Journal of the Association for Information Systems (2020) **21**(3), 520-544
doi: 10.17705/1jais.00611

EDITORIAL

ISSN 1536-9323

Special Issue Editorial – Accumulation and Evolution of Design Knowledge in Design Science Research: A Journey Through Time and Space

Jan vom Brocke[1], Robert Winter[2], Alan Hevner[3], Alexander Maedche[4]

[1]University of Liechtenstein, Liechtenstein, jan.vom.brocke@uni.li
[2]University of St. Gallen, Switzerland, robert.winter@unisg.ch
[3]University of South Florida, USA, ahevner@usf.edu
[4]Karlsruhe Institute of Technology, Germany, alexander.maedche@kit.edu

Abstract

Sir Isaac Newton (1676) famously said, “If I have seen further, it is by standing on the shoulders of giants.” Research is a collaborative, evolutionary endeavor—and it is no different with design science research (DSR), which builds upon existing design knowledge and creates new design knowledge to pass on to future projects. However, despite the vast, growing body of DSR contributions, scant evidence of the accumulation and evolution of design knowledge has been articulated in an organized DSR body of knowledge. Most contributions rather stand on their own feet than on the shoulders of giants, and this continues to limit how far we can see, curtailing the extent of the broader impacts that can be made through DSR. In this editorial, we aim at providing guidance on how to position design knowledge contributions in wider problem and solution spaces. We propose (1) a model conceptualizing design knowledge as a resilient relationship between problem and solution spaces, (2) a model that demonstrates how individual DSR projects consume and produce design knowledge, (3) a map to position a design knowledge contribution in problem and solution spaces, and (4) principles on how to use this map in a DSR project. We show how fellow researchers, readers, editors, and reviewers, as well as the IS community as a whole, can make use of these proposals, and also illustrate future research opportunities.

Keywords: Science Research, Design Knowledge, Knowledge Bases, Problem Space, Solution Space, Accumulation, Evolution

1 Introduction

Design science research (DSR) aims to generate prescriptive knowledge about the design of information systems (IS) artifacts, such as software, methods, models, or concepts (Hevner et al., 2004). Design knowledge (DK) is about means-end relationships between problem and solution spaces (Venable, 2006) and can be represented in different forms, such as designed artifacts (Hevner et al., 2004), design principles (Chandra, Seidel, & Gregor, 2015), and design theories (Gregor & Jones, 2007). As such, DK has been described as taking different forms—for example, the situated implementation of an artifact, nascent design theory, and well-developed midrange design theory (Gregor & Hevner, 2013).

Given the aim of generating prescriptive knowledge, DSR contributes to both the theory and practice of solving real-world problems. DSR projects must provide both intellectual merit in creative designs and broader impacts to the application domain via original problem solutions (Baskerville et al., 2018; Hevner et al., 2004) and have the opportunity to demonstrate the rigor and relevance of IS as an academic field (Lee, 2015; vom Brocke et al., 2013; Watson, Boudreau, & Chen, 2010). The wide-ranging discourse on the goals and the potential impact of the IS discipline (e.g., Bichler, Heinzl, & Winter, 2015; Grover & Lyytinen, 2015; Gupta, 2017; Nunamaker, Twyman, Giboney, & Briggs, 2017) has led to a broad understanding that IS research should contribute solutions to real-world challenges (e.g., Becker, vom Brocke, Heddier, & Seidel, 2015). The increasing digitalization in all areas of the economy and society offers a particular opportunity and responsibility for the IS field. In particular, the *MIS Quarterly* editorial on the diversity of DSR (Rai, 2017) highlights many diverse opportunities to effectively apply DSR for the solution of important IS research challenges.

The methodological discourse of DSR has made significant progress during recent years and robust guidance is now available on how to conduct DSR and how to derive prescriptive knowledge in addressing practically relevant challenges (e.g., Gregor & Hevner, 2013; Hevner et al., 2004; Peffers et al., 2007; Sein et al., 2011). In addition, an increasing number of studies applying DSR methodology have been conducted, such as those discussed in the study of Prat et al., (2015), who analyzed 10 years of DSR publications in the Association for Information Systems Senior Scholars' Basket of journals.

Despite the potential of DSR to guide impactful IS research, we observe a major hurdle that limits our ability to realize this potential, namely the scarce reuse of extant contributions and the limited accumulation and evolution of DK in DSR. To date, most studies focus on a single DSR project, aiming at deriving DK *within* a project, while knowledge accumulation and evolution *across* projects is rarely considered as an antecedent or contribution of the project. Peffers et al. (2007), for instance, define the nominal DSR process sequence as starting with problem identification and continuing through objectives of definition, design and development, demonstration, evaluation, and the communication of results. While conceptually multiple entry points to this process could be considered, most studies start by carving out a problem and eventually presenting a solution to the problem identified, usually resulting from multiple iterations of the DSR process. The limited knowledge accumulation and evolution of DK in DSR as observed in the IS community is problematic because single contributions tend to remain isolated with little to no relation to other solutions. We refer to this as the *monolithic structure* of DK, which hinders the *reuse* of DK. Since solutions to real-world challenges tend to be complex and often require contributions from various contributors, it would be beneficial to follow a model to *compose* DK of extant DK from multiple perspectives over time. Further, both problem and solution spaces are subject to constant and increasing change, so that past DK is prone to rapid aging, which we refer to as the *ephemeral nature* of DK. Hence, DK requires constant *updates* in the form of revision and further evolutionary development.

In summary, we identify the following problems. First, current DSR projects miss the opportunity to *reuse* DK, which would increase both the efficiency and effectiveness of the research process. Second, DSR projects miss the opportunity to *compose* DK contributions toward building solutions to more complex real-world problems. Third, DSR projects, (once they are published), lack validity checks of DK such as currency and timeliness, thus missing the opportunity to *update* DK as needed.

As a prerequisite to support the accumulation and evolution of DK, elementary DSR contributions need to be specifically positioned in terms of the problems (within the problem space) and the solutions (within the solution space) they address. This supports future research that would build on and extend knowledge contributions and would thus reuse extant knowledge, compose DK, and evaluate the validity of DK over time. As more specific DSR contributions are positioned within the problem and solution spaces, it will become easier to establish processes to update DK in terms of the constantly changing aspects of problem and solution spaces. At present, however, the field lacks conceptual and methodological support to specify these problem and solution spaces. Our objective in this editorial, then, is to develop approaches and models that would better position DK contributions to support knowledge accumulation and evolution in DSR.

2 A Model of Design Knowledge

Simply stated, the goal of DSR is to generate knowledge on how to effectively build innovative solutions to important problems. However, the DK produced in a DSR project can be richly multifaceted. DK includes information about the *important problem*, the *designed solution*, and the *evaluation evidence*, as well as *measures of timely progress* regarding how well the problem solution satisfies the key problem stakeholders. The basic three components of DK are *problem space*, *solution space*, and *evaluation*. While both problem space knowledge and solution space knowledge exist independently, it is only through relating them to each another that DK emerges. Figure 1 provides a simple model conceptualizing the important components of DK, and the following discussion presents a brief summary of the three key components.

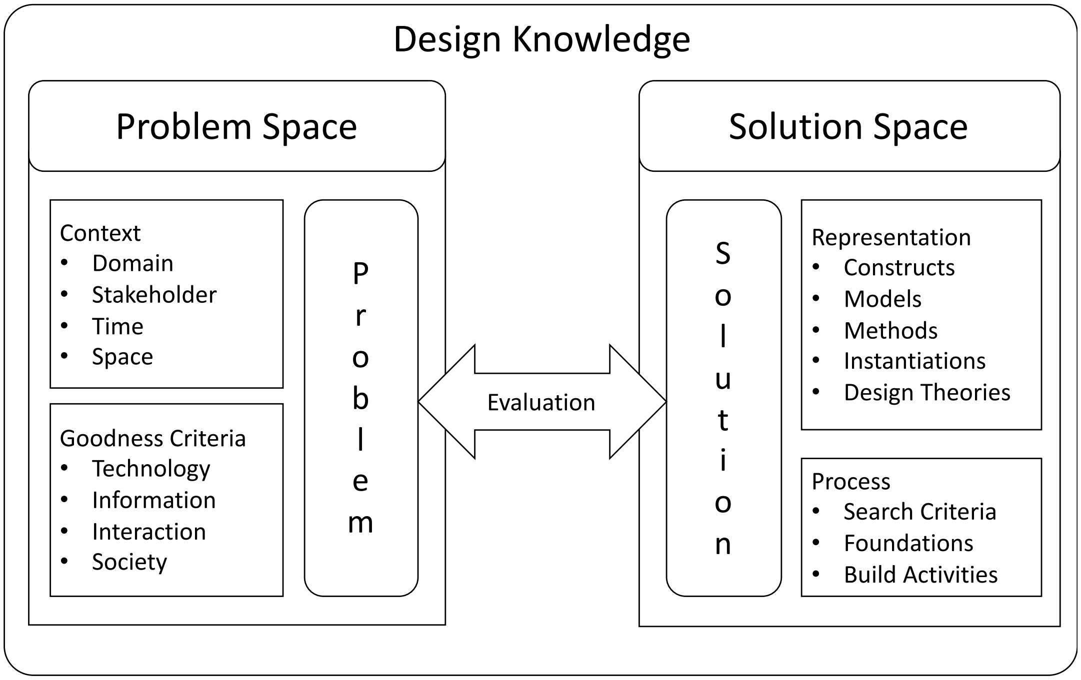

Figure 1. DK Model: Components of Design Knowledge for a Specific DSR Project

2.1 The Problem Space as Design Knowledge

DSR projects seek to produce knowledge about how to solve important problems in a defined application domain. A detailed understanding and description of the problem and its positioning in the problem space are essential to demonstrate the relevance of the research project. In fact, carving out the problem, learning about its relevant space, scoping and sizing a problem to be focused on in a DSR project, and rescoping and resizing it as the DSR project evolves, are important activities within every DSR project. As shown in Figure 1, there are two key DK components that describe the problem space to which DK relates: the application context and the goodness criteria for solution acceptance.

The application context information provides a rich description of the problem in context. What is the problem domain? Who are the key stakeholders in the problem space who will impact and be impacted by the design solution? Also, problem spaces are closely tied to time and location (i.e., space). A problem that is relevant today may not be as relevant tomorrow. Therefore, a clear fixing of the time period during which the problem was perceived and understood as such is essential. Contextual aspects of location include relevant geographic particulars, such as rural versus urban environments and developed versus developing countries. Overall, the application context of a DSR project defines an idiographic basis for the dissemination of DK (Baskerville, Kaul, & Storey, 2015).

The second key DK component regarding the problem space addresses the meaning and requirements for how well a design solution solves the problem in context. When describing the goodness criteria for the problem, we must recognize the sociotechnical aspects of any practical design solution. Thus, design requirements for satisfactory solutions should include a rich mix of goals from the categories of technology (e.g., security, reliability, performance), information quality (e.g., accuracy, timeliness), human interaction (e.g., usability, user experience), and societal needs (e.g., accessibility, fairness). The description of these solution goodness criteria provides a rigorous set of acceptance criteria for the evaluation of potential design solutions and establishes guidance for the design of both formative and summative evaluation methods (Sonnenberg & vom Brocke, 2012; Venable, Pries-Heje, & Baskerville, 2016).

Thus, positioning a DSR project in the problem space establishes the project's situational context and research goals (i.e., goodness criteria for design innovation). The effective reuse of DK for future research is predicated on how well this problem space *projects* onto a new research project. The *projectability* of DK is defined as how well the new research context and goals align with the context and goals of the grounding projects from the knowledge base (Baskerville & Pries-Heje, 2014; Baskerville & Pries-Heje, 2019). This context, as outlined in the DK model (see Figure 1), can be described in terms of different dimensions including domain, stakeholder, time, and place. Low projectability of DK in a project would indicate a very specific context with restrictive goals. In contrast, high projectability of DK would support more general applications of the DK to problem classes within and/or between different application domains.

2.2 The Solution Space as Design Knowledge

DK in the solution space encompasses knowledge that can be used to solve related problems. It specifically includes both the results and activities of DSR (Gregor & Hevner, 2013; Hevner et al., 2004; Peffers et al., 2007). Results of DSR can take different forms, such as designed artifacts (i.e., constructs, solution models, methods, and instantiations) as well as design principles or design theories. Artifacts are representations that support replication and reuse by future research projects. Design theories and principles in the form of nascent theories and midrange theories generalize an understanding of how and why artifacts satisfy the goals of the problem space.

Knowledge in the solution space can also refer to design processes that encompass build activities that contribute to creating, assessing, and refining the DSR results in iterative build-evaluation cycles. Build activities incorporate a search process to identify the best design candidates in the solution space. Information on goodness criteria from the problem space is used to guide a goal-driven search to maximize value that is nevertheless constrained by the availability and feasibility of resources. For future DK reuse, it is important to include support for the design foundations in, for example, the form of kernel theories, and record the creative insights that led to innovative design improvements.

In the solution space model, specific DSR project solutions vary in their *fitness* to solve selected aspects of the target problem. Research may begin with rather incomplete solutions that only cover parts of the problem or only solve certain aspects of the problem. In the course of continuing design activities within a project and across projects, a solution can improve its fit by addressing a larger part of the problem space in more effective ways.

The more “fit” a solution is, the more operational the solution will be for users seeking to apply it (to solve) a targeted real-world problem. The level of fitness also relates to the normative power of a solution, in that lower-levels of fitness (e.g., principles of design) may cause a solution to have lower normative power to guide actual situated problem-solving behavior, meaning that it is less prescriptive than more detailed reference models or manuals. Thus, the lower the fitness of the solution, the greater the effort necessary to apply the DK to a new problem.

It stands to reason that there is a trade-off between the projectability and the fitness of DK. Often, higher levels of DK fitness imply greater limitation to a specific context. A less fit representation of DK, in turn, may support higher projectability. Techniques to represent DK in reusable ways have been developed (vom Brocke & Buddendick, 2006) and include configurative models or methods, which allow for managing the tradeoff of projectability and fitness of DK. In the application of such techniques, DK presents alternative variants of solutions that can be selected to fit different contexts in the problem space. Beyond configuration, alternative techniques, such as configuration, instantiation, specialization, aggregation, and analogy, have been developed in conceptual modeling research (cf. vom Brocke, 2007). For example, configuration techniques have been used to develop situational methods in order to reuse solutions in a wide range of problem settings (Winter, 2012).

2.3 The Evaluation as Design Knowledge

Evaluations link solutions (in the solution space) to problems (in the problem space) and provide evidence of the extent to which a solution solves a problem using the chosen, specific evaluation method. Conceptually, both formative and summative evaluations can be distinguished (Venable et al., 2016). Increasingly, evaluation is being described as a continuously organized process (Sonnenberg & vom Brocke, 2012) that derives early feedback information on how to further develop a solution (Abraham, Aier, & Winter, 2014).

We use the term *confidence* to measure the assessed qualities of the evaluations performed on the existing DK. The level of DK confidence assesses such qualities as the types of evaluation performed (Hevner et al., 2004), the rigor of the evaluation methods, and the convincing nature of the evaluation results. DK with higher evaluation confidence is less risky to use than DK with lower evaluation confidence.

We note that not all DSR projects have the opportunity to test new design artifacts in realistic environments. In such cases, opportunities for evaluations in artificial environments should be considered (e.g., simulation) (Prat et al., 2015). Given the great variety of different methods and application scenarios for evaluations, transparency of both the process and the results of the evaluation are important quality confidence criteria for DK contributions.

Beyond the utility a solution provides regarding a problem (along the lines of Gill & Hevner, 2013), we differentiate two distinct types of design evaluations that can be performed in a DSR project. *Fitness for use* evaluations assess the ability of a design artifact to perform in the current application context with the current set of goals in the problem space. This is the most common type of evaluation in DSR today. *Fitness for evolution* evaluations assess the ability of the solution to adapt to changes in the problem space over time. This type of evaluation is critical for application environments in which rapid technology or human interaction changes are inevitable and successful solutions must evolve. These two forms of evaluation demand a focus on very different measures of goodness, as discussed in Gill and Hevner (2013).

We also propose that measures of solution progress be included as part of evaluation DK (Aier & Fischer, 2011; Lukyanenko, Evermann, & Parsons, 2014). As DSR projects are longitudinal over time with continually changing problem and solution spaces, we hope to identify and measure points of stability amid evolutionary progress. Thus, it can be claimed that design improvements are measurable advances on well-defined goodness criteria from the problem space. Such measures of progress may change over time but some common understanding of solution progress and improvement is essential for tracking DK evolution.

2.4 Applying the DK Model

When applying the DK model in DSR projects, multifaceted interdependencies between the problem and solution space must be considered. Neither a pure analysis of the problem space (without considering the relevant subset of the solution space) nor a pure analysis of the solution space (without a certain subset of problem space in mind) would be very useful. Thus, we propose that an analysis of both the problem and solution space should be performed simultaneously. Carving out the problem space for a specific DSR project is just as important as articulating the current state of solution DK (e.g., existing artifacts and design theories in use) for this domain. In early phases of a project, these activities are often conducted without formal problem specification or solution design and evaluation; rather, they are generally based on expert opinion and prior knowledge.

Our conceptualization shows that there are different levels of maturity that DK can assume and that such maturity can be differentiated in terms of each of the three DK components, i.e., *projectability* of the problem in problem space, *fitness* of the solution in solution space, and *confidence* in the current evaluation evidence. Beyond evaluating changes in DK as a result of progress in design activities, changes in both the problem and solution space also must be considered in light of the *ephemeral nature* of DK. Available technologies, scientific theory bases, government regulations, national and international laws, and societal mores change over time. Also, changes in these spaces may require new evaluations to be performed in order to maintain and increase confidence in DK use.

The three components of DK can be used to plan, coordinate, and communicate complex design research activities over time and space, even if they involve multiple projects and different researchers. In the following section, we outline related mechanisms in more detail.

3 Modes of Design Knowledge Production and Consumption

Basic knowledge can be represented by two major types: (1) research activities that primarily grow Ω-knowledge (comprising descriptive, explanatory and predictive knowledge), and (2) research activities that primarily grow λ-knowledge (comprising design knowledge) (Gregor & Hevner, 2013). Contributions to λ-knowledge typically deal with technological (in the sense of means-end) innovations that directly impact individuals, organizations, or society and also enable the development of future technological innovations (Winter & Albani, 2013). We refer to λ-knowledge as DK. Contributions to Ω-knowledge enhance our understanding of the world and the phenomena that technologies harness (or cause). Research projects may combine both genres of inquiry and contribute to both knowledge bases.

The relationship of specific DK created in DSR projects and the general DK base is illustrated in Figure 2. This figure is adapted and simplified from Drechsler and Hevner (2018) and clearly illustrates paired modes of consuming and producing knowledge between the DSR project and the descriptive and prescriptive knowledge bases. The λ-knowledge is divided into two subcategories. *Design entities* collect prescriptive knowledge as represented in tangible artifacts and processes that are designed and applied in the solution space. The growth of design theories around these solutions is captured in the *design theories* knowledge base (Gregor & Hevner, 2013). Knowledge can be projected from the application into nascent theories around solution actions, entity realizations, and design processes based on the new and interesting DK produced in a DSR project. Thus, we can describe the interactions of a DSR project with the extant knowledge bases in the following consuming and producing modes:

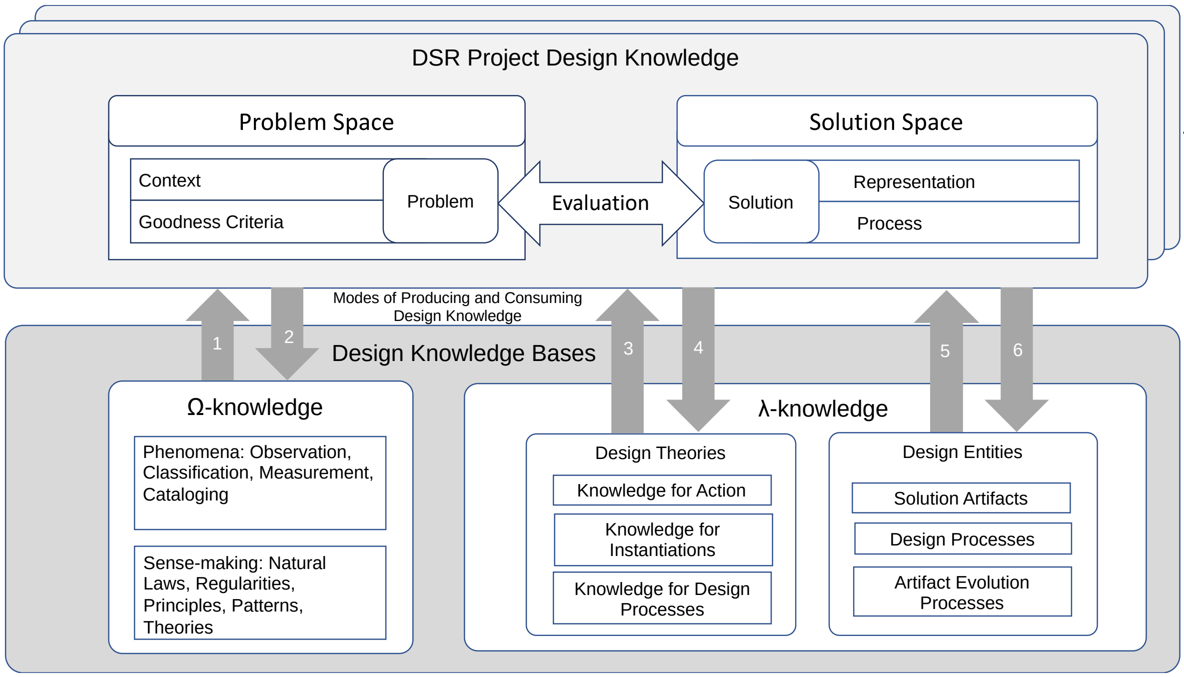

Figure 2. Interactions of DSR Projects and Design Knowledge Bases (Adapted from Drechsler & Hevner, 2018)

- **Modes 1 and 2—building on and contributing to Ω-knowledge:** Ω-knowledge informs the understanding of a problem, its context, and/or the development of a design entity (Mode 1: kernel theory to design entity/theory grounding). The design and real-world application of solution space DK also change the world, thus inducing the testing and building of Ω-knowledge, which enhances our descriptive understanding of how the world works given the new DK—Mode 2: design entity/theory to (kernel) theory complement.
- **Modes 3 and 4—building on and contributing to design theory:** Solution DK, in the form of expanding design theory, informs the development of a design entity, i.e., a design process or a design system (Mode 3: design theory to design entity grounding). Within a DSR project, effective principles, features, actions, or effects of a design entity are generalized and codified in solution DK (e.g., design theories or technological rules)—Mode 4: design entity to design theory complement.
- **Modes 5 and 6—Building on and contributing to design entities:** Previously effective design entities and design processes are reused to inform novel designs of new design entities (Mode 5: design entity to design entity reuse). Within a DSR project, effective design entities are contributed to DK (Mode 6: design entity to design entity complement).

The six modes of producing and consuming DK illustrate the multifaceted opportunities for knowledge accumulation and evolution that arise when looking beyond single DSR projects and organizing DK contributions over time and across projects. In the following section, we provide further support for planning, coordinating, and communicating longer "journeys" of DK consumption and production.

4 Design Knowledge Map

In order to organize DK contributions over time and across projects, researchers need to be able to position contributions appropriately in the DK space. More specifically, it is important to (1) allocate a single DK contribution, and (2) articulate the relationships among DK contributions. Researchers can then build on extant DK contributions more easily and further develop extant DK according to specific directions in the wider DK space. In the following, we propose the concept of a "design knowledge map" (DK map; see Figure 3), which allows us to allocate "design knowledge chunks" (DK chunks) as well as plan, coordinate, and communicate "design knowledge journeys" (DK journeys).

Based on our conceptualization of DK, DSR projects can be viewed as contributions to "journeys" through the DK space. Each project can be understood as contributing a well-defined "chunk" of DK. In analogy to method engineering (Ralyte, 2004), we describe a chunk of reusable DK as a component that has both process character (reproducible design activities) and outcome character (a justified claim that links a certain solution space to a certain problem space via evaluation). We call it a "chunk" to express that this DSR project (and DK, respectively) is making a partial contribution toward potentially multiple future design projects in the course of a design journey. A design journey, in turn, is a set of DK activities that transforms DK from one state to another. It is therefore a process (vom Brocke & Rosemann, 2015) that is referred to as a "journey" in order to emphasize DK activities that span multiple DSR projects.

Every DSR project has a starting point that is grounded on existing DK, i.e., on one or more relationships between specific solution spaces and specific problem spaces. The DSR project then creates new DK by linking the same problems to a different (or changed) solution space, the same solution to a different (or changed) problem space, or by conducting a different evaluation of DK that corresponds to the same problem and solution space. Based on the DK model, a design journey can take at least three routes from each point of departure, which we define as dimensions conceptualizing DK in DK space: namely, *projectability*, *fitness*, and *confidence*. Each of these dimensions can exist at different levels, which, for the purposes of exposition, we designate simply as high, medium, or low, as illustrated in Figure 3.

Projectability (in the Problem Space)	low	medium	high
Fitness (in the Solution Space)	low	medium	high
Confidence (in the Evaluation)	low	medium	high

Figure 3. DK Space: Three Dimensions of Position Design Knowledge in Design Knowledge Space

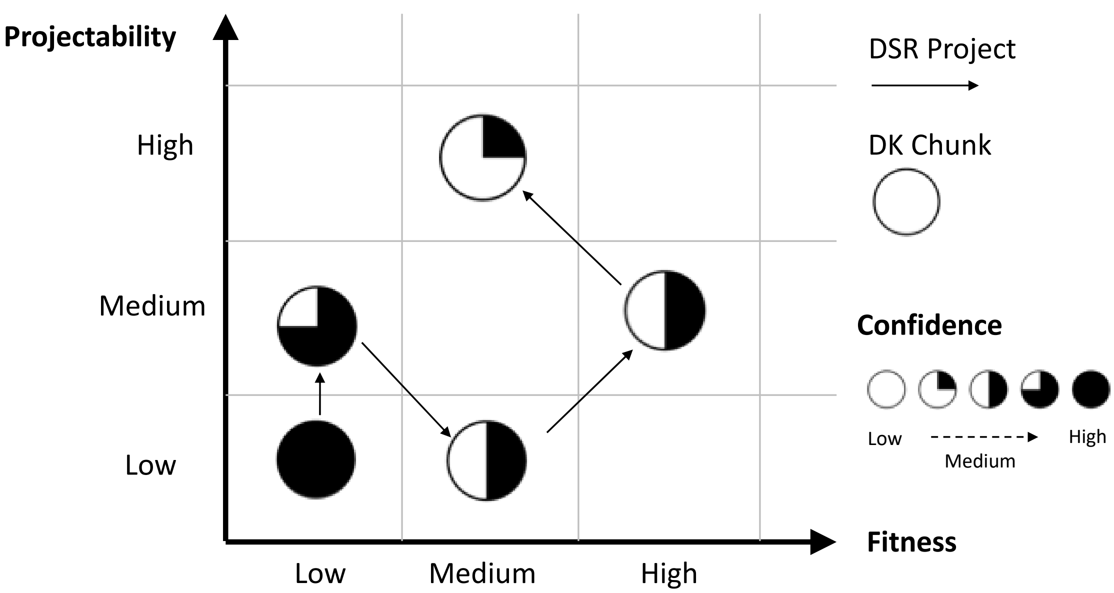

Figure 4. Example of a DK Map Representing a Journey of Design Knowledge Creation Through a Series of DSR Projects in Design Knowledge Space

Future research could develop more detailed and rigorous scales for each of these dimensions. Naturally, one would proceed on each of these routes seeking to increase projectability, fitness, and/or confidence. However, a DK journey can travel in various directions and combinations of directions, including backwards, as we discuss below. The three dimensions can be used in order to create directional representations of projectability, fitness, and confidence that serve as DK maps. To illustrate, Figure 4 presents a 3x3 matrix that spans the DK space using the two dimensions of "projectability" and "fitness." Evaluation "confidence," as the third dimension is represented using Harvey balls. Future research could improve DK maps by choosing different combinations of the DK dimensions.

The DSR journey illustrated in Figure 4 starts with DK of low projectability, low fitness, and high confidence (lower-left corner of the matrix)—in this case, a draft description of a "goods received" process based on essential process characteristics such as input, output, and stakeholders. In our hypothetical illustration, this DK has proven useful in an instantiated problem area of a retail company, e.g., onboarding new staff in order to inform them about the process developed in over 50 applications over the past three years. Then, the next project in a series of DSR projects seeks to generalize the existing DK so that it relates to a broader problem space. For instance, the process description should not only be perceived as useful for the specific company to which it directly applies but also for a wider range of retail companies. Increasing projectability in this example comes at the cost of reducing the confidence of evaluation, since—in course of the new project—three informal evaluations were carried out in other retail companies, while more formal evaluations only took place in the initial company.

In terms of this example, the third DSR project seeks to increase the fitness of the DK by adding a more detailed description of the timely logical flow of the process in the form of a semistructured process model. This increase in fitness then leads to a decrease in projectability, since the process flow was designed based on data drawn from one of the many companies in the relevant problem space, which thus reduces the level of confidence. In the fourth project, both projectability and fitness are increased by implementing an improved workflow to execute the process in multiple new organizations. The improved workflow supports the ability to customize different variants of the process. The fifth project studies in greater detail the workflow engine by analyzing process log data and identifying process patterns. This final project produces DK, which provides a higher level of projectability but a lower level of fitness.

In more general terms, if a design solution is evaluated regarding broader goodness criteria or is applicable to a wider range of problems, as compared to existing DK (which only met more narrow goodness criteria or were applied to a narrower problem class), it relates to a broader problem space. In the matrix illustrated above, the DK chunk increases *projectability* with unchanged *fitness*. On the other hand, if a DSR project extends the level of detail for which the solution is developed by adding a modeling method or modeling tool, then fitness of the solution would increase. If the extended research evaluates a solution in another organizational context with no change to projectability and fitness, then it would seek to foster evaluation rigor and increase the *confidence* in the resulting DK. In the example above, DK was accumulated and evolved across five projects. Following the notion of DSR across projects, it is important to notice that each DSR project produces reusable DK of its own that other DSR projects can consume and use in order to further develop the DK in certain aspects. In this way, multiple, very different, and unforeseen DK journeys can evolve.

For complex problems or solutions, DK accumulation and evolution usually requires a progression across several DSR projects longitudinally. For instance, pioneering projects might start by producing situated DK with low reuse potential (e.g., understanding certain problem subclasses or envisioning certain situated solutions). Based on these early contributions, DK may eventually become more fit (e.g., by grounding solution components in kernel theories or specifying solutions more thoroughly) and more widely projectable (e.g., by covering a broader range of problems or by addressing additional goal-based requirements).

Using the DK map, it is important to notice that in our initial proposal no objective metric regarding any of the three DK dimensions exists. Therefore, the DK map is not intended to create a catalog or archive of DK chunks in absolute terms. Rather, the map seeks to support communicating DK in relative terms, i.e., to communicate from a specific starting point on the map how a DSR project aims to further develop (or has further developed) existing DK. Therefore, the DK map is intended to position a limited number of DSR projects and DK chunks in relation to one another in the relevant subset of the three-dimensional DK space. The DK map primarily serves as a conceptualization to provide terminology for expressing how different DK chunks relate to one another.

The smaller the subset of the DK map is, the more likely it is that people would share an understanding of a metric for the DK dimensions. In a later section, for example, we present a real-world example from the area of enterprise architecture management (EAM) in which we make use of established terminology to position DK chunks regarding projectability, fitness, and confidence. In some domains, well-established coding schemas exist—for example in medicine, where both diagnoses and treatments are specified using international standards. Most application areas in IS research, however, do not offer such standards; thus, even within one domain, any terminology would be restricted to a language group sharing this terminology. Initiatives to standardize terminology in IS domains may be helpful, but heretofore the nature of systems in terms of complexity and dynamics has limited the possibilities of standardization. Therefore, in using the DK map, it is important to understand that, in most cases, positioning DK chunks in terms of the dimensions is a subjective process, meaning that researchers should provide reasoning and evidence justifying their positioning strategies.

5 Design Knowledge Movements

The DK map identifies typical "movements" that articulate archetypical forms of DK accumulation and evolution across DSR projects, which we illustrate in Figure 5. In many cases, DSR projects may create problem-solution relationships that, compared to existing DK, increase and/or decrease multiple dimensions at the same time, impacting, for instance, both *projectability* and *fitness*, *projectability* and *confidence*, *fitness* and *confidence*, or *all three dimensions*. For example, the enhanced reference model transforms additional descriptive knowledge (e.g., from IS success theories) to address additional stakeholder requirements. Very often, however, the enhancement of one dimension comes at the expense of diminishing another dimension. Certain DSR projects may, for instance, enhance fitness at the expense of projectability or enhance projectability at the expense of fitness. Although this might appear undesirable at first sight, such projects may also constitute useful contributions in the context of the DK journey as a whole.

Identifying movement archetypes of DK accumulation and evolution, we focus on movements in the DK map that advance DK either in terms of the projectability or fitness of the DK. We identify and describe four interesting archetypes of DK accumulation and evolution below. However, we do not further investigate the role of evaluation, which might create further archetypes. We offer the four unlabeled arrows (in Figure 5) and the impacts of evaluation confidence in the movements for future research consideration.

- **Generalization:** Projectability is enhanced with constant fitness. This research broadens the targeted design problem class or covers more goodness criteria without decreasing the fitness of the solution DK. An example is the enhancement of a method that would, for instance, include the perspective of further stakeholders (e.g., applying different value systems in a performance measurement system, which originally only considered time, cost, and quality as a fixed value system), while maintaining a specific level of detail.

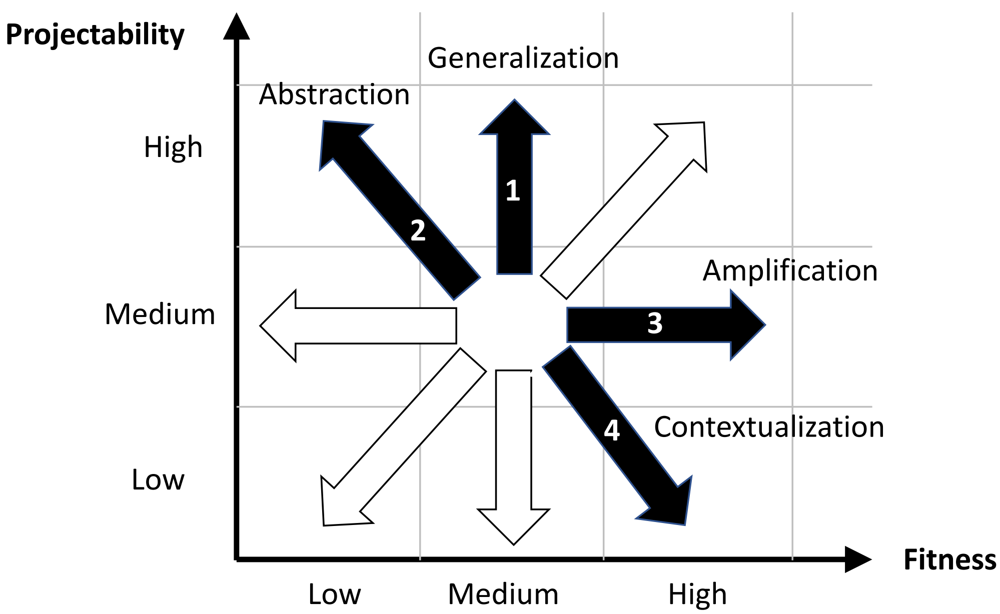

Figure 5. Four Movement Archetypes of DSR Projects

- **Abstraction:** Projectability is enhanced at the expense of fitness. While broadening the targeted design problem class or a coverage of more goodness criteria may increase an artifact's projectability, these enhancements may also lead to a decrease in fitness in that less detail is provided in order to fit a wider problem space. An example of this is an enhanced method that is more general (e.g., covering service lifecycle management instead of service delivery management) but less specific in its prescription (e.g., only naming and describing activities, instead of outlining substeps for each activity along with input and output objects). Abstraction may seek to identify the essence of a less fit design, which can then be projected onto a wider problem space.
- **Amplification:** Fitness is enhanced with constant projectability. Our research contributes to solving the problem or increases the level of detail of the solution design while the enhancements still cover the same problem space. An example is a more detailed reference model, which in addition to process models also provides data models or which, in addition to models only, also includes an application that customizes the processes.
- **Contextualization:** Fitness is enhanced at the expense of projectability. While additional justificatory foundations and/or adaptability enhancements increase an artifact's fitness, these enhancements can only be evaluated in limited or artificial environments. In this case, it may be that only a small subset of goodness criteria that are more controllable is studied. Thus, the DSR project might lead to lower projectability of DK. An example is an enhanced reference model for sales processes, which is enriched by configuration features and additional details but which, at this level of fitness, would be tied to a narrower context (e.g., instead of sales in general only sales through online channels).

In practice, movements in the DK map occur that make less (or more) than one advancement in one of the dimensions. DSR projects, which make more than one advancement, e.g., moving from mid-projectability and mid-fitness to high-projectability and high-fitness, can be (de)composed as a combination of generalization and amplification. DSR projects that diminish both dimensions may be the result of corrections or changes in the problem and solution spaces. For example, expanding the goals of a problem space to include security may require a rigorous evaluation of security issues (increasing confidence) in a limited problem context (decreasing projectability) for a specific part of the solution (decreasing fitness) for that new DK chunk. Thus, a decrease in more than one dimension can generate new value for DK as a whole.

While existing process models that rationalize DSR (e.g., Peffers et al., 2007) focus on the fitness dimension, its combination with the projectability dimension enhances the understanding of archetypical contributions, their combined movements along knowledge accumulation paths, and, eventually, the support of suitable DSR planning and steering activities both within and across projects. Our main intention, however, is not to support DSR process planning and steering, but to clarify how to position, locate, and, eventually, reuse DK. In particular, in complex DSR projects, long-running DSR projects, or those involving multiple researchers, it becomes crucial to be able to map and locate the various chunks of created DK in a structured way. Understanding the character of DK more completely can facilitate the identification of "related" chunks, coordinate parallel subprojects, plan and control complex design processes, and better communicate DK.

Particularly for complex design problems, DK accumulation follows a multistep "journey." In the following subsection we present the design of an enterprise architecture management (EAM) method to exemplify DK accumulation across projects and use the proposed map to represent the accumulation and evolution processes. We believe that using the proposed map can both improve the planning of specific design projects and foster knowledge evolution and accumulation across projects.

6 Design Knowledge Accumulation: An Illustrative Example

To illustrate DK accumulation, we describe the development of a situational IS management method. Winter (2012) summarizes the method's multi-project development process in a domain-independent form. The overall artifact design idea in this DK accumulation example is to (1) empirically identify existing design factors and solution clusters in a certain IS management domain, (2) empirically identify "ideal" solution clusters that promise to meet observed performance requirements in that domain, (3) use a capability-based comparison of ideal vs. existing solution clusters to derive transformation paths in that domain, and finally (4) design a configuration model that is able to compose all relevant transformation paths from a minimal set of capability clusters.

Since the space limitations of research papers prohibit a comprehensive description of the method's application, we refer to three publications that each focus on a different domain and document the method's components (understood as design knowledge chunks):

- Identification of design factors and as-is solution clusters in the study by Aier, Gleichauf, and Winter (2011), applied to enterprise architecture management (EAM)
- Identification of to-be solution clusters and transition paths in the study by Cleven, Winter, and Wortmann (2011), applied to process performance management (PPM)
- Design of method components and method configuration rules in the study by Bucher and Winter (2010), applied to business process management (BPM)

Figure 7 illustrates the knowledge accumulation path across DSR projects in different domains as it would be mapped according to the DK model. Characters denote the creation of different DK chunks (e.g., as-is solution clusters, transition paths, method configuration rules) and indexes indicate the domains for which knowledge was created (e.g., EAM, PPM, BPM).

As a whole, this multi-project development process was initiated by the (exaptation) idea that the body of knowledge in situational method engineering could and should be applied not only to software systems development, but also to the development of management methods (Design Project A). Among others, EAM, PPM, and BPM were identified as application domains because of access to a sufficient amount of empirical data. As a consequence, the DK map's projectability dimension can be instantiated as "company and domain specific" (low), "domain specific" (medium), and "cross domain" (high). Since the goal of the process is to design situated management methods, the map's fitness dimension can be instantiated to "understand existing practices" (and their configuration rules and their performance; low), "master method configuration" (for to-be solution design; medium) and "fulfill all objectives" (for the management method; high).

In this multi-project development process, Projects B1 (for EAM, documented in Aier et al., 2011), B2 (for PPM), and B3 (for BPM) identified design factors and as-is solution clusters in the respective domain. Project C then generalized the results into an as-is management analysis method.

In Project D, different conceptual options for identifying to-be solution clusters and transition paths were identified, ranging from success theories and surveys to maturity models. Based on the identified approaches, projects E1 (documented in Winter, 2012), E2 (documented in Cleven et al., 2011), and E3 (documented in Bucher & Winter, 2010) created method configurators for the EAM, PPM and BPM, respectively. While E2 only identified to-be solution clusters and transition paths, E1 and E3 also developed method components and configuration rules.

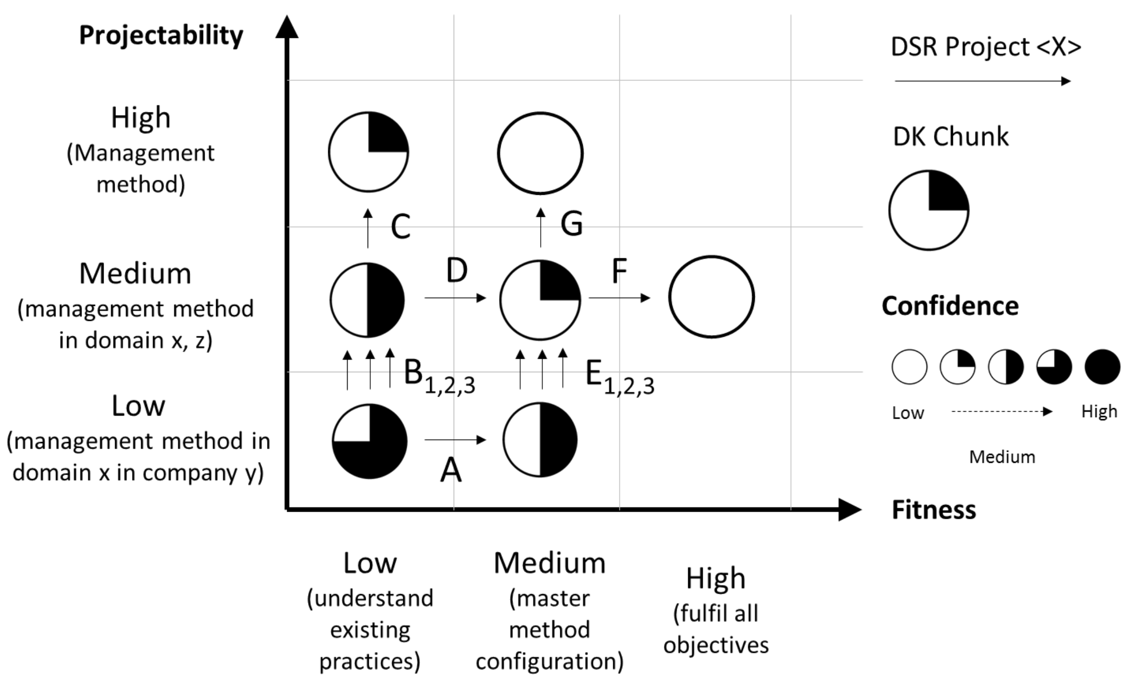

Figure 7. Design Knowledge Accumulation Path for EAM Method Design

The (still ongoing) Project F seeks to test situated EAM method interventions in real-world settings, thereby producing a proof of use of the management method configurator design in one domain (medium projectability). Project G generalizes experience from all three covered IS management domains into a generic management method design approach as documented in Winter (2012).

These illustrative examples not only show that complex but monolithic DSR projects (e.g., those documented in a PhD thesis or in a large system development project) can be mapped with the proposed model, but also multi-project design processes within a researcher group or even across researcher groups. For every DK chunk, it is possible to position input DK, output DK, and research contribution. Thus, the creation and growing maturity of DK can be better comprehended and relevant DK (e.g., on more or less contextualized levels) can be more easily identified.

7 Design Knowledge Principles

Reflecting on potential uses of the DK map, we propose a set of principles that facilitate knowledge accumulation and evolution in DSR projects. Each DSR project and its research contributions need to be specified regarding its problem space and solution space and the maturity of existing knowledge in both spaces (Gregor & Hevner, 2013). The following principles will help researchers in planning, conducting, and communicating their research accordingly.

1. **Positioning:** Each DSR project needs to clearly state which subsets of the problem and solution spaces it contributes to. More specifically, this means that (1) the relevant problem is identified, (2) the solution is investigated or created, and (3) the evaluation evidence is convincing in how the problem and solution relate. The DK model proposed in this editorial provides a template for describing the relevant subsets of problem and solution spaces. This necessitates a clear statement of the problem, complete with context and goodness criteria, the essence of the solution with artifact representations and design processes, and techniques and results of the evaluation. In a DSR project, positioning is a continuous task that helps shape and reshape the understanding and identification of the DK to be created. In a final paper documenting research at a certain stage, positioning is typically presented in the introduction of the paper and is further elaborated in the background section.
2. **Grounding:** Each DSR project must be transparent regarding the extent to which it builds on prior knowledge. Both processes and results of the search for related extant contributions should be reported. The model for producing and consuming DK that is presented in this article provides a suitable structure supporting the search for and documentation of extant knowledge. Specifically, both propositional knowledge, and prescriptive knowledge in the form of design theory or design entities must be investigated. Literature reviews could be leveraged to perform and document the search (vom Brocke et al., 2015; Webster & Watson, 2002). Also, a meta-analysis could be conducted that draws together the results of multiple DSR studies (e.g., Denyer, Tranfield, & van Aken, 2008). Grounding produces a form of DK that is particularly important to account for knowledge accumulation and evolution because it identifies extant knowledge that informs specific DSR projects. Specifically, it identifies knowledge already available to address the problem. The rigor with which a search is conducted determines the confidence of the grounding and is therefore an important quality criteria for DSR.
3. **Aligning:** Each DSR project should be transparent about how its design processes evolved. DSR projects can build on methodological guidance regarding how to structure DSR processes (e.g., Gregor & Hevner, 2013; Peffers et al., 2007), but individual DSR projects often deviate from the conceptual reference structure because they must contend with specific constraints and seize opportunities to adjust knowledge progression throughout the process (Sonnenberg & vom Brocke, 2012; Abraham et al., 2014). Important quality criteria for DSR, then, are (1) the transparency with which the design process is documented, as well as (2) the assurance created in the appropriateness of the process, typically established by arguing for the rationale of the design process. The DK map can help plan and document design processes over multiple design activities. For instance, in terms of instantiation validity (Lukyanenko et al., 2014), alignment seeks to provide clear evidence that an instantiation, created to demonstrate or evaluate design principles, would actually fall within the same subset of problem and solution spaces as the principles it seeks to instantiate; thus demonstrating projectability.
4. **Advancing**: Completed DSR research should clearly state how it advances prior design knowledge (Hevner et al., 2004). In the absence of a conceptualization to position (and compare) contributions in problem and solution spaces, however, statements on such advancement lack a frame of reference and thus are difficult to formulate and comprehend. The DK map in this

editorial provides a conceptualization that expresses specific advancements a DSR project makes. Using the dimensions of *projectability*, *fitness*, and *confidence*, researchers may argue how a DK chunk provided through a DSR project builds upon and adds to extant DK. A dedicated discussion section, as suggested by Gregor and Hevner (2013), allows researchers to argue that the DK contributed in a specific DSR project "engaged with existing DK," similar to the way in which contributions are discussed in grounded theory research. A graphical representation of the DK map can help explicate extant knowledge, the contribution a paper makes, and avenues for future research.

We deem it important to adhere to these four principles in order to allow for individual research projects to contribute to knowledge accumulation and evolution. The extent to which a paper succeeds in demonstrating the principles is an important metric that can be used to measure the scientific rigor of such research.

8 The Special Issue Papers

In this *JAIS* special issue, we worked closely with the author teams to showcase the potential of the suggested principles for enhancing design knowledge accumulation and evolution. In the following, we briefly outline each article. The Appendix gives detailed information about each paper, demonstrating the principles of positioning, grounding, aligning, and advancing, where applicable.

"Monitoring the Complexity of IT Architectures: Design Principles and an IT Artifact" by Thomas Widjaja and Robert Wayne Gregory aims at providing IT support for IT architects who need to monitor the structural and dynamic complexity of a firm's IT architecture in the context of digital business strategy. In the form of design principles inferred by heuristic theorizing, the design knowledge they propose is accumulated over three cycles and several iterations with five large companies over eight years. During the evolution of this design knowledge, both fitness and confidence (more cases, more evaluative evidence) are increased in parallel. Also, the understanding of the problem improves by moving from a standardization focus via a heterogeneity focus to a complexity focus (the most comprehensive form).

"Accumulating Design Knowledge with Reference Models: Insights from Twelve Years of Research on Data Management" by Christine Legner, Tobias Pentek and Boris Otto addresses the problem of managing data as a strategic resource in global corporations. They base their article on data from a 12-year research program involving practitioners from more than 30 enterprises and more than 15 researchers from three universities. Based on these data, they report on mechanisms of accumulating design knowledge over time. In particular, they investigate the use of reference models as a specific form of representing design knowledge in order to support DK accumulation processes.

"A Design Theory for Visual Inquiry Tools" by Hazbi Avdiji, Dina Elikan, Stéphanie Missonier, and Yves Pigneur delivers design knowledge in the form of a design theory for visual inquiry tools for strategic management. The design theory is based on a theorizing process for three existing visual inquiry tools that are developed in dedicated design science research projects and tested in the real world with practitioners. The authors perform a within- and cross-projects analysis of the three DSR projects and generalize the project-specific design knowledge into 12 design principles to guide the design of visual inquiry tools.

"Accumulating Design Knowledge: A Mechanisms-Based Approach" by Ana Paula Barquet, Lauri Wessel, and Hannes Rothe investigates the mechanisms that explain design knowledge creation (KC) in DSR projects and show how these mechanisms impact design knowledge accumulation over time and across projects. They perform two studies: The first is an in-depth case study of three DSR projects (industry-academic) that is supervised by the author team and identifies three KC mechanisms—injection, folding, and enhancement. The second is a meta-analysis of two longitudinal DSR projects that tracks the use of these mechanisms over time. Since this paper does not focus on a specific DSR project, no application of DK principles is included in our Appendix.

9 Research Implications

The DK model, map, and the principles proposed in this editorial contribute to knowledge accumulation and evolution in DSR in a number of ways, serving researchers, readers, editors, and reviewers:

Researchers: The principles of positioning, grounding, aligning, and advancing help researchers plan, conduct, and document their research. In the planning phase, researchers can more systematically carve out and identify clusters in the problem and solution spaces to which they intend to contribute and plan strategies for building intermediate contributions, aligning them with an intended new knowledge contribution. When conducting research, reuse of DK from neighboring areas in the problem and solution spaces is supported, fostering both the effectiveness and efficiency of design from a researcher's perspective and the consistency of the knowledge base from a community perspective. In publishing their work, these principles will allow researchers to more precisely report the subsets of the problem and solution spaces to which their contributions relate and provide reasoning for each design activity. This will reduce ambiguity and improve communication with reviewers, editors, and readers.

Readers: Explicit positioning of each DSR contribution supports readers in grasping the key novel contributions of individual papers and the logic of research designs spanning multiple intermediate contributions because the boundaries of both intermediate contributions and new contributions are more clearly described in relation to other contributions. This also supports DSR collaboration in that readers can draw connections to intermediate contributions and consider extensions of extant research on other projects—for example, by taking given problem specifications and designing and evaluating new solutions to such problems. If authors are better able to specifically position their contributions and relate them to extant contributions, readers will also be supported in locating related contributions, which will increase their ability to assess the body of DK available in specific areas both more efficiently and more effectively.

Editors and reviewers: Clearly positioning and relating publications in the problem and solution spaces enables editors and reviewers to more rigorously evaluate the novel contributions of single papers. Beyond assessing the contributions a paper makes to the field, by more specifically explicating the type of contribution, editors can also better evaluate the fit of submitted DSR manuscripts in terms of journal profiles, e.g., those that privilege theoretical over applied contributions, or exploratory over confirmatory papers. The explicit alignment of intermediate contributions constituting the overall design process can enable reviewers to better evaluate the rigor of research processes. The terminology provided in this editorial also supports communication with authors when discussing the contribution of a paper and the rigor of the research process, which should eventually support the publication of more significant papers in faster publication cycles and create positive effects for the impact and recognition of DSR for contributing to real-world problem solving.

Community as a whole: The ideas contained in this editorial increase the accessibility of design knowledge for analysis and reuse, and will hopefully encourage high-quality, impactful DSR in the community as a whole. From a community perspective, both higher coherence and efficiency (e.g., by avoiding redundancies) of DK can be achieved, fostering the impact that DSR can have in contributing to real-world challenges. Furthermore, a collaborative effort of designing for reuse and designing by reuse (vom Brocke, 2007) may also facilitate increasing certainty about common and shared scales for projectability, fitness, and evaluation confidence, at least within specific domains. Such shared knowledge extends beyond utility for single researchers or research projects and would be beneficial for the entire research community.

10 Future Research

This editorial provides a first step toward establishing a methodological foundation for the systematic accumulation and evolution of design knowledge (DK) created by DSR projects. We explore several questions that remain unresolved, opening up an interesting space for future research.

First, finding the right way to describe and link new DK with existing DK is challenging across multiple projects. It is often difficult to assess the reuse potential of existing design solutions for different but related problems. Thus, there is a need for future research that systematically describes and classifies problems and embeds them into existing problem space. It may be necessary to decompose higher-level problems into lower-level problems and create corresponding classifications. The same issues are also relevant for solutions and solution space. There is a need to further investigate representation languages and repositories that can help structure and classify problems and solutions in DSR. The challenge of creating problem and solution descriptions and classifications may also be approached with data-driven approaches that support this nontrivial process.

Second, there is a need to provide more detailed and elaborated methodological guidance for planning iterations in DSR. This requires, as mentioned above, the definition of new types of quality criteria that may inform the iteration decision. Besides providing decision support for iterating in design knowledge creation, there is also a need to govern design knowledge creation and sharing across multiple design research projects. Governance in this context refers to choosing structures and mechanisms that can influence the processes of creating and sharing knowledge.

Third, it is rather unrealistic to create a comprehensive and holistic design knowledge base that basically addresses all types of problems and solutions. Rather, we believe that design knowledge will be created by different actors or actor groups. Individuals may create design knowledge as part of specific conference paper or an entire PhD thesis project. DK may be created within a research grant or a research group, or multiple researchers can collaborate and jointly create DK. Beyond this, one should consider DK creation on a broader level of communities and subcommunities. For example, the business process management (BPM) community may build its own design knowledge base, with specific subcommunity perspectives (e.g., a "process mining" design knowledge base). A key question is how DK created by different actors and actor groups should be accumulated, reused, and better connected. Similarly, the packaging of DK for the purpose of increasing knowledge sharing should be further explored. Sharing knowledge is, in principle, always desirable, but it is unclear who should share what with whom and for what purpose. Further, the issue of proprietary DK creates important access constraints on DK that researchers may find challenging for grounding and extending research.

Fourth, future research should explore the potentials of tool-support in furthering knowledge accumulation and evolution in DSR. Researchers have only recently started to investigate the requirements for tool support in DSR (Morana et al., 2018; vom Brocke et al., 2017), and tools have been developed to document individual design processes (e.g., vom Brocke et al., 2017). Using DSR tools will support integrating DSR processes across projects (e.g., by finding or being recommended to related DK contributions). Also, as tools mature for planning, conducting, and documenting DSR, more data will be available to generate insight into the types and semantics of DSR projects. Text mining, for instance, has been applied in IS research to automatically analyze the semantics in large amounts of text (e.g., Müller, Junglas, vom Brocke, & Debortoli, 2016; Müller, Junglas, Debortoli, & vom Brocke, 2016) and to create and maintain taxonomies for different application contexts (Debortoli, Müller, & vom Brocke, 2014; Debortoli, Junglas, Müller, & vom Brocke 2016; Schmiedel, Müller, & vom Brocke, 2019). The more the community makes use of such tools and the more it adopts open data principles, the more the community will be able to learn from single DSR projects to derive further conceptualizations supportive of knowledge accumulation and evolution.

In general, the IS community should consider building on best practices from other communities when it comes to systematically building design knowledge. One example is the machine learning community in computer science. Here, specific and narrowly defined problems are articulated and approached. For example, high-quality labeled data sets (structured data, image, text, sound, etc.) are established as foundations to develop algorithms for classification problems and are provided as a reference. On this basis, the quality of the proposed algorithms can be systematically compared.

In medicine, the importance of medical knowledge representation is driven by the societal relevance and accumulation of good medical care over many decades. Building on structured languages (e.g., the Unified Medical Language System https://www.nlm.nih.gov/research/umls/), this community has established systematic ways of accumulating and reusing medical knowledge. Establishing a common metadata schema-building system that captures the structural elements of design knowledge with and across DSR projects introduced in this editorial may be an important next step for the DSR community.

11 Conclusion

In this editorial, we make contributions to conceptualize design knowledge (DK) models, maps, and guidelines for knowledge accumulation and evolution in DSR. A proposed model of DSR project-based DK is presented in Figure 1, clearly defining DK as knowledge in the relationship between problem and solution spaces with a certain confidence of evaluation. We define specific components of DK that are specifically relevant in order to document and communicate DK. This is followed by a model (Figure 2) of how a DSR project consumes and contributes to the cumulative design knowledge bases of descriptive (Ω) and prescriptive (λ) knowledge that defines six specific modes of knowledge production and consumption in DSR projects, which further support articulating the significant contribution of a project. Next, a DK map (Figure 3) with the dimensions of projectability, fitness, and confidence provides a conceptual foundation specifying which subsets of the broader problem and solution space a DSR contribution relates to, and, in turn, allows for relating individual contributions to one another. In essence, the map serves as a navigator, allowing a DSR project journey to walk the problem and solution spaces across DSR projects and DSR activities, respectively. We value this as an important prerequisite for knowledge accumulation and evolution in DSR. Based on the DK map, we identify four archetypes of DK accumulation and evolution—generalization, abstraction, amplification, and contextualization—that represent typical DK movements through problem and solution spaces. We then present four guidelines on how to apply our contributions in DSR studies, through positioning, grounding, aligning, and advancing. We outline how each of the papers published in this special issue can be characterized according to the guidelines using the conceptualization for DK accumulation and evolution presented in this article. Finally, we draw implications for researchers, readers, editors, and reviewers, as well as the community as a whole, and present future research opportunities.

References

Abraham, R., Aier, S., & Winter, R. (2014). Fail early, fail often: Towards coherent feedback loops in design science research evaluation. *Proceedings of the 35th International Conference on Information Systems*.

Aier, S., & Fischer, C. (2011). Criteria of progress for information systems design theories. *Information Systems and E-Business Management*, *9(1)*, 133-172.

Aier, S., Gleichauf, B., & Winter, R. (2011). Understanding enterprise architecture management design: An empirical analysis. *Proceedings of the 10th International Conference on Wirtschaftsinformatik*.

Baskerville, R. L., & Pries-Heje, J. (2014). Design theory projectability. In B. Doolin, E. Lamprou, N. Mitev, & L. McLeod (Eds.), *Information systems and global assemblages: (Re)configuring actors, artefacts, organizations* (pp. 233-245). Springer.

Baskerville, Richard and Pries-Heje, Jan (2019). Projectability in design science research. *Journal of Information Technology Theory and Application*, *20*(1), Article 3.

Baskerville, R. L., Kaul, M., & Storey, V. C. (2015). Genres of inquiry in design-science research: Justification and evaluation of knowledge production. *MIS Quarterly*, *39(3)*, 541-564.

Baskerville, R. L., 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.

Becker, J., vom Brocke, J., Heddier, M., & Seidel, S. (2015). In search of information systems (grand) challenges: A community of inquiries perspective. *Business & Information Systems Engineering*, *57(6)*, 377-390.

Bichler, M., Heinzl, A., & Winter, R. (2015). Practice impact of IS research. *Business & Information Systems Engineering*, *57(2)*, 87-89.

Bucher, T., & Winter, R. (2010). Taxonomy of business process management approaches: An empirical foundation for the engineering of situational methods to support BPM. In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 2* (pp. 93-114). Springer.

Chandra, L., Seidel, S., & Gregor, S. (2015). Prescriptive knowledge in IS research: Conceptualizing design principles in terms of materiality, action, and boundary conditions. *Proceedings of the 48th Hawaii Conference in System Science*.

Cleven, A., Winter, R., & Wortmann, F. (2011). Process performance management: Illuminating design issues through a systematic problem analysis. *Proceedings of the 2011ACM Symposium in Applied Computing*.

Debortoli, S., Junglas, I., Müller, O., & vom Brocke, J. (2016). Text mining for information systems researchers: An annotated topic modeling tutorial. *Communications of the Association for Information Systems*, *39(7)*, 110-135.

Debortoli, S., Müller, O., & vom Brocke, J. (2014). Comparing business intelligence and big data skills: A text mining study using job advertisements. *Business & Information Systems Engineering*, *6(5)*, 289-300.

Drechsler, A., & Hevner, A. R. (2018). Utilizing, producing, and contributing design knowledge in DSR projects. In S. Chatterjee, K. Dutta, & R. Sundarraj (Eds.), *Lecture Notes in Computer Science* (pp. 82-97). Springer.

Denyer, D., Tranfield, D., & van Aken, J. E. (2008). Developing design propositions through research synthesis. *Organization Studies*, *29(3)*, 393-413.

Gill, T. G., & Hevner, A. (2013). A fitness-utility model for design science research. *ACM Transactions on Management Information Systems*, *4(2)*, 1-24.

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

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

Grover, V., & Lyytinen, K. (2015). New state of play in information systems research: The push to the edges. *MIS Quarterly*, *39(2)*, 271-296.

Gupta, A. (2017). Editorial thoughts: What and how ISR publishes. *Information Systems Research*, *28(1)*, 1-4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems. *MIS Quarterly*, *28(1)*, 75-105.

Lee, J. K. (2015). Research framework for AIS grand vision of the bright ICT initiative. *MIS Quarterly*, *39(2)*, iii-xii.

Lukyanenko, R., Evermann, J., & Parsons, J. (2014). Instantiation validity in IS design research. In M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta, & V. Yoon (Eds.), *Advancing the impact of design science: moving from theory to practice* (pp. 321-328). Springer.

Morana, S., vom Brocke, J., Maedche, A., Seidel, S., Adam, M. T. P., Bub, U., …, & Winter, R. (2018). Tool support for design science research—Towards a software ecosystem: A report from a DESRIST 2017 Workshop. *Communications of the Association for Information Systems, 43*(17), 237-256.

Müller, O., Junglas, I., vom Brocke, J., & Debortoli, S. (2016). Utilizing big data analytics for information systems research: Challenges, promises and guidelines. *European Journal of Information Systems, 25*(4), 289-302.

Müller, O., Junglas, I., Debortoli, S., & vom Brocke, J. (2016). Using text analytics to derive customer service management benefits from unstructured data. *Management Information Systems Quarterly Executive, 15*(4), 243-258.

Newton, I. (1676). *Letter from Sir Isaac Newton to Robert Hooke*. Historical Society of Pennsylvania.

Nunamaker Jr, J. F., Twyman, N. W., Giboney, J. S., & Briggs, R. O. (2017). Creating high-value real-world impact through systematic programs of research. *MIS Quarterly*, *41*(2), 335-351.

Peffers, 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.

Prat, N., Comyn-Wattiau, I., & Akoka, J. (2015). A taxonomy of evaluation methods for information systems artifacts. *Journal of Management Information Systems*, *32*(3), 229-267.

Rai, A. (2017). Diversity of design science research. *MIS Quarterly*, *41*(1), iii-xviii.

Ralyte, J. (2004). Towards situational methods for information systems development: Engineering reusable method chunks. *Proceedings of the 13th International Conference on Information Systems Development.*

Schmiedel, T., Müller, O., & vom Brocke, J. (2019). Topic modeling as a strategy of inquiry in organizational research: A tutorial with an application example on organizational culture. *Organizational Research Methods*, *22*(4), 941-968.

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

Sonnenberg, C., & vom Brocke, J. (2012). Evaluations in the science of the artificial - Reconsidering the build-evaluate pattern in design science research. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Advances in Theory and Practice* (pp. 381-397). Springer.

Venable, J. R. (2006). The role of theory and theorising in design science research. *Proceedings of the 1st Conference on Design Science Research in Information Systems and Technology*.

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2016). FEDS: A framework for evaluation in design science research. *European Journal Information Systems*, *25(1)*, 77-89.

vom Brocke, J. (2007). Design principles for reference modelling: Reusing information models by means of aggregation, specialisation, instantiation, and analogy. In P. Fettke & P. Loos (Eds.), *Reference Modelling for Business Systems Analysis for Business Systems Analytics* (pp. 47-76). Idea Group Publishing.

vom Brocke, J., & Buddendick, C. (2006). Reusable conceptual models: Requirements based on the design science research paradigm. *Proceedings of the 1st Conference on Design Science Research in Information Systems and Technology*.

vom Brocke, J., Watson, R. T., Dwyer, C., Elliot, S., & Melville, N. (2013). Green information systems: Directives for the IS discipline. *Communications of the Association for Information Systems*, *33*(30), 509-520.

vom Brocke, J., & Rosemann, M. (2015). Business process management. *Wiley Encyclopedia of Management Information Systems*, *7*. https://doi.org/10.1002/9781118785317.weom070213

vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., & Cleven, A. (2015). Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. *Communications of the Association for Information Systems*, 37*(9)*, 205-224.

vom Brocke, J., Fettke, P., Gau, M., Houy, C., Maedche, A., Morana, S., & Seidel, S. (2017). Tool-support for design science research: Design principles and instantiation. *SSRN Electronic Journal*, 10.2139/ssrn.2972803.

Watson, R. T., Boudreau, M.-C., & Chen, A. J. (2010). Information systems and environmentally sustainable development: Energy informatics and new directions for the IS community. *MIS Quarterly*, *34*(1), 23-38.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly, 26*(2), xiii-xxiii.

Winter, R. (2012). Construction of situational information systems management methods. *International Journal of Information System Modeling and Design*, *3*(4), 67-85.

Winter, R., & Albani, A. (2013). Restructuring the design science research knowledge base: A one-cycle view of design science research and its consequences for understanding organizational design problems. In R. Winter, & A. Albani (Eds.), *Designing organizational systems: An interdisciplinary discourse* (pp. 381-397). Springer.

Appendix A

Paper – Monitoring the Complexity of IT Architectures: Design Principles and an IT Artifact

1. Positioning: What problem is addressed through which solution to what confidence?

Problem: State and characterize the problem.

- **Problem statement:** How can IT support be provided for reducing the problem-solving complexity of monitoring the structural and dynamic complexity of IT architectures?
- **Context description:** Management of historically grown IT architectural complexity in large established organizations pursuing a digital business strategy.
- **Goodness description:** Usefulness of design principles and implemented IT tools for monitoring structural and dynamic IT architectural complexity as perceived by IT architects and IT management.

Solution: Outline and characterize the solution.

- **Solution essence:** When used for developing IT tools for monitoring IT architectural complexity, the four proposed design principles help to measure, visualize and analyze those aspects of structural and dynamic IT architectural complexity that help IT architects to reduce problem-solving complexity.
- **Representation description:** Inferring design principles (as a nascent design theory) from multiple monitoring tool development and application projects carried out across five large companies and eight years.
- **Process description:** Heuristic theorizing involving three cycles and several iterations comprising abstract artifacts (design principles) and artifact instantiations (IT architectural complexity monitoring tools used in companies).

Evaluation: Describe evaluation activities and results.

- **Method:** Concurrent evaluation of instantiated tools involving feedback from IT architects in five companies.
- **Results:** After evaluations with unsatisfactory results in early iterations, the evaluation in the final iteration provided sufficient evidence for the usefulness of the instantiated tools.

2. Grounding: What knowledge is informing the design?

Search Process: Which search strategy was applied?

- Review of relevant domain literature regarding the focused problem class in each heuristic theorizing cycle. Evaluation results were used to determine the problem class and related literature on which to focus in later iterations. In the final cycle the boundaries of the search were expanded to general problem class literature (i.e., monitoring complexity of systems-of-systems literature).

Search Results:

- **Kernel theory (Ω-knowledge):** Theory on complex systems and IT complexity (no integral theory yet, structural and dynamic models); the focus on this kernel theory was preceded by a focus on knowledge about IT standardization and IT heterogeneity in earlier cycles.
- **Design knowledge (λ-knowledge):** Process—heuristic theorizing framework; entities—conceptual models for IT architecture (including mathematical models and simulation models); design and configuration of IT tools; existing tools for IT architecture modeling and analysis.

3. Aligning: How do the design activities contribute to creating the DK?

Design process documentation: What activities were conducted in which sequence?

- Switching between problem structuring and artifact design within each heuristic theorizing cycle; switching between design principle (projectable) and model/tool prototype (instantiation) levels within each heuristic theorizing cycle; explorative journey from standardization via heterogeneity to complexity focus across three major heuristic theorizing cycles.

Design process rationale: Why were the activities conducted in this sequence?

- Each iteration allowed the team to develop a more complete understanding of the problem and the development of more comprehensive solutions (nested problem structure).

4. Advancing: How does the DK chunk provided compare to existing DK?

- In the course of developing and evaluating five tools for five companies over eight years, design principles were formulated and revised incrementally (see Figure 4), increasing both fitness and confidence (more cases, more evaluative evidence) in parallel. Also, the understanding of the problem improved (see Figure 2) by moving from a standardization focus via a heterogeneity focus to the (most comprehensive) complexity focus.

Appendix B

Paper – Accumulating Design Knowledge with Reference Models: Insights from Twelve Years of Research on Data Management

1. Positioning: What problem is addressed through which solution to what confidence?

Problem: State and characterize the problem.

- **Problem statement:** How can data be managed as a strategic resource in global corporations?
- **Context description:** Global corporations typically have complex organizational structures and distributed operations, resulting in data silos and a lack of transparency concerning the data resources.
- **Goodness description:** Relevant evaluation criteria are the reference model's structure (i.e., the completeness, simplicity, clarity, style, homomorphism, level of detail, and consistency), the adaptability (i.e., robustness and learning capability), and the environmental fit (i.e., personal and organizational utility, understandability, and organizational fit).

Solution: Outline and characterize the solution.

- **Solution essence:** The data excellence model (DXM) is a reference model for data management. It builds on the understanding of data management as the organizational capability to deploy data resources that is contingent on business objectives.
- **Representation description:** The DXM comprises eleven design areas, which represent the main constituents (or domains) of data management. Each of the design areas is ontologically defined through the entities (or constructs) it addresses and through result documents that represent the outcomes of design activities. The constructs and their relationships are specified in the form of a metamodel, i.e., a conceptual data model of the domain intended to build the ontological foundation and to create a shared understanding among experts from academia and practice.
- **Process description:** The reference model was developed following consortium research in a longitudinal and multilateral research program involving practitioners from more than 30 enterprises and more than 15 researchers from three universities over 12 years. This research program develops design knowledge in the form of DSR artifacts, that are systematically consolidated and resulted in different versions of a reference model for data management. The artifact's structure and content evolved around the eleven design areas, starting with the design area's definition (setting the boundaries and defining the key objects) through refinement (analyzing and defining practices, results, and principles), extension (broadening the scope), and modification (improving/changing/ correcting).

Evaluation: Describe evaluation activities and results

- **Method**: We applied artificial (analytical and formal, questionnaire-based) evaluation methods as well as naturalistic methods through analyzing the reference model's adoption in practice.
- **Results**: From an analytical perspective, we could show that the artifact addressed the requirements by means of purposeful design decisions. In the formal, summative evaluation in a focus group with 25 experienced data managers, respondents confirmed that the reference model is useful for their data management activities (86%), that the reference model covers all relevant areas of data management (88%) and depicts the reality of data management (83%). They assessed it as robust enough to reflect future changes in the data management environment (80%). The relatively low scores relating to the visualization of the reference model—only 48% of the participants agreed that the style and design are appropriate—led to a redesign of the model's graphical shape, involving professional designers. Naturalistic evaluation confirmed the design areas' validity as well as the reference model's applicability and usefulness. Typical adoption scenarios can be categorized in (1) translating the abstract design knowledge into concrete situational designs (i.e., instantiation), and (2) using the reference model as abstract situational knowledge for communication, education, maturity assessment and benchmarking purposes (i.e., mobilization).

2. Grounding: What knowledge informs the design?

Search process: Which search strategy was applied?

- We applied consortium research that unfolds in four activity categories: analysis (exploration of the problem space, leading to problem identification and requirements definition), design (development of the solution space via the iterative design and development of artifacts), demonstration and evaluation (via expert evaluation and situational instantiations), and diffusion (presentation and publication of the research results, targeted at general and local practice, and the scientific community).

Search Results:

- **Kernel theory (Ω-knowledge):** The understanding of data management as an organizational resource builds on the conceptualization of data as an economic good and the resource-based view (RBV). As an interdisciplinary field, data management draws on concepts and theories from various disciplines, most importantly computer science (specifically databases and data analytics), information systems, and management.
- **Design knowledge (λ-knowledge):** Data management's design knowledge base is created in both the research and the practitioner communities. Reference models for data management synthesize descriptive and prescriptive knowledge in the form of conceptual, capability, and maturity models. In addition, implicit design knowledge is inherent in emerging (situational and generic) solution designs and artifacts.

3. Aligning: How do the design activities contribute to creating the DK?

Design process documentation: What activities were conducted in which sequence?

- Following consortium research, the reference model was developed and refined in iterative design processes that unfold in four activity categories: analysis (exploration of the problem space, leading to problem identification and requirements definition), design (development of the solution space via the iterative design and development of artifacts), demonstration and evaluation (via expert evaluation and situational instantiations), and diffusion (presentation and publication of the research results, targeted at general and local practice, and the scientific community). The reference model evolved in three phases: (1) framing the problem and creating a shared understanding about data management (*ontology*), (2) assessing maturity and building the required data management capabilities (*capability building*), and (3) addressing the growing data requirements of a digital and data-driven enterprise (*reorientation*).

Design process rational: Why were the activities conducted in this sequence?

- The consortium research method is a proven approach for developing design knowledge in a research-industry collaboration. It allows for addressing a general problem (conceived of as a problems class) through the design of artifacts and learning from situational inquiry and materialized instantiations.
- The different versions of the artifact represent design knowledge accumulation on data management in both practitioner and research communities. Knowledge accumulation occurred in stages as a result of maturing abstract and situational domain knowledge (solution space), and in response to the evolving roles of data (problem space).

4. Advancing: How does the DK chunk provided compare to existing DK?

- Phase 1 (ontology): The initial version of the reference model (alpha version) as a conceptual model with six design areas focused on building the ontological foundation and creating a shared understanding of data management among experts from academia and practice.
- Phase 2 (capability building): Artifact development in this phase was driven by practical experiences that companies only very slowly built their data management capabilities. As a capability and maturity model, the reference model details each of the six design areas and comprises, at its most detailed level, 30 practices and 56 measures.
- Phase 3 (reorientation): The artifact was revised and adapted to cope with the data resource's extended scope and strategic relevance. The beta version (i.e., DXM) extends the six design areas (modifies one of them) and introduces five new design areas.

Appendix C

Paper – A Design Theory for Visual Inquiry Tools

1. Positioning: What problem is addressed through which solution with what confidence?

Problem: For cross-boundary teams to address a strategic management problem (e.g., new product development, business modeling), they should follow a process of joint inquiry, i.e., a process through which team members discuss to (1) articulate and explore the problem, and (2) develop and evaluate alternative solutions. However, the complexity of the process requires a variety of material and discursive support.

- **Problem statement:** How can cross-boundary teams be supported in their process of joint inquiry for strategic management problems?
- **Context description:** Strategic management problems are increasingly addressed through iterative and nonlinear approaches, such as design thinking and joint inquiry, as these problems are typically uncertain, ill-defined, and complex.
- **Goodness description:** Usefulness of support for framing the strategic management problem and facilitating the development and evaluation of alternative solutions.

Solution: An extensive number of visual inquiry tools for a variety of strategic management problems have been developed to support the process of joint inquiry. As these developments have merely relied on the designers' intuitions or the imitation of existing tools (e.g., the business model canvas), we propose a set of 12 design principles to guide the design of visual inquiry tools.

- **Solution essence:** The 12 design principles are grouped under three broad aspects (i.e., framing the problem through conceptual modeling, facilitating communication between cross-boundary team members through shared visualization, and specifying directions for use for effective joint inquiry). The principles ensure that the three visual inquiry tools are thoroughly reflected on and designed.
- **Representation description:** The design principles were inferred from three existing visual inquiry tools which were developed through a design science research process and extensively adopted by practitioners.
- **Process description:** Within- and cross-project analysis of the development and artifacts of the three design science research projects, with a focus on the design requirements and design features.

Evaluation: No evaluation of the design theory. The three design science research projects involved an extensive number of evaluation activities for the visual inquiry tools (i.e., the artifacts on which the design theory is based).

2. Grounding: What knowledge informs the design?

Search process: The development of the design theory was achieved through a theorizing process in which the instance problems and solutions were identified (i.e., by focusing on each visual inquiry tool), and later expanded to the abstract problems and solutions to identify the similarities and differences in the problems the three visual inquiry tools address and through which design features.

Search Results:

- **Kernel theory (Ω-knowledge):** Literature review on the characteristics of strategic management problems (for the problem domain), joint inquiry and design thinking techniques (for the solution domain), and existing studies on visual inquiry tools (for the solution domain).
- **Design knowledge (λ-knowledge):** Identification and analysis of the design requirements (problem domain) and design features (solution domain) of the three visual inquiry tools and comparison with additional visual inquiry tools.

3. Aligning: How do the design activities contribute to creating the DK

Design process documentation: (1) identification of the instance design requirements (i.e., the problem class that each visual inquiry tool addresses), (2) identification of the instance design features of each visual inquiry tool, (3) analysis of the design knowledge accumulated within each project, (4) abstraction of the design knowledge through a cross-project analysis of the instance design features and design requirements, (5) formalization into the design theory.

Design process rationale: Why were the activities conducted in this sequence?

- The first three activities of the process allowed us to perform a within-project analysis to derive the design knowledge that was accumulated for each visual inquiry tool. The last two activities consisted of the cross-project analysis in which the instance knowledge was abstracted to be projectable onto a broader class of problems/design requirements (i.e., strategic management problems) and design features (i.e., abstract design features that are not instance specific but can be used for a variety of visual inquiry tools).

4. Advancing: How does the DK chunk provided compare to existing DK?

- Previous design knowledge for developing visual inquiry tools only included instantiations. No design principles for guiding the development of visual inquiry tools were available to designers.

About the Authors

Jan vom Brocke is a professor of information systems, the Hilti Chair of Business Process Management, and director of the Institute of Information Systems at the University of Liechtenstein. His research focusses on business process management and related aspects of digital innovation and transformation. He has published, among others, in *MIS Quarterly, Journal of Management Information Systems*, *Journal of Information Technology*, *European Journal of Information Systems, Information Systems Journal*, *Communications of the ACM, MIT Sloan Management Review*, and *Management Science*. He is a Fellow of the Association for Information Systems and has held various editorial roles and leadership positions in information systems research and education.

Robert Winter is a full professor of business & information systems engineering at the University of St. Gallen (HSG) and director of HSG's Institute of Information Management. He was founding academic director of HSG's Executive Master's of Business Engineering program and academic director of HSG's PhD in Management program. He received master's degrees in business administration and business education as well as a doctorate in social sciences from Goethe University, Frankfurt, Germany. He served as vice editor-in-chief of *Business & Information Systems Engineering*, he is currently a senior editor at the *European Journal of Information Systems* and serves on several editorial boards including *MIS Quarterly Executive*. His research interests include design science research methodology, enterprise architecture management and the governance of very large IT projects/programs. He publishes in leading information systems conferences and journals, such as *MIS Quarterly*, *European Journal of Information Systems, Journal of Information Technology,* and *Business & Information Systems Engineering*. One of his publications received the AIS Senior Scholars' Best Paper award for 2017.

Alan R. Hevner is a Distinguished University Professor and Eminent Scholar in the Information Systems and Decision Sciences Department in the Muma College of Business at the University of South Florida. He holds the Citigroup/Hidden River Chair of Distributed Technology. Dr. Hevner's areas of research interest include design science research, information systems development, software engineering, distributed database systems, healthcare systems, and internet of things computing. He has published over 250 research papers on these topics and has consulted for a number of Fortune 500 companies. Dr. Hevner received a PhD in computer science from Purdue University. He has held faculty positions at the University of Maryland and the University of Minnesota. Dr. Hevner is a Fellow of the American Association for the Advancement of Science, a Fellow of the IEEE, and a Fellow of the Association for Information Systems. He is a member of ACM and INFORMS. Additional honors include selection as a Parnas Fellow at Lero, the Irish software research center, a Schoeller Senior Fellow at Friedrich Alexander University in Germany, and the 2018 Distinguished Alumnus award from the Purdue University Computer Science Department. From 2006 to 2009, he served as a program manager at the US National Science Foundation in the Computer and Information Science and Engineering Directorate.

Alexander Maedche is a full professor of information systems at Karlsruhe Institute of Technology (KIT), Germany. He heads the research group Information Systems & Service Design (http://issd.iism.kit.edu/) at the Institute of Information Systems and Marketing (IISM) and serves as a director of the Karlsruhe Service Research Institute (KSRI). Dr. Maedche received a PhD in 2001 from the University of Karlsruhe (TH). Alexander Maedche worked as full professor at the University of Mannheim from 2009 to 2015. Between 2003-2009, he worked in industry at Bosch and SAP in managerial positions in the fields of corporate IT management and enterprise software product development. His research work is positioned at the intersection of information systems and human-computer interaction (HCI). Specifically, he focuses on designing interactive intelligent systems. He publishes in leading information systems and computer science conferences and journals, such as *Management Information Systems Quarterly, Journal of the Association of Information Systems*, *Business & Information Systems Engineering*, *Information & Software Technology*, *Computers & Human Behavior*, *International Journal of Human-Computer Studies, IEEE Transactions on Software Engineering*, *IEEE Intelligent Systems, VLDB Journal,* and *AI Magazine.*