can also inform the realization of a solution entity, albeit indirectly. Although Ω-knowledge is evaluated as true/false and λ-knowledge as effective/ineffective [13], it is nevertheless possible to ground substantive technological theories for action or technological rules (as instances of solution design knowledge) on scientific knowledge. This is achieved by predicting and retrodicting (or abducting [40]) reasonably stable norms of behavior and combining these existing behavioral norms with nomopragmatic statements that capture effects of human action. A second source to inform such solution design knowledge are operative theories of action, which draw on non-scientific knowledge, experience, or intuitive insights. However, a rule’s effectiveness cannot be inferred and the whole process is fraught by ambiguity (cf. Section 5.1), assumptions, and the need to reduce the complex reality to specific factors that are understood to significantly affect a rule’s effectiveness [41]. However, in a DSR project such newly produced tentative solution design knowledge is only a precursor to artifact design to inform and justify specific design decisions. Moreover, further layers of ‘obfuscation’ are added by having more than one rule informing the design of an artifact that is later used as a coherent entity to affect the real world.

7.2 Mode 2: The Design and Real-World Application of Solution Entities or Knowledge Enhances our Understanding of the World

During artifact evaluation, the respective findings may confirm, contradict, or extend our original understanding of the real world with respect to the nature of the context or problem (mode 2A) or to the knowledge that that originally informed the formulation of the technological rules underlying artifact design (mode 2B). In a sense, mode 2A corresponds to mode 1A and mode 2B to 1B above.

Mode 2A: Improving the Understanding of the Context and Problem. This design theorizing mode allows improving or extending our general understanding of the real-world context and the addressed problem in particular by uncovering yet un-researched aspects or facets of the context during the artifact evaluation.

Mode 2B: Improving our Understanding of Behavioral Regularities. A separate design theorizing mode may contribute to confirming, challenging, or extending our given understanding of how people, organizations, or societies behave. This can be achieved by comparing the actual effects of an artifact’s interaction with its context with the intended ones, and relating the findings to the interim solution design knowledge developed in mode 1B. A prerequisite to design theorize in this mode 2B is a high traceability of the process in the preceding mode 1B from Ω over the developed interim solution design knowledge to the eventual artifact design. Moreover, such design theorizing has to take into account the resulting technological rules' inherent ambiguity, their combination with other rules, their 'embeddedness' within an artifact, and the limitations when design theorizing from more idiographic to more nomothetic knowledge.

7.3 Mode 3: Solution Design Knowledge Informs the Design of a Solution Entity, a Design Process or a Design System

Each of the three types of solution design knowledge distinguished in Section 5 corresponds to a sub-type of design theorizing.

Mode 3A: Knowledge for Action Informs Solution Entity Design. As already hinted at in the description of Mode 1B, extant knowledge for action (such as technological rules) can inform artifact design. Technological rules represent possible ways of action to achieve goals, while artifact design principles and features seek ways to implement specified requirements to reach a particular goal to address the real-world problem (class) in question. Relying on existing technological rules to inform design decisions (either more abstract (meta-)principles or less abstract (meta-)features) means selecting particular options for action from the range of possible options to be embedded into the (meta-)artifact later on. Here, the artifact thus serves as a vessel to eventually trigger or change human or systems' behaviors so that the problem is addressed or solved. In this context, this mode of design theorizing makes it explicit how particular design decisions are related to the body of solution design knowledge for action.

Mode 3B: Knowledge for Entity Realization Informs Solution Entity Design. As mentioned in Section 5.2 above, one benefit of explicitly documenting (meta-)requirements, principles and features of solution entities is their possible re-use when designing similar entities to address similar problems in similar contexts. A key aspect of this design theorizing mode is to deal with the issue of 'similarity' between problems and contexts and the corresponding 'projectability' of the existing (meta-)requirements, principles and features to different problems, contexts, and – eventually – solution entities [8]. Note that designing specific artifact features that implement more abstract principles to satisfy requirements and that are based on the most suited action options is still a wholly distinct step from actually implementing these features in instances of social or information technologies. Often, there are a multitude of ways how even well-defined features can actually be implemented in a design entity. The same applies for the corresponding implementation / instantiation / intervention to bring about the necessary changes in the social and/or technical system.

Mode 3C: Knowledge for Solution Design Processes and System Informs the Design of the DSR Project System. A third design theorizing mode can inform the design of the design processes and design system. However, as stated above, we focus mainly on the artifact or solution entity design and not the DSR project system design in this paper, due to space restrictions.

7.4 Mode 4: Effective Principles, Features, Actions, or Effects of a Solution Entity or a Design Process or System are Generalized and Codified in Solution Design Knowledge

There are three sub-types of this design theorizing mode that are counterparts to the three sub-types for mode 3.

Mode 4A: Codifying Effective Actions. During artifact evaluation, it is among the researchers' tasks to evaluate the chosen technological rules that informed the design decisions. In this design theorizing mode, effective technological rules that are new or need to be changed are codified to add / change them in the knowledge base. In additional, the evidence level of extant technological rules that proved to be effective can be raised [18]. Due to the pervasive ambiguities both in the artifact's application context, the preceding 'chain' of design decisions and developed artifact knowledge, and a possible necessity to move from more idiographic to more nomothetic knowledge, such attributions may not be trivial, yet are crucial for this design theorizing mode.

Mode 4B: Codifying Effective Design Principles or Features. In this mode, it is the researchers' task to evaluate how effective facets of the resulting artifact can be isolated with respect to underlying features, principles, and/or requirements that future DSR projects can draw on. The challenges mentioned in mode 4A apply here likewise.

Mode 4C: Codifying Effective Aspects of Design Processes or Design Systems. This mode is the counterpart to mode 3C and largely covers individual methodological knowledge contributions that arise out of a DSR project.

7.5 Mode 5: Previously Effective Solution Entities, Design Processes, or Design Systems are Re-used for or Inform Future Designs of New Entities, Processes, or Systems

There are two sub-types for this design theorizing mode, one regarding solution entity design and one regarding the design of design processes and systems.

Mode 5A: Re-using Previous Solution Entities. This mode is similar to mode 3B, except that the researchers here draw on previously effective artifacts to either re-use or adapt them in similar contexts for similar problems, or to merely inform their (meta-)principles, features or actual solution entity design. The afore-mentioned issues of similarity and projectability apply likewise.

Mode 5B: Re-using Previous Design Processes and Systems. Similarly to mode 3C, this design theorizing mode is concerned with re-using not mere knowledge about design processes and systems, but a more coherent and extensive set of design processes, methodologies, and systems in a new DSR project.

7.6 Mode 6: Effective Solution Entities, Design Processes, or Design Systems are Contributed to λ-knowledge

As with the previous even-numbered design theorizing modes, the number of sub-types correspond to the number of sub-types among the preceding odd-numbered design theorizing mode, and form their counterparts.

Mode 6A: Contribute Effective Solution Entities. This design theorizing mode is probably the most established mode in IS DSR among all the discussed ones in this paper. It essentially concerns documenting the final resulting solution entity (e.g., artifact) and demonstrating its utility to address the initial problem (class) in a (class of) contexts.

Mode 6B: Contribute Effective Design Processes and Systems. Similar to mode 4C – however, the emphasis here is less on individual methodological contributions, but more on contributing a coherent set of design processes, methods, techniques, and systems that effectively work together in a specific context to address a specific problem.

8 Discussion, Conclusion and Outlook

We identified two substantially different forms of design knowledge (project design knowledge and solution design knowledge), developed a comprehensive perspective on knowledge utilization, production, and contribution in DSR, and identified several design theorizing modes that either utilize knowledge to inform entity realization or contribute knowledge to the two knowledge bases Ω and λ.

Separating DSR project design knowledge from the two knowledge bases extends the prevalent perspective on knowledge growth [23] by highlighting the DSR project's role as 'knowledge engine'. DSR projects produce a plethora of – possibly temporary, conjectural, creative, and untested – knowledge and utilizing and contributing well-tested knowledge (including artifacts) to the knowledge bases (= the modes of design theorizing) are separate and distinct actions that researchers need to take, possibly through several iterations. We envision project repositories that maintain this design knowledge for future use and reflection.

Distinguishing solution design knowledge from Ω and design entities such as artifacts contributes to a more refined understanding of artifact-independent technological and actionable knowledge. While – what we have called – solution design knowledge has received its share of attention in the literature so far, there has been a lack of a comprehensive perspective that can unify several different extant research streams that all approach solution design knowledge from different angles (and possibly conflate solution design knowledge with solution entities). Here, our perspective provides a foundation for refining our understanding of further solution design knowledge types and their relation with one another and with Ω-knowledge and artifacts in the future.

We further discussed several design theorizing modes that utilize the different knowledge types within Ω and λ to improve the process and the outcome of DSR projects (including the production of project design knowledge). Our proposed design theorizing modes advances the DSR discourse beyond the debate around artifact-centric or design theory-centric DSR by emphasizing (1) the important roles all the other forms of knowledge can play within all DSR project phases and (2) the different knowledge contributions a DSR project can make to Ω and λ. These contributions may comprise additions to the knowledge base as well as 'subtractions' in the form of challenging or refuting knowledge that was believed to be valid or effective.

We believe that an increased attention to growing all forms of knowledge – in addition to designing artifacts and developing design theories – can play a crucial role for the maturing DSR paradigm within and beyond IS research. We therefore issue a call to all design researchers for bolder and more explicit design theorizing to make increasing and stronger knowledge contributions of all types, including artifact or design theory-independent contributions. Simultaneously, we would like to emphasize that we do not see every design theorizing mode to be mandatory for design science researchers to consider. Instead, we see these as opportunities for knowledge re-use, enhancement, or the stimulation of creativity. Sometimes, a bold leap of imagination or deep intuitive insight may be what a design project requires instead of a rigorous but ultimately more limited by-the-book design theorizing, in order to have a substantial impact on organizations and society and to make substantial knowledge contributions at the end.

Note that our perspective on design knowledge as proposed in this paper is limited in several ways, however. First, space restrictions only allow us to begin exploring the various extant contributions in the literature that cover – what we have called – solution design knowledge and its different roles and representations. The same applies to the corresponding modes of design theorizing. Second, based on our focus on the artifact in design knowledge, we have not proposed a definitive representation of *design theory* in our Figure 1 framework. As discussed in Section 3, we find aspects of design theory throughout the knowledge bases. Finally, our current perspective only remains on the conceptual level and lacks a practical example or application. We therefore see it as a task for further research to develop an even more comprehensive understanding of design knowledge and modes of design theorizing and to demonstrate the corresponding benefits and challenges for design science researchers in actual DSR projects. By doing so, we ultimately believe to improve the utility of the resulting designs and to improve the quantity and quality of contributions to all knowledge bases.

9 References

1. Aken, J.E. van: Design Science and Organization Development Interventions. J. Appl. Behav. Sci. 43, 1, 67–88 (2007).
2. Aken, J.E. van: Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. J. Manag. Stud. 41, 2, 219–246 (2004).
3. Alter, S.: Nothing is more practical than a good conceptual artifact… which may be a theory, framework, model, metaphor, paradigm or perhaps some other abstraction. Inf. Syst. J. n/a-n/a (2016).
4. Alter, S.: The concept of 'IT artifact' has outlived its usefulness and should be retired now. Inf. Syst. J. 25, 1, 47–60 (2015).
5. Avison, D., Malaurent, J.: Is theory king?: a rejoinder. J. Inf. Technol. 29, 4, 358–361 (2014).
6. Avison, D., Malaurent, J.: Is theory king?: questioning the theory fetish in information systems. J. Inf. Technol. 29, 4, 327–336 (2014).
7. Baskerville, R., Pries-Heje, J.: Design Logic and the Ambiguity Operator. In: Global Perspectives on Design Science Research. pp. 180–193 Springer, Berlin, Heidelberg (2010).
8. Baskerville, R., Pries-Heje, J.: Design theory projectability. In: Working Conference on Information Systems and Organizations. pp. 219–232 Springer (2014).

9. Baskerville, R.L. et al.: Genres of inquiry in design-science research: Justification and evaluation of knowledge production. Mis Q. 39, 3, 541–564 (2015).
10. Benbasat, I., Zmud, R.W.: The Identity Crisis within the Is Discipline: Defining and Communicating the Discipline's Core Properties. MIS Q. 27, 2, 183–194 (2003).
11. Braun, R. et al.: Proposal for Requirements Driven Design Science Research. In: Donnellan, B. et al. (eds.) New Horizons in Design Science: Broadening the Research Agenda. pp. 135–151 Springer International Publishing (2015).
12. Briggs, R.O., Schwabe, G.: On Expanding the Scope of Design Science in IS Research. In: Jain, H. et al. (eds.) Service-Oriented Perspectives in Design Science Research. pp. 92–106 Springer Berlin Heidelberg (2011).
13. Bunge, M.: Scientific Research II: The Search for Truth. Springer, Berlin (1967).
14. Carlsson, S.A.: Design Science Research in Information Systems: A Critical Realist Approach. In: Hevner, A. and Chatterjee, S. (eds.) Design Research in Information Systems: Theory and Practice. pp. 209–233 Springer, New York (2010).
15. Chatterjee, S. et al.: The Information Systems Artifact: A Conceptualization Based on General Systems Theory. Presented at the HICSS (2017).
16. Denyer, D. et al.: Developing Design Propositions through Research Synthesis. Organ. Stud. 29, 3, 393–413 (2008).
17. Dorst, K.: The core of 'design thinking' and its application. Des. Stud. 32, 6, 521–532 (2011).
18. Fettke, P. et al.: On the Relevance of Design Knowledge for Design-Oriented Business and Information Systems Engineering - Conceptual Foundations, Application Example, and Implications. Bus. Inf. Syst. Eng. 2, 6, 347–358 (2010).
19. Gamma, E. et al.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley Longman, Amsterdam (1994).
20. Gill, T.G., Hevner, A.R.: A Fitness-Utility Model for Design Science Research. ACM Trans. Manag. Inf. Syst. 4, 2, 5:1–5:24 (2013).
21. Gregor, S.: The nature of theory in information systems. MIS Q. 30, 3, 611–642 (2006).
22. Gregor, S.: Theory – still king but needing a revolution! J. Inf. Technol. 29, 4, 337–340 (2014).
23. Gregor, S., Hevner, A.R.: Positioning and Presenting Design Science Research for Maximum Impact. MIS Q. 37, 2, 337-A6 (2013).
24. Gregor, S., Jones, D.: The Anatomy of a Design Theory. J. Assoc. Inf. Syst. 8, 5, 312–335 (2007).
25. Hassan, N.R., Lowry, P.B.: Seeking Middle-Range Theories in Information Systems Research. Presented at the (2015).
26. Hevner, A.: A Three Cycle View of Design Science Research. Scand. J. Inf. Syst. 19, 2, 87–92 (2007).
27. Hevner, A. et al.: Design Science in Information Systems Research. MIS Q. 28, 1, 75–105 (2004).
28. Iivari, J.: Distinguishing and contrasting two strategies for design science research. Eur. J. Inf. Syst. 24, 1, 107–115 (2015).
29. Iivari, J.: Information system artefact or information system application: that is the question. Inf. Syst. J. n/a-n/a (2016).
30. Iivari, J.: The IS Core - VII: Towards Information Systems as a Science of Meta-Artifacts. Commun. Assoc. Inf. Syst. 12, 1, Article 37 (2003).
31. Kuechler, W., Vaishnavi, V.: A Framework for Theory Development in Design Science Research: Multiple Perspectives. J. Assoc. Inf. Syst. 13, 6, 395–423 (2012).
32. Lee, A.S. et al.: Going back to basics in design science: from the information technology artifact to the information systems artifact. Inf. Syst. J. 25, 1, 5–21 (2015).

33. Lee, A.S.: Theory is king? But first, what is theory? J. Inf. Technol. 29, 4, 350–352 (2014).
34. Majchrzak, A., Markus, M.L.: Methods for Policy Research: Taking Socially Responsible Action. SAGE Publications, Ltd, 1 Oliver's Yard, 55 City Road London EC1Y 1SP (2014).
35. March, S., Smith, G.: Design and natural science research on information technology. Decis. Support Syst. 15, 4, 251–266 (1995).
36. Markus, M.L.: Maybe not the king, but an invaluable subordinate. J. Inf. Technol. 29, 4, 341–345 (2014).
37. Meth, H. et al.: Designing a Requirement Mining System. J. Assoc. Inf. Syst. 16, 9, (2015).
38. Miller, D.: Paradigm prison, or in praise of atheoretic research. Strateg. Organ. 5, 2, 177–184 (2007).
39. Niederman, F., March, S.T.: Design science and the accumulation of knowledge in the information systems discipline. ACM Trans Manage Inf Syst. 3, 1, 1:1–1:15 (2012).
40. Niiniluoto, I.: Defending Abduction. Philos. Sci. 66, S436–S451 (1999).
41. Niiniluoto, I.: The aim and structure of applied research. Erkenntnis. 38, 1, 1–21 (1993).
42. Nunamaker, Jr., J.F., Briggs, R.O.: Toward a broader vision for Information Systems. ACM Trans Manage Inf Syst. 2, 4, 20:1–20:12 (2012).
43. Orlikowski, W.J., Iacono, C.S.: Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact. Inf. Syst. Res. 12, 2, 121–134 (2001).
44. Otto, B., Osterle, H.: Principles for Knowledge Creation in Collaborative Design Science Research. ICIS 2012 Proc. (2012).
45. Pawson, R.: Evidence-based policy a realist perspective. SAGE, London; Thousand Oaks, Calif. (2006).
46. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare, http://purao.ist.psu.edu/working-papers/dare-purao.pdf, (2002).
47. Rising, L., Manns, M.L.: Fearless Change: Patterns for Introducing New Ideas: Introducing Patterns into Organizations. Addison-Wesley Longman, Amsterdam (2004).
48. Seddon, P.B., Scheepers, R.: Generalization in IS research: a critique of the conflicting positions of Lee & Baskerville and Tsang & Williams. J. Inf. Technol. 30, 1, 30–43 (2015).
49. Seddon, P.B., Scheepers, R.: Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. Eur. J. Inf. Syst. 21, 1, 6–21 (2012).
50. Simon, H.A.: The Sciences of the Artificial. MIT Press, Cambridge, Massachusetts (1996).
51. Sutton, R.I., Staw, B.M.: What Theory is Not. Adm. Sci. Q. 40, 3, 371–384 (1995).
52. Vaishnavi, V., Kuechler, W.: Design Science Research Methods and Patterns. Auerbach, Boca Raton, Florida (2008).
53. Walls, J.G. et al.: Building an information system design theory for vigilant EIS. Inf. Syst. Res. 3, 1, 36–59 (1992).
54. Weber, R.: Evaluating and Developing Theories in the Information Systems Discipline. J. Assoc. Inf. Syst. 13, 1, 1–30 (2012).
55. Weick, K.E.: Theory Construction as Disciplined Imagination. Acad. Manage. Rev. 14, 516–531 (1989).
56. Weick, K.E.: What Theory is Not, Theorizing Is. Adm. Sci. Q. 40, 3, 385–390 (1995).
57. Whetten, D.A.: What constitutes a theoretical contribution? Acad. Manage. Rev. 14, 4, 490–495 (1989).
58. Yadav, M.S.: The Decline of Conceptual Articles and Implications for Knowledge Development. J. Mark. 74, 1, 1–19 (2010).