Synthesizing a Solution Space for Prescriptive Design Knowledge Codification

Möller, Frederik; Hansen, Magnus Rotvit Perlt; Schoormann, Thorsten

Published in:
Scandinavian Journal of Information Systems

Publication date:
2022

Document Version
Peer reviewed version

Citation for published version (APA):
Möller, F., Hansen, M. R. P., & Schoormann, T. (2022). Synthesizing a Solution Space for Prescriptive Design Knowledge Codification. *Scandinavian Journal of Information Systems*, *34*(2), [1].
https://aisel.aisnet.org/sjis/vol34/iss2/1

Synthesizing a Solution Space for Prescriptive Design Knowledge Codification

Article *in* Scandinavian Journal of Information Systems · December 2022

Frederik Möller
Technische Universität Dortmund

Thorsten Schoormann
Universität Hildesheim

*This is the accepted author version of "**Synthezising a Solution Space for Prescriptive Design Knowledge Codification"** by Frederik Möller, Magnus Rotvit Perlt Hansen, and Thorsten Schoormann to appear in the 2022 Winter issue of the Scandinavian Journal of Information Systems (SJIS).*

The final published version will not be identical (e.g., visual design, pages)

Accepted Author Version – Forthcoming Winter 2022

Synthesizing a Solution Space for Prescriptive Design Knowledge Codification

Frederik Möller

TU Dortmund University and Fraunhofer ISST
Frederik.Moeller@tu-dortmund.de

Magnus Rotvit Perlt Hansen

Roskilde University
magnuha@ruc.dk

Thorsten Schoormann

University of Hildesheim
thorsten.schoormann@uni-hildesheim.de

Abstract. One of Design Science Research's (DSR) principal purposes is to generate and codify design knowledge. Codification in DSR is done by providing clear chunks of prescriptive knowledge that guide the design of future solutions, including instructions on how to design (parts of) artifacts. Although various codification mechanisms have emerged over the last years, design principles are among the most prominent mechanisms. Yet, distinguishing between different codification mechanisms is often blurry, hindering designers from making informed decisions regarding appropriate mechanisms for their research aim and leveraging the full potential of the prescriptive knowledge. We seek to bridge the challenge of selecting from the fuzzy array of codification mechanisms by proposing an inductively generated solution space. We provide a taxonomy to organize essential elements of prescriptive knowledge based on an analysis of design-oriented literature in four meta-dimensions (i.e., communication, application, development, and justification). These meta-dimensions make transparent how codified prescriptive design knowledge works. Overall, the taxonomy guides designers in reflecting on and selecting from the set of suitable elements for their statements. Also, providing a synthesis of options for codifying prescriptive design knowledge will simplify the identification and advance the positioning of DSR contributions.

Key words: Design Science, Design Knowledge, Prescriptive Statements, and Codification.

1 Introduction

Design Science Research (DSR) is fundamentally different from other sciences as per its focus on artifacts (Baker 2008). Artifacts translate a set of requirements from a problem state to a more satisfactory solution state that fulfills these requirements (Purao et al. 2001; Simon 1996). The paradigmatic difference refers to the *mutandum*, i.e., such objects of observation that change their form over time, enabling the problem state to be transformed into a preferable solution state (Järvinen 2007; Simon 1996; van Strien 1997). Design science focuses on generating novel and purposeful solutions brought into existence artificially, contrary to explaining natural phenomena (Gregor 2006). Through these artifacts, designers aim to solve organizational problems with a real-world impact (Romme 2003). The designer generates different design knowledge types during building artifacts, usually prescriptive, which is one of the most critical outcomes of design science (Denyer et al. 2008; Möller et al. 2020; Seidel et al. 2017; van Aken 2004; van Aken 2005a). Significantly, design knowledge differs from other types of knowledge per its inherent focus on prescription rather than description (Gregor and Jones 2007; Romme 2003). Only by accumulating and codifying prescriptive design knowledge a successful design can transcend the boundaries of a single instance and be reused by others (Chandra Kruse et al. 2019; McAdams 2003; Schoormann et al. 2021). Codification is the process of condensing knowledge that enables other designers to adopt such knowledge in different scenarios at different times (Cohendet and Meyer-Krahmer 2001; Hall 2006; Nowack 1997).

Given the importance of prescriptive design knowledge, scholars have proposed numerous mechanisms for codification (Gregor and Jones 2007), including *design principles* (e.g., Chandra Kruse et al. 2015), *technological rules* (e.g., Bunge 2012), *design rules* (e.g., Romme and Endenburg 2006), and *design propositions* (e.g., Denyer et al. 2008). Although different termini are used to describe these codifications, we see potential and a significant contribution in identifying the mechanism's actual differences and underpinning assumptions, which need to be considered when formulating design knowledge. For example, while *design rules* (Baldwin and Clark 2000) are associated with the modularization of an artifact, *technological rules* are typically specified sequentially (Bunge 2012), yet these properties are not mandatory for other mechanisms. Moreover, in a recent study, Gregor et al. (2020) refer to different codification mechanisms, including *technological rules* and *design guidelines,* as "(…) range of view and nomenclature for design principles." To the best of our knowledge, little research investigates the characteristic attributes of codification mechanisms in detail; a notable exception is Hansen and Haj-Bolouri (2020).

Since the research stream on codification mechanisms is vast and unstructured, leading to a high degree of blurriness in distinguishing each mechanism's properties, we believe disclosing potential trade-offs will guide designers in selecting appropriate mechanisms and considering relevant characteristics for their design knowledge. We see a promising potential for creating a holistic view of prescriptive design knowledge codification in the basic transferability and differences. Distinguishing between mechanisms and highlighting central characteristics can ease the instantiation of an artifact (e.g., provide less room to misapply the prescriptions if it is indicated that users should follow a sequence), leverage the full potential of such knowledge, and make the building process more transparent to reviewers. Against this backdrop, we follow a taxonomic approach to “structure or organize the body of knowledge that constitutes a field” (Glass and Vessey 1995 p. 65). We draw from the notion of a solution space, which we see as the overview of possible options to codify prescriptive design knowledge (Purao et al. 2001; Simon 1995). Hence, we asked: *What are the options to codify prescriptive design knowledge based on their inherent characteristic elements?*

To answer this, we structure prescriptive design knowledge characteristics in the form of a taxonomy that first breaks down existing prescriptive knowledge into its inherent characteristic elements. A taxonomy with its many-faceted visualization options (Szopinski et al. 2020) is a powerful tool to contrast objects of interest against each other. We choose to visualize the taxonomy morphologically, as it gives intuitive insights into the dimensions and characteristics of prescriptive design knowledge (i.e., their *Gestalt*, e.g., Ritchey 2014). Once developed, the taxonomy should represent elements of codification mechanisms for prescriptive design knowledge. In doing so, we aim at making the spectrum of prescriptive design knowledge codification transparent and accessible. Our work synthesizes codification mechanisms that share commonalities but often differ in terminology and origin. This is important from a research point of view as it means that researchers should not quarrel too much with finding a “correct” codification mechanism but instead focus on which dimensions are needed to provide the most clarity for sharing prescriptive knowledge. It is essential from a practitioner's viewpoint since more informed and clearly communicated design knowledge can aid in finding new ways to follow and instantiate the knowledge into usable solutions.

The paper is structured as follows. Following the introduction, Section 2 illustrates the theoretical background of prescriptions in design science. Section 3 details the research design. Section 4 reports on the final taxonomy, the primary outcome of the paper. In Section 5, we show the potential by outlining the value of the solution space. Section 6 discusses our findings and implications for design science as a research field. Lastly, Section 7 outlines contributions, limitations, and avenues for further research.

2 The relevance of prescriptions for design science

The design sciences produce actionable prescriptive knowledge that enables their users to instantiate an artifact more efficiently (van Aken 2005a). Unlike behavioral sciences, the design sciences bridge the gap between a problem space that captures problems, needs, goals, and requirements and a solution space containing solutions to address the problems through artifacts (Maedche et al. 2019; Purao et al. 2001; Simon 1996). In that process, the designer generates design knowledge that must be stored and accumulated to advance the knowledge base on artifact design (Vom Brocke et al. 2020). The "(…) practical ethos (…)" of design science research implies a necessity to make its products reusable in other instances that exceed the initial scenario of their development (Iivari et al. 2018, p. 1). Codification is the mechanism used to store, elevate, and make chunks of design knowledge reusable that emerge during designing and its research (Hall 2006). Codification is the process of accumulating knowledge and representing it in a format so that it can be reused in additional instances, by other designers and at a different point in time (Cohendet and Meyer-Krahmer 2001; Hall 2006). Examples of these formats include *prescriptive statements* (Chandra Kruse et al. 2015), *books* (Davenport and Prusak 1998), or *design exemplars* (van Aken 2005a). Codification mechanisms enable designers to leverage the past experiences of other designers and surmount errors that have already been made (McAdams 2003). They allow transcending singularity and go beyond a "single success story" (Chandra Kruse and Seidel 2017, p. 180). Rather than repeating problems made in the prior projects or activities, codified, prescriptive design knowledge "eases the burden of applying the problem-situational knowledge" (Nowack 1997, p. 51). Naturally, using design knowledge in other instances enhances the probability of requiring fewer design iterations in subsequent design projects, reducing cost and effort (Kim 2010). Effectiveness is especially important, as prescriptive design knowledge often lags behind artifact design processes and requires someone to 'take the first step' (Gurzick and Lutters 2005; Kim 2010).

Most types of prescriptive statements in design science are *heuristics*. Rather than guaranteeing an outcome, heuristics give guidance to increase the chance of succeeding in successful design (Fu et al. 2015). These outcomes usually require grounding in some primary mechanism, explaining why and how it should work (Romme and Endenburg 2006). For instance, prescriptive design knowledge can be grounded in several input sources, such as a kernel theory, natural law, or empirical evidence (Goldkuhl 2004; Romme 2003; Romme and Endenburg 2006; van Aken 2004; Walls et al. 1992). The codification mechanisms are usually targeted to enhance a designer's ability to achieve a particular outcome yet require the designer to possess adequate knowledge to implement them (Kim 2010; van Aken 2004; van Aken 2005a). Usually, prescriptive design knowledge explains some form of causality, implying that a particular outcome can be achieved if one follows a set of specific steps. Goldkuhl (2004, p. 64) defines prescriptiveness as "[i]f act A then Goal G ("ought") where act A equals cause C and Goal G equals effect E in the explanatory statement." Table 1 gives an overview of the underlying prescriptive logic of codification mechanisms that we consider in the article (based on the list in Gregor et al. 2020).

Mechanism	*Domain*	*Prescriptive logic*
Technological Norm	General	"if we want to achieve the aim A, and the situation is of type B, then we should bring about the cause X" (Niiniluoto 1993 p. 13)

Design Law[1]	General	"Functional property A in situation B can be achieved by imposing structural property X." (Kuipers 2013, p. 460)
Technological Rule	General	"if you want to achieve Y in situation Z, then perform action X" (van Aken 2001 p. 3)
Design Proposition	Organization and Management Studies	"if you want to achieve outcome O in context C, then use intervention type I". (Denyer et al. 2008 p. 395)
Design Guideline	Engineering	"if S(G,C) do A and achieve E(sG)." Roozenburg and Eekels (1995) as cited in (Nowack 1997, p. 45)
Design Principle	Information Systems	"If you want to design intervention X [for the purpose/function Y in context Z], then you are best advised to give that intervention the characteristics A, B, and C [substantive emphasis], and to do that via procedures K, L, and M [procedural emphasis], because of arguments P, Q, and R." (van den Akker 1999, p. 9)
Design Rule	Engineering	"to achieve A in situation S, do D" (Romme 2003, p. 566)

Table 1. Underlying prescriptive logic of different codification mechanisms (adapted from and based on Gregor et al. 2020)

To stress the relevance of the selected codification mechanisms, we explored their distribution in the IS disciplines. Therefore, we searched for 'term of codification' and "information systems" in Google Scholar using Harzing (2007)'s citation tool 'Publish & Perish' and analyzed the frequency of occurrences since 2000, performed in 09/2021. We, of course, have to note that while we cannot ensure that all of the papers produce knowledge on the different mechanisms, they can indicate a degree of interest within the IS community. Figure 1 shows that all of the mentioned mechanisms are addressed and that Design principles are the most dominant ones, with 12.102 hits. This is followed by design guidelines (2.356 hits) and design rules (1.657 hits).

[1] Kuipers (2013) explains that *design laws* cover what 'ought' to be, i.e., clear prescriptiveness implicitly in comparison to Niiniluoto (1993)'s definitions of *technical norms*.

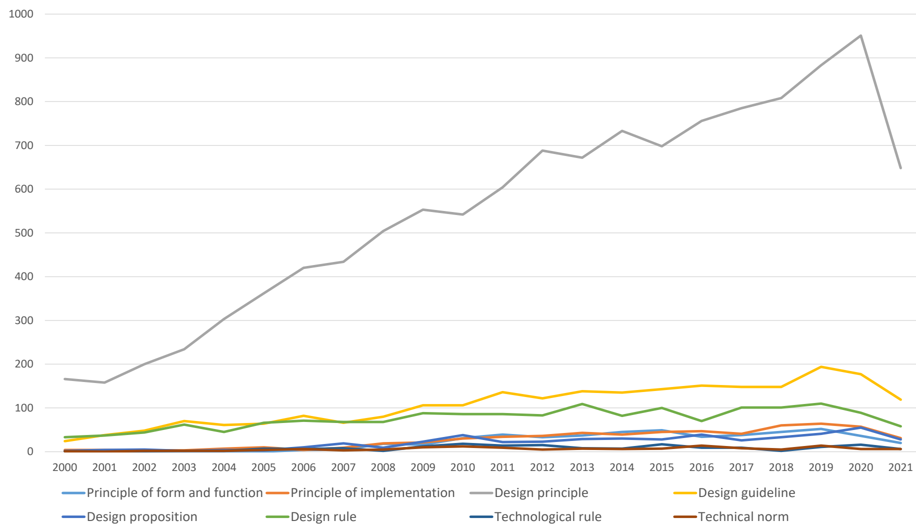

Figure 1. Distribution of codification mechanisms in the literature

3 Research design

To codify the solution space for prescriptive codification mechanisms, we used a taxonomy because it encloses the available options visually and intuitively as well as enables the deconstruction of an object of analysis into designable dimensions and characteristics (Nickerson et al. 2013). For collecting information on codification mechanisms, we combine the taxonomy approach with Webster and Watson (2002)'s notion of a concept-matrix. We screened a sub-sample of papers for elements on prescriptive design knowledge codification and jointly discussed each dimension and characteristic in the team of authors (see Table 3). Because of the heterogeneity of the papers and terminology, we south to scope the review to include predominantly seminal or pivotal papers (Cooper 1988). Following our method, our literature review can be positioned as *narrative*. We strive to give an overview of previous codification mechanisms and highlight the issues that we have learned during the process (Schryen et al. 2020).

Facing the heterogeneity of this field, we began with gathering potential codification mechanism candidates from our experience individually and in brainstorming sessions. We combined those from our experience with the 'views of design principles' by Gregor et al. (2020). From this, in line with this study's focus, we explicitly sought codification mechanisms that pointed to a prescriptive logic and used this as the primary inclusion criterium (see Table 1). For example, we used Gregor et al. (2020)'s list as a starting point and excluded those that did not follow a clear prescriptive logic (e.g., computing principles). We also excluded *technological knowledge* since it is not a codification mechanism per se but a concept that differentiates knowledge produced in engineering disciplines against knowledge produced in other disciplines (e.g., Houkes 2009). Next to the list of Gregor et al. (2020), we also looked for additional mechanisms referenced in the literature corpus:

incorporating any paper that included further evidence of other codification mechanisms in successively studied papers. Examples included Romme (2003), who named *design propositions* and *design rules* (the latter was not part of Gregor et al. 2020), as well as the exclusion of *design laws* due to their normative nature that only implicitly refers to a prescriptive logic (Kuipers 2013).

We also filtered mechanisms alongside their conceptual position, which means that we excluded preceding and subsequent concepts. For example, design principles are often (but not always) located between (meta-) requirements and features (e.g., see Meth et al. 2015; Wache et al. 2022). While requirements are mostly used to describe a class of goals (e.g., Walls et al. 1992) and address "(…) the opportunity/problem to be addressed (…)" (Hevner 2007, p. 89), design principles typically serve as the central concept for the codification of prescriptive knowledge. Following this, we decided to primarily consider the 'design principle level.' Three observations argued for the usefulness of the above-mentioned reviewing and filtering strategy. First, there seems to be no structured way to extract prescriptive design knowledge codification mechanisms because the typical terms (e.g., principles, guidelines, or rules) are considered as regular expressions and elements in the academic literature that do not necessarily refer to design-oriented research. Second, there are large differences between the cumulative body of literature on the different mechanisms. Many publications develop design principles, but few develop technical norms (see Figure 1). In this respect, the focus on predominantly conceptual papers was necessary. Third, in some cases, it was rather intuitive to identify seminal papers (e.g., the works of Bunge 1966; Bunge 2012 for technological rules or Niiniluoto 1993 for technical norms). In other cases, we relied on finding relevant papers either through cross-references (e.g., Gregor et al. 2020; Romme and Endenburg 2006) or high-ranking hits on Google scholar. Using an interdisciplinary search engine was, in our case, the most sensible option since it has no restriction in terms of domain or community and gave us a way to find adjacent terms and highly relevant papers; see also Figure 1 and the list of Gregor et al. (2020).

After collecting a sample of papers, we examined how a particular codification mechanism presented in a paper is described and specified. Table 3 lists the codification mechanisms that we considered in our literature review and analysis with corresponding definitions. Our approach to analysis is concept-centric (Webster and Watson 2002), in which we have examined conceptual papers on codification mechanisms and inductively generated codes. We did this threefold. First, we looked for properties that are expressively mentioned in the literature (e.g., *design principles* addressing form, function, and implementation and *technological rules* being sequential). Second, we looked for information on properties that we can infer from context or use other terminology in the original sources. In a third step, we used our findings to synthesize the final taxonomy through a series of discussions and activities (e.g., promoting, merging, or renaming dimensions, see Kundisch et al. 2021), resulting in a taxonomy that was a product of numerous elaborations among the team of authors.

4 The solution space for prescriptive design knowledge codification

In a nutshell, we collected a variety of codification mechanisms that are more similar to each other than they are different. Naturally, all of the mechanisms require the knowledge to be prescriptive, i.e., instructing users to achieve a specific goal through some action. Furthermore, the codification mechanisms used seem to aim to address a class of problems rather than an instance. Addressing a class of problems is beneficial because it enables reuse in other cases and the accumulation of design knowledge (Chandra Kruse et al. 2016). In terms of users, these chunks of prescriptive design knowledge are intended for use by professionals, such as designers, with the relevant expertise to instantiate it. For example, how and if prescriptive design knowledge is instantiated might strongly differ depending on the various levels of skill, experience, and designer's environment (Chandra Kruse et al. 2016). Table 2 summarizes the main commonalities we see. In contrast, some elements either pose blurry or clear differences. For example, technological rules are distinctively part of a sequence that determines the order they are supposed to be executed (Bunge 1966; Bunge 2012). Another example is that design rules should be strictly followed, while other codification mechanisms are recommendations (Baldwin and Clark 2000).

Characteristic	*Example*
Prescription	See Table 1.
Adress a class of problem	Design propositions: “A design proposition can be seen as offering a general template for the creation of solutions for a particular class of field problems” (Denyer et al. 2008, p. 395) Technological rule: “(…) means that it is not a specific prescription for a specific situation, but a general prescription for a class of problems.” (van Aken 2004, p. 228)
Professional users	Design guideline: “It follows that the end user of the design guidelines is the designer of the interface” (Kim 2010, p. 670) Design propositions: “Typically demand much professional knowledge and expertise (…)” (Denyer et al. 2008, p. 396)

Table 2. Commonalities of prescriptive design knowledge codification

From our analysis, we choose to construct the taxonomy of the solution space in the form of a morphology. Using a morphology has the significant advantage of being an intuitive tool for understanding and building objects based on various design characteristics (Ritchey 2014; Szopinski et al. 2020). We inductively derived four meta-dimensions, namely *Communication*, *Application*, *Development*, and *Justification*, which serve as aggregated theoretical lenses for the dimensions. The inductive process was done by sorting each dimension to intuitively superordinate meta-dimensions and attaching it with a label that subsumes them logically. In the following, we will explain each dimension structured through the meta-dimensions. Table 3 lists codification mechanisms considered in the study and the corresponding sources that we have analyzed to construct the solution space.

Prescription	*Definition*	*Sources*
Principles of Form and Function	"The abstract "blueprint" or architecture that describes an IS artifact, either product or method/intervention" (Gregor and Jones 2007, p. 322)	(Gregor et al. 2013; Gregor and Jones 2007; Markus et al. 2002)
Principles of Implementation	"A description of processes for implementing the theory (either product or method) in specific contexts." (Gregor and Jones 2007, p. 322)	(Gregor et al. 2013; Gregor and Jones 2007; Markus et al. 2002)
Design Principles	"(…) a recommendation or suggestion for a course of action to help solve a design issue." (McAdams 2003, p. 357)	(Chandra Kruse et al. 2015; Gregor and Jones 2007; McAdams 2003; van den Akker 1999)
Design Guidelines	"A design guideline is a prescriptive recommendation for a context sensitive course of action to address a design issue." (Nowack 1997, p. 62)	(Greer et al. 2002; Gurzick and Lutters 2005; Kim 2010; Nowack 1997)
Design Propositions	"Design propositions, as the core of design knowledge, are similar to knowledge claims in science-based research, irrespective of differences in epistemology and notions of causality." (Romme 2003, p. 567)	(Carlsson 2007; Denyer et al. 2008; Romme 2003; Romme and Endenburg 2006; van Aken et al. 2016)
Design Rules	"Design rules are elaborate solution-oriented guidelines for the design process (e.g., "if condition C is present, to achieve A, do B"). These rules serve as the instrumental basis for design work in any organizational setting." (Romme and Endenburg 2006, p. 288)	(Baldwin and Clark 2000; Brusoni et al. 2006; Romme 2003; Romme and Endenburg 2006)
Technological Rules	"(…) a chunk of general knowledge, linking an intervention or artefact with a desired outcome or performance in a certain field of application." (van Aken 2004, p. 228)	(Bunge 1966; Bunge 2012; van Aken 2001; van Aken 2004; van Aken 2005a, 2005b)
Technical Norm	"Technical norms are concerned with the means to be used for the sake of attaining a certain end." (Bulygin 1992, p. 212)	(Niiniluoto 1993, 2014; Wright 1963)

Table 3. Overview of prescriptive codification mechanisms used in the study

We derived a taxonomy with 11 dimensions and several corresponding characteristics based on selected mechanisms (see Figure 2). Below, we explain the solution space alongside the four inductively generated meta-dimensions (MD): **Communication** that subsumes dimensions describing the extent of the prescriptive design knowledge, e.g., who develops it, what medium is used to codify it, or whether it addresses the artifact as a product or the process of designing it. **Application** comprises dimensions describing the prescriptive knowledge 'looks like, e.g., whether it is modular, sequential, or heuristic. **Development** that contains dimensions describing the method used to codify the design knowledge, e.g., whether the approach reflects on a finished design or synthesizes a priori data into design knowledge. **Justification** entails dimensions for evaluation and grounding of the mechanisms. Each dimension must at least have two characteristics to enable decision-making (Nickerson et al. 2013). Usually, taxonomies should strive for mutually exclusive dimensions (e.g., Bailey 1994). However, in terms of usability and conciseness (see Nickerson et al. 2013), we refrained from adding more characteristics that would subsume multiple other characteristics (e.g., through a characteristic called *both* or *other*). We did this to enhance understandability and prevent potential patterns consisting of many characteristics that subsume others. A notable exception is the dimension of *developers* since the characteristic both here also indicates a notion of working in a transdisciplinary team to codify prescriptive design knowledge.

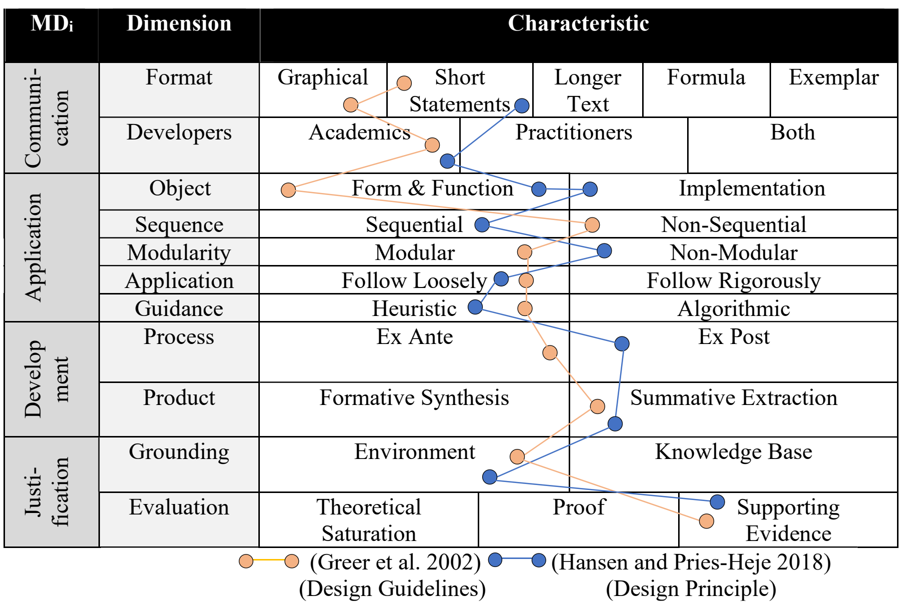

Figure 2. Solution space of prescriptive design knowledge codification using two examples of papers contributing with design guidelines and design principles, respectively

We can show the applicability of the taxonomy in two illustrative cases. The taxonomy results from conceptual papers, and not all dimensions are made transparent in papers proposing prescriptive design knowledge. For the purpose of a simple illustrative application scenario, we will assume that the two following examples were developed by researchers and are recommendations (see the patterns in Figure 2). First, Greer et al. (2002)

present design guidelines for product evolution modularized in four domains (relative motion, graph structure, function, and analysis). Next to prescriptive statements, the design guidelines are supported by visual illustrations. The guidelines are extracted through observation of empirical examples (products). Second, contrarily, the design principles in Hansen and Pries-Heje (2018) stem from the analysis of two in-depth cases and are to be used in a sequence of both principles of form and function and principles of implementation. Both examples show typical features of prescriptive design knowledge that only differ slightly. For instance, while the design principle case proposes five condensed design principles (typical for these studies, e.g., Iivari et al. 2020), the design guideline case proposes 29 guidelines. From the cases, we can extract knowledge that prescriptive design knowledge codification would benefit from finding similarities and differences and generating transparency and clarity by elaborating on essential elements of the solution space in Figure 2. We will describe these issues in more detail in the following section.

4.1 Meta dimension 1: Communication

Communication (MD1) subsumes design parameters that logically outline the prescriptive codification mechanisms they are supposed to look like and achieve. It illustrates the format the prescriptive design knowledge is codified in (Format) and who develops the prescriptive design knowledge (Developer).

Prescriptive design knowledge varies in how it is codified **(Format)**, ranging from *graphical* illustrations, *short statements*, *longer text* (e.g., books), *formulas*, or *exemplars*. For example, while design principles are usually short prescriptive statements or structures (e.g., see the formulation template of Chandra Kruse et al. 2015 or structure of Gregor et al. 2020), van Aken (2005a, p. 23) states that “the actual description of a rule may fill an article, a report or even a whole book.” Greer et al. (2002) propose a set of design rules that include prescriptive statements complemented through graphical aides and formulas.

In the examples given above, the **Developers** of the prescriptive design knowledge are *academics*. However, multiple sources point to developers being also *practitioners* (e.g., see Romme and Endenburg 2006) or *both* (van Aken 2005a).

4.2 Meta dimension 2: Application

Application (MD2) includes all dimensions influencing the form and application of the prescriptive knowledge.

Object addresses the notion of design as both a verb and a noun and equivocally describes the design product and design process (Walls et al. 1992). The design product usually refers to principles that outline an artifact's *form and function*. Complementarily, principles of *implementation* address the process required to design the artifact.

The dimensions **Sequence** and **Modularity** binarily indicate whether the codified set of design knowledge is supposed to be executed *sequentially* (coupled in a sequence) or whether they are to be seen as *modular* (decoupled from one another) design system components, respectively.

The **Application** of the prescriptive design knowledge can either be a recommendation that is to be *followed loosely* or a *strictly to be followed* instruction (e.g., a medical recipe). Most codification mechanisms are recommendations, yet, Baldwin and Clark (2000, p. 6) explain *design rules* to be "(…) not just guidelines or recommendations: they must be rigorously obeyed in all phases of design and production."

Guidance detailed how the prescriptive design knowledge is to be designed regarding the 'guaranteed' effectiveness of its outcome. The dichotomy is between *heuristic* and *algorithmic* prescriptions. van Aken (2004, p. 227) differentiates heuristic and algorithmic technological rules by the example of treating disorders as follows: Algorithmic technological rules are as follows "in order to cure disorder Y, you follow a course of treatment consisting of taking 0.3 milligrams of medicine X during 14 days." In contrast, heuristic technological rules "in order to cure disorder Y, you follow a course of treatment consisting of rest, exercising and a fat-free diet." The pivotal difference is that heuristic prescriptions infer no guarantee for success, while the effectiveness of algorithmic prescriptions (often formulated quantitatively) can be proven (van Aken 2004; van Aken 2005a).

4.3 Meta dimension 3: Development

Development (MD3) includes all dimensions describing the activities employed to develop the codified prescriptive design knowledge. First, the general distinction **(Process)** is between *ex post* prescriptive statements that come into existence reflectively through experience or *ex ante* design knowledge, for instance, collected in multiple case studies before the design process. In our sample, we obtained different terminology that refers to the actual development of prescriptions: For example, design principles can be derived *reflectively* or *supportive* (Möller et al. 2020). Technical norms are developed by following the two approaches 'from above' (e.g., from theoretical input) and 'from below' (e.g., from case-based input) (Niiniluoto 1993, p. 13), as well as technological rules which can be *developed* or *extracted* (van Aken 2005a). We chose the terminology 'ex ante' and 'ex post,' since it is the most generical formulation that is not associated with one specific codification mechanism to differentiate 'before designing artifact' and 'after designing the artifact.'

Second, the actual **Product** can be generated through *formative synthesis* or *summative extraction*. Practically, that happens in two ways. In terms of formative development, through the *synthesis* of data from various sources, such as theory, scientific literature, or qualitative studies (Möller et al. 2020; van Aken 2004; van Aken 2005b). Alternatively, prescriptive design knowledge can be developed after a project is completed by means of extraction from finished cases or design projects (Gregor 2009; Möller et al. 2020; van Aken 2004). We used the terminology *formative synthesis* and *summative extraction* since they convey the action taking place more precisely.

4.4 Meta dimension 4: Justification

The meta-dimension Justification (MD4) subsumes dimensions of argumentative mechanisms on why the design principles should work and how they are evaluated. Concerning the grounding **(Grounding)**, we draw from the well-established dichotomy in DSR proposed by Hevner et al. (2004), namely *environment* and *knowledge base*. Grounding "(…) services to develop a robust understanding of how and why the design (rules) operates" (Romme and Endenburg 2006, p. 289). In terms of the environment, there are different ways to ground prescriptive design knowledge, including in the designer's own experience, experiments, or empirical evidence. For example, Kim (2010) explains that design rules can be grounded in laboratory experiments or expert opinions. Contrarily and complementarily, design knowledge should be grounded in the existing knowledge base, which, from our analysis, comprises scientific literature, natural laws, or kernel theories. We distinguish between literature and kernel theories to demarcate a systematic literature review from using kernel theories, such as the Resource-Based View (RBV) (Barney 1991). For instance, van Aken (2004) points to technological rules being grounded in natural laws.

In terms of evaluation **(Evaluation)**, we distinguish between *theoretical saturation,* meaning that prescriptive design knowledge is evaluated progressively through accumulating empirical evidence, for example, in a series of case studies (Eisenhardt 1989; van Aken 2005a). Contrarily, prescriptive design knowledge might be *provable*, which usually is possible when they are algorithmic (van Aken 2005a). Also, collecting *supporting evidence* can support the validity of prescriptive design knowledge.

5 Discussion and reflection

Based on our analysis, we reflect on what can be learned for prescriptive design knowledge codification and formulate two of them as a proposition for further guidance in formulating prescriptive design knowledge. In detail, we see two significant propositions that we discuss in detail below. First, identifying paradigmatic differences between design knowledge codification (e.g., deciding on algorithmic or heuristic codification). Second, thinking in a comprehensive solution space and entangling it with the particular design context opens up the opportunity for formulating specific, well-fitting prescriptions rather than using a pre-defined one. For example, consider the artifact required to be built in a specific sequence. The prescription should reflect that.

5.1 Highlighting paradigmatic differences in prescriptions

Prescriptive design knowledge codification has many similarities across the individual concepts we have analyzed. We can derive a 'smallest common denominator' in the literature agreeing that codified prescriptive design knowledge as instructions for professional users to address a class of problems. Necessarily, they also require to be prescriptively formulated to guide action, i.e., formulate a clear causal chain to achieve some goal through a specific action (Goldkuhl 2004) (see commonalities summarized in Table 2).

Contrarily, some characteristics let us distinguish different types of codification. The characteristics in Table 4 are not, *per se*, different for each mechanism, yet they show more variance than other characteristics. Subsequently, we see them as paradigmatic because they are distinguishing features shaping the underlying logic of the prescription. For example, *technological rules* can both be *algorithmic* as well as *heuristic* (van Aken 2004). That is contrasted with the other codification mechanisms, which are typically *heuristics* (e.g., see van den Akker 1999 or Kim 2010). In terms of developing prescriptive design knowledge, most codification mechanisms have a bottom-up empirical route and a top-down theoretical, deductive route pointing to multiple grounding mechanisms (see Figure 2). Perhaps the most significant unique selling points, at least *de nomine*, are the *sequentiality* (Bunge 1966; Bunge 2012) of *technological rules* and the *modularity* of *design rules* (Baldwin and Clark 2000). *Sequentiality* is to "(…) perform a finite number of acts in a given order and with a given aim" (Bunge 2012, p. 338), and *modularity* is "(…) based on the twinned principles of interface standardization and components decoupling." (Baldwin and Clark 2000, p. 179). While we did not find evidence in other types of prescriptive design knowledge for the opposite, i.e., for *non-sequentiality* or *non-modularity*, technological and design rules are the only two that have explicitly mentioned these concepts (see Table 4).

Dimension	*Sequentially*	*Modularity*	*Guidance*
Technological Rule	Sequential (Bunge 1966)	Non-Modular*	Algorithmic or Heuristic (van Aken 2005b)
Design Rules	Non-Sequential*	Modular (Brusoni et al. 2006)	Algorithmic or Heuristic (Greer et al. 2002; Romme 2003; Roozenburg and Eekels 1995)
Technical Norm	Non-Sequential*	Non-Modular*	Heuristic (Niiniluoto 1993)
Design Principles	Non-Sequential*	Non-Modular*	Heuristic (van den Akker 1999)
Design Guidelines	Non-Sequential*	Non-Modular*	Heuristic (Kim 2010; Nowack 1997)
Design Proposition	Non-Sequential*	Non-Modular*	Alogirthmic or Heuristic (Carlsson 2007; Romme and Endenburg 2006)

Table 4. Comparison of different paradigmatic properties of codification mechanisms. *We did not find evidence for dimensions being non-sequential or non-modular. Yet, these concepts are only explicitly mentioned in technological rules and design rules, respectively

5.2 Configuring prescriptive design knowledge codification based on context

Reflecting on the solution space, we see no combinations of characteristics that seem impossible to use. However, in terms of *face validity*, some combinations might not work as well as others or make as much sense. Suppose that the designer uses a formula to express the prescriptions yet also selects it to be heuristic. Most likely, that would not make sense given that a formula usually does not leave room for interpretation but normally is algorithmic and needs to be followed rigorously. Given the span of forms a design project can take and the options that we have identified, we propose combining the solution space with the notion of *design context* (e.g., Herwix and zur Heiden 2021) to find sensible combinations. We see this as necessary since designing something that works in practice is fundamentally shaped by the requirements an artifact is supposed to fulfill and the people that do it, i.e., the context in which it takes place (Cross 1999; Purao et al. 2001). As a result, we see a need to make users aware of the taxonomy that configuring their solution space fundamentally reflects what they want to achieve for what purpose. Visualization might be a preferable way to codify knowledge instead of written statements in some cases. In other cases, it might be crucial to formulating prescriptive knowledge algorithmically to minimize the degrees of freedom the user has to apply them (e.g., in the case of medical recipes).

In the following, we will provide an illustrative example that underlines the utility of our solution space in light of contexts in DSR. For that purpose, we will draw from the example of designing a jug to illustrate how design principles work based on Gregor et al. (2013). In the example, Gregor et al. (2013) propose prescriptions for designing the form and function of a jug with an array of prescriptions. Consider the options you would have to communicate how to design a jug to a different person. Naturally, one could use textual prescriptions, such as (Gregor et al. 2013, p. 7):

"1. Choose a shape that has the capacity to hold liquid."
"2. Provide an opening through which liquid can be added."
"3. Provide a feature that allows liquid to be poured (i.e., a spout), which can be contiguous with the main opening, or not."
"4. Provide a feature that allows it to be picked up by a human (i.e., a handle)."
"5. Ensure it is of a suitable size and weight when full to be lifted and manipulated by a human."
"6. Ensure that the jug can stay upright on a horizontal surface."
"7. Place a handle opposite the spout in order to get maximum leverage."
"8. Ensure that the spout of the jug is above the highest point at which liquid can be held in order to make maximum usage of capacity."

Alternatively, other means of communicating prescriptions could be sensible. In this case, our solution space proposes, for instance, visual aids. A conceptual, visual representation might guide the designer in placing the *handle* more concretely. However, there is room for interpretation and communicating prescriptions even here. Figure 3 shows three easy examples of conceptualizations of jugs. Conceptualization **I** simply prescribes putting the handle somewhere on the right side of the jug. The designer now has the freedom to choose the exact spot, maybe depending on the shape of the jug and the shape of the handle. Conceptualization **II** is one step more concrete, prescribing the handle to be placed somewhere in the middle. Yet, given that the prescription is a heuristic recommendation, 'the middle' might differ from where the designer perceives to measure the height of the jug (e.g., including the foundation, just using the body, and so on). Last, Conceptualization **III** includes a formulaic prescription, communicating to designers that the handle has to be placed exactly at *L/2* with the visual aid indicating that L is the complete height of the jug, including all elements. The simple example outlined in Figure 3 illustratively shows the conundrum in formulating prescriptions. They need to prescribe a course of action, yet, the spectrum of how they are communicated leaves more or less expansive room for interpretation. It also shows that there is more than one way to communicate prescriptions and even combine them.

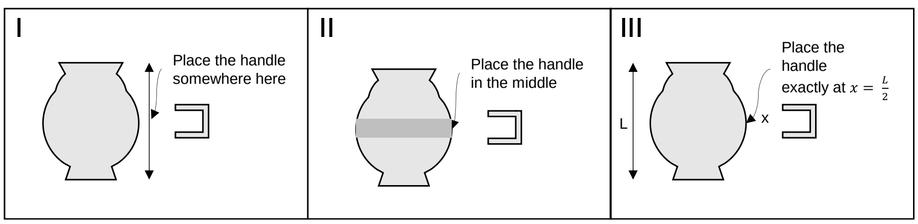

Figure 3. An illustration of different ways to communicate prescriptions

Since the options we illustrate in Figure 3 are reasonable in different situations and scenarios, we use the concept of **context** in DSR (Herwix and zur Heiden 2021) to derive abstract learnings for each dimension. Table 5 summarizes the illustrative scenario of designing a jug based on Gregor et al. (2013) and derives abstracted context-sensitive learnings from it. We use the definition of Herwix and zur Heiden (2021)[2], who see context as "the environment which surrounds an artifact or the source of the requirements that an artifact is to be evaluated against."

*D**	*Illustrative scenario*	*Abstracted context-sensitive learning*
Format	Suppose the jug requires a concrete visual design. Using visual aids or existing models might be more helpful than textual prescription. Subsequently, using additional communication mechanisms could benefit in designing a jug.	Depending on the context of the artifact-to-be-designed, it might be more sensible and practical to use specific means of communicating prescriptions above others. For example, providing visual aides or formulas might be more beneficial than giving highly interpretable short text.

[2] The definition in Herwix and zur Heiden(2021) is based on the understanding of context in DSR outlined in Hevner et al. (2004) and Maedche et al.(2019).

Developers	In the context of designing a jug, having developers that are practitioners and have design experience is different than academics researching them. It is not unreasonable to assume that practitioners and academics (or both as a team) would formalize prescriptions differently.	Depending on the context of the artifact-to-be-designed, the codified prescriptions may vary based on who codified them. For example, knowledge codified by practitioners might be closer to an instance level, while academics might reach a more abstracted level.
Object	Designing a jug can be done regarding how to use it, how it works, and how it is designed (see above).	Depending on the context of the artifact-to-be-designed, indicating whether prescriptions address form, function, and/or implementation can provide users with easier access to use them.
Sequence	Designing a jug requires, at least in part, following a sequence. If there is no jug to place a handle on, it is impossible to execute principle seven. However, it might be irrelevant whether first to attach the handle or first to attach a spout as they have no direct coupling.	Depending on the context of the artifact-to-be-designed, prescriptions should be entirely or partly sequential since some components might require others to exist beforehand. However, there can also be cases where no sequence is necessary. Indicating sequentiality can prevent errors in use.
Modularity	Designing a jug has multiple pieces that depend on each other. For example, the shape of the jug indicates where a handle can be sensibly placed. So, adjusting the shape once the handle is already placed would mean that, potentially, the handle would also need to be adjusted.	Depending on the context of the artifact-to-be-designed, prescriptions should indicate that some components of an artifact might depend on each other, meaning that the design of one component influences the design of another. Indicating these interdependencies through explicating whether the prescriptions are modular or non-modular is of value to clarify these relationships and prevent potential errors.
Guidance	The principles outlined above indicate a heuristic prescription to designing a jug. They recommend a course of action rather than prescribe strictly which dimensions the jug should have. The principles assist designers in bringing a jug into existence, yet, they leave a broad room of freedom in how the resulting jug will look and work.	Depending on the context of the artifact-to-be-designed, there might be different requirements on whether the prescription requires to be heuristic or algorithmic. For example, designing prescriptions with implications for human health should be indicated as algorithmic to narrow the room for interpretation as closely as possible. Designing a chair, room for interpretation, and flexibility in heuristic prescriptions might be sufficient or even more suitable to allow room for creativity.

Application	The principles outlined above can either be followed loosely or rigorously. Yet, designing might allow for more freedom. So, it might not be necessary to follow the order exactly. The third principle indicates that the spout can be contiguous with the main opening or not. Subsequently, the principle has a degree of looseness inherently built into it.	Depending on the context of the artifact-to-be-designed, prescriptions should be indicated as to be followed loosely or rigorously. Take, for example, the design of a bridge. Here, it is paramount that design prescriptions are followed rigorously.
Process	In designing a jug, the designer can develop prescriptions by accumulating knowledge before the design process, observing a jug in use, and inferring design principles.	Depending on the context of the artifact-to-be-designed, there might be enough design knowledge available prior to design, which merits formulating prescriptions a priori. Contrarily, prescriptions can be derived from successful implementation through reflections on the outcome and underlying process.
Product	Complementing the dimension process generating prescriptions for designing jugs can be achieved through synthesizing available design knowledge (.e.g., from the literature)	Depending on the context of the artifact-to-be-designed, either synthesizing prior knowledge or extracting new knowledge might be more efficient or even necessary.
Grounding	Designing a jug has multiple reasonable grounding mechanisms, such as prior experience in grounding other jugs (or similar artifacts) or laws of nature that somewhat prescribe some aspects of a jug (e.g., the hole should be at the top).	Depending on the context of the artifact-to-be-designed, prescriptions may be grounded in either the Environment (e.g., experience) and/or the Knowledge Base (e.g., theories or literature).
Evaluation	Evaluating if a jug works as intended, most likely, is easiest done by instantiating the prescription and seeing whether the artifact works.	Depending on the context of the artifact-to-be-designed, there are different strategies for evaluation. For example, algorithmic prescriptions should be provable, while heuristics are usually supportable with empirical evidence.

Table 5. Taxonomy application for designing a jug based on Gregor et al. (2013). * D = Dimension from the taxonomy.

6 Implications

6.1 Implications for design science

Prescriptive design knowledge is a critical outcome that enables designers to elevate their findings to a more abstract level and apply their learned knowledge in different scenarios. Our work has multiple implications for prescriptive design knowledge in design science as a research field and paves the ground for future investigations on prescriptive knowledge. Our work extends the existing knowledge on codifying prescriptive design knowledge through a cross-disciplinary and cross-conceptual element. Thus, our work is not a replacement for templates and structures (e.g., the anatomy of a design principle by Gregor et al. 2020) but enriches the field by highlighting options, i.e., visualization of prescriptive design knowledge instead of textual codification or algorithmic against heuristic formulation. The dominant mechanism in IS research is the design principle, which is frequently published in high-ranking journals and premiere conference proceedings (e.g., Möller et al. 2021). The existing body of literature includes formulation templates and structures for constructing and communicating design principles (e.g., Cronholm and Göbel 2018; Heinrich and Schwabe 2014). This is complemented by many conceptual works investigating the actual use of design principles (e.g., Chandra Kruse et al. 2022), tensions in their formulation (Chandra Kruse and Seidel 2017), investigations of their origins (e.g., Purao et al. 2020), propositions for their reusability (e.g., Iivari et al. 2020), or papers conceptualizing design principles in relation to design theory (e.g., Gregor and Jones 2007). In relation to these conceptual works and the existing body of literature in IS research, we see two ways our solution space advances the field. **First (1)**, by expanding the array of options, one has to codify prescriptive design knowledge through synthesizing and learning from other disciplines. **Second (2)**, using our solution space as an additional tool to support transparency and clarity when formulating and communicating prescriptive design knowledge.

Design options for prescriptive design knowledge codification. Arguably, the general process of designing is messy, especially when addressing ill-structured problems that demand a search process (Cross 2007; Hevner et al. 2004). Subsequently, we can argue that codifying what has been learned during the design process is hard to grasp and actually achieve. Mainly since codification for reuse, to some degree, requires abstraction from detail, generalization, and focus on 'important' aspects (Hansen and Haj-Bolouri 2020; Walls et al. 1992). Thinking in comprehensive solution spaces that transcend singular codification mechanisms, we intend to spur a discussion in the DSR community on finding common ground in codifying prescriptive design knowledge. By continuously developing the solution space (e.g., finding fundamental dimensions and characteristics), the DSR community can position their knowledge and find new ways to codify prescriptive design knowledge. Subsequently, our work contributes to prior research addressing the issue of consistency in DSR knowledge contributions (Dwivedi et al. 2014). Given the above, our approach is also a proposition to learn from individual concepts, be they design principles or technological rules, and to cross-profit from their particular advantages. For example, design principles are usually codified linguistically in textual form. Yet, we argue, they could be complemented through design examples or visual aids.

Clarity in formulating and instantiating. Clarifying how prescriptive design knowledge should be used as the potential to strengthen its *reliability* and *operationalizability* (Nowack 1997). For example, design principles usually do not seem sequential or modular, yet, implementing these concepts could maintain rigor and usability. Differentiating and indicating whether prescriptive design knowledge is a loose recommendation or needs to be followed strictly is essential for formulating, publishing, and instantiating such knowledge. It enables users to understand better how instantiation is supposed to happen and what codifier of prescriptive design knowledge had in mind. Algorithmic prescriptions are highly specific in what constitutes the prescriptions (e.g., as known from chemistry: take x or y mg of a particular substance to achieve a reaction), which also has implications for the probability of attaining a specific outcome. Contrarily, heuristic prescriptions, which are context-based and only "increase[s] the chance of reaching a satisfactory but not necessarily the optimal solution." (Fu et al. 2015, p. 4). Carlsson (2007, p. 80) explains that "[a]n algorithmic design proposition can in principle guarantee a good (best) outcome," and that "[a] heuristic design proposition does not guarantee success, but it supports the development of a successful system." Explaining why prescriptive design knowledge works are of paramount importance to convince others of its utility. The range of different variants of grounding mechanisms requires strong argumentation and complementation. For example, while initial prescriptive knowledge might be derivable from theory, empirical evidence could complement it. That is primarily important in heuristic prescriptive design knowledge, which usually cannot be proven but can only be argued based on *grounding mechanisms* and theoretical saturation of empirical evidence (van Aken 2004).

6.2 Limitations and outlook

Our work is subject to limitations. First, our sample of papers only is an excerpt that we have identified to be seminal (e.g., see Bunge 2012) or that we have found following our procedure outlined in Section 3. Hence, there likely exist more articles that should be included when the study is developed further. Given that the field of prescriptive codification mechanisms is unstructured, it is likely that there are more mechanisms that we have missed and that could be uncovered in subsequent studies. As our sample is not comprehensive, other papers might explicitly attribute prescriptive codification mechanisms with a characteristic that we might have missed. Yet, as we strive for comprehensiveness in the solution space, we see that issue as mitigated, as additional individual characteristics already covered would not change the outcome. Additionally, using that taxonomic approach explicitly demands extendability once new characteristics or dimensions of the phenomenon under investigation emerge (Nickerson et al. 2013). Second, as our research is qualitative, other researchers might find other dimensions and characteristics more significant or use different terminology. Third, we have assumed that each codification mechanism is, conceptually, on a similar level and of equal value. Subsequently, we did not include transformation mechanisms between codification mechanisms in our study. Fourth, we have not distinguished between *sets* of prescriptive knowledge mechanisms and *singular* mechanisms and instead treated both as a single piece of prescriptive knowledge. Using the dimensions of modularity and sequentiality, the value of a set of prescriptive knowledge mechanisms could be further uncovered.

With the taxonomy, multiple avenues for further research are opened. First, as we took a narrow view of selected codification mechanisms, there is an opportunity to widen the frame to additional concepts. We strictly worked with mostly seminal, conceptual papers. Subsequently, there is potential to integrate papers into the study to develop prescriptive design knowledge in one of the above forms. A possible road ahead is to identify transformation mechanisms between specific types of design knowledge. For example, Greer et al. (2002) refer to the possibility of transforming design rules and design guidelines into each other by scoping the level of detail and abstraction. Lastly, as taxonomies (i.e., our *solution space*) should, generally, be extendable (Nickerson et al. 2013), we hope that it is used and extended by other researchers in subsequent works analyzing prescriptive design knowledge codification. Based on our findings, a fruitful road for further research is using specific dimensions (e.g., grounding) as a basis to indicate the maturity of design knowledge. Our study did not analyze the usefulness or weaknesses of specific configurations. However, we see this as a potential avenue for further research, resulting in overarching strategies or archetypical configurations tailored to particular contexts and design projects that can exceed existing knowledge on codifying prescriptive design knowledge. Another fruitful route for research is to extract combinations in our solution space and analyze how they contribute to spanning boundaries in an interdisciplinary team in socio-technical system designs and what tensions arise in the transfer of codified design knowledge (e.g., see Baxter and Sommerville 2011; Guzman and Trivelato 2008). Scholars already propose viewing prescriptive design knowledge (e.g., principles and design rules) as boundary objects (e.g., Romme and Endenburg 2006). A deeper analysis of different codification mechanisms with the resulting implications (e.g., the implementability of heuristic versus algorithmic prescriptions) would be an interesting route for new research. Consider the following: Although algorithmic prescriptions usually have to be followed rigorously, there may be reasons in reality that prevent this. Alternatively, using heuristics requires to ‘pay the price’ that success is not guaranteed, and more interpretation is required. Both can result in tensions that might need to be mitigated. Last, our work can be a starting point to find new ways of combining characteristics of prescriptive design knowledge codification and to assess the validity of specific configurations.

7 Conclusion

With our in-depth analysis of codification mechanisms for prescriptive design knowledge, we make contributions to academia and practice. Our results indicate that most of the codification mechanisms in our sample are highly similar yet differ in *nuances*. Our work provides a comprehensive solution space for prescriptive design knowledge codification that transcends the borders of singular concepts enclosed in silos and illustratively explicates these *nuances* against each other. For example, the notion of sequentiality in technological rules could easily be integrated into studies developing design principles, giving them (if it provides merit for the study) additional structure that can enhance insatiability. Given that we synthesize a solution space and show that it is sensible to think of these dimensions depending on the context one develops prescriptions in, we see a significant contribution to the efficiency, usability, and usefulness of prescriptions in DSR. As another example, researchers and practitioners should be specific about whether their prescriptive design knowledge is heuristic or algorithmic, whether it is a recommendation or strictly to be followed to increase reliability. Next to the reliability, our research makes prescriptive design knowledge codification more transparent, potentially benefitting researchers in presenting their work to peer-reviewers, practitioners, and other researchers. Accordingly, this study enables researchers to leverage the entire solution space of prescriptive design knowledge rather than singular concepts and transcend the field of their origin. We see that avenue as fruitful, as our analysis shows, that the primary paradigm behind the codification mechanisms is a shared understanding of prescriptive action, addressing a class of problems, and guiding designers. For practitioners, we see increasing reliability of prescriptive design knowledge as a way to make instantiation easier, operationalizable, and more accessible. For instance, it might be easier to apply design principles once they have clear instructions on how to be used (e.g., strictly) and for what (e.g., for the form or function).

We found promising indications for promoting prescriptive design knowledge within the DSR community and beyond and shed light on the discourse of what are fundamental differences and components of such prescriptions, which will help academia and practice alike.

References

Bailey, K. D. 1994. *Typologies and Taxonomies: An Introduction to Classification Techniques*, Thousand Oaks, London, New Dehli: Sage Publications. ISBN: 9780803952591.

Baker, L. R. 2008. "The shrinking difference between artifacts and natural objects," *American Philosophical Association Newsletter on Philosophy and Computers* (7:2).

Baldwin, C., and Clark, P. 2000. *Design Rules: The power of modularity*, Cambridge, MA: MIT Press. ISBN: 9780262024662.

Barney, J. 1991. "Firm Resources and Sustained Competitive Advantage," *Journal of Management* (17:1), pp. 99-120 (doi: 10.1177/014920639101700108).

Baxter, G., and Sommerville, I. 2011. "Socio-Technical Systems: From Design Methods to Systems Engineering," *Interacting with Computers* (23:1), pp. 4-17 (doi: 10.1016/j.intcom.2010.07.003).

Brusoni, S., Provera, S., and Prencipe, A. 2006. "Making Design Rules: A Multidomain Perspective," *Organization Science* (17:2), pp. 179-189 (doi: 10.1287/orsc.1060.0180).

Bulygin, E. 1992. "On norms of competence," *Law and Philosophy* (11:3), pp. 201-216 (doi: 10.1007/BF01000642).

Bunge, M. 1966. "Technology as Applied Science," *Technology and Culture* (7:3), pp. 329-347 (doi: 10.2307/3101932).

Bunge, M. 2012. *Scientific Research II: The Search for Truth*, Springer Berlin Heidelberg. ISBN: 9783642481383.

Carlsson, S. A. 2007. "Developing knowledge through IS design science research," *Scandinavian journal of information systems* (19:2), pp. 75-86.

Chandra Kruse, L., Purao, S., and Seidel, S. 2022. "How Designers Use Design Principles: Design Behaviors and Application Modes," *Journal of the Association for Information Systems* (Forthcoming).

Chandra Kruse, L., and Seidel, S. 2017. "Tensions in Design Principle Formulation and Reuse," in *Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology*, Karlsruhe: Germany.

Chandra Kruse, L., Seidel, S., and Gregor, S. 2015. "Prescriptive Knowledge in IS Research: Conceptualizing Design Principles in Terms of Materiality, Action, and Boundary Conditions," in *Proceedings of the*

48th Hawaii International Conference on System Sciences, Hawaii: USA.

Chandra Kruse, L., Seidel, S., and Purao, S. 2016. "Making Use of Design Principles," in *Proceedings of the 11th International Conference on Design Science Research in Information Systems and Technology,* St. John's: Canada, pp. 37-51.

Chandra Kruse, L., Seidel, S., and Vom Brocke, J. 2019. "Design Archaeology: Generating Design Knowledge from Real-World Artifact Design," in *Proceedings of the 14th International Conference on Design Science Research in Information Systems and Technology,* Worcester; USA.

Cohendet, P., and Meyer-Krahmer, F. 2001. "The theoretical and policy implications of knowledge codification," *Research Policy* (30:9), pp. 1563-1591 (doi: 10.1016/S0048-7333(01)00168-8).

Cooper, H. M. 1988. "Organizing Knowledge Syntheses: A Taxonomy of Literature Reviews," *Knowledge in Society* (1:1), pp. 104-126 (doi: 10.1007/BF03177550).

Cronholm, S., and Göbel, H. 2018. "Guidelines Supporting the Formulation of Design Principles," in *Proceedings of the 29th Australasian Conference on Information Systems,* Sydney: Australia.

Cross, N. 1999. "Design Research: A Disciplined Conversation," *Design Issues* (15:2), pp. 5-10 (doi: 10.2307/1511837).

Cross, N. 2007. "From a Design Science to a Design Discipline: Understanding Designerly Ways of Knowing and Thinking," in *Design Research Now: Essays and Selected Projects,* R. Michel (ed.), Basel: Birkhäuser Basel, pp. 41-54.

Davenport, T., and Prusak, L. 1998. *Working Knowledge: How Organizations Manage What They Know,* Harvard Business Press.

Denyer, D., Tranfield, D., and van Aken, J. E. 2008. "Developing design propositions through research synthesis," *Organization studies* (29:3), pp. 393-413 (doi: 10.1177/0170840607088020).

Dwivedi, N., Purao, S., and Straub, D. W. 2014. "Knowledge Contributions in Design Science Research: A Meta-Analysis," in *Advancing the Impact of Design Science: Moving from Theory to Practice,* M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta and V. Yoon (eds.), Cham: Springer International Publishing, pp. 115-131.

Eisenhardt, K. 1989. "Building Theories from Case Study Research," *The Academy of Management Review* (14:4), pp. 532-550 (doi: 10.2307/258557).

Fu, K. K., Yang, M. C., and Wood, K. L. 2015. "Design Principles: The Foundation of Design," in *Proceedings of the ASME 2015 International*

Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Boston: USA.

Glass, R. L., and Vessey, I. 1995. "Contemporary Application-Domain Taxonomies," *IEEE Software* (12:4), pp. 63-76 (doi: 10.1109/52.391837).

Goldkuhl, G. 2004. "Design Theories in Information Systems-a Need for Multi-Grounding," *JITTA: Journal of Information Technology Theory and Application* (6:2), pp. 59-72.

Greer, J., Wood, J., Jensen, D., and Wood, K. 2002. "Guidelines for Product Evolution Using Effort Flow Analysis: Results of an Empirical Study," in *DETC '02 2002*.

Gregor, S. 2006. "The Nature of Theory in Information Systems," *MIS Quarterly: Management Information Systems* (30:3), pp. 611-642 (doi: 10.2307/25148742).

Gregor, S. 2009. "Building Theory in the Sciences of the Artificial," in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology,* Malvern: USA, New York, NY, USA: ACM, 4:1-4:10.

Gregor, S., Chandra Kruse, L., and Seidel, S. 2020. "The Anatomy of a Design Principle," *Journal of the Association for Information Systems* (21:6), pp. 1622-1652 (doi: 10.17705/1jais.00649).

Gregor, S., and Jones, D. 2007. "The Anatomy of a Design Theory," *Journal of the Association of Information Systems* (8:5), pp. 312-335 (doi: 10.17705/1JAIS.00129).

Gregor, S., Müller, O., and Seidel, S. 2013. "Reflection, Abstraction, and Theorizing in Design and Development Research," in *Proceedings of the 21st European Conference on Information Systems,* Utrecht: Netherlands.

Gurzick, D., and Lutters, W. 2005. "Towards a design theory for online communities," in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems,* Utrecht: Netherlands.

Guzman, G., and Trivelato, L. 2008. "Transferring codified knowledge: Socio-technical versus top-down approaches," *Learning Organization, The* (15), pp. 251-276 (doi: 10.1108/09696470810868873).

Hall, M. 2006. "Knowledge management and the limits of knowledge codification," *Journal of Knowledge Management* (10:3), pp. 117-126 (doi: 10.1108/13673270610670894).

Hansen, M. R. P., and Haj-Bolouri, A. 2020. "Design Principles Exposition: A Framework for Problematizing Knowledge and Practice in DSR," in *Designing for Digital Transformation. Co-Creating Services with*

Citizens and Industry, S. Hofmann, O. Müller and M. Rossi (eds.), Cham: Springer International Publishing, pp. 171-182.

Hansen, M. R. P., and Pries-Heje, J. 2018. "Principles for Unbounded Secondary design," in *Proceedings of the 26th European Conference on Information Systems*, Portsmouth: United Kingdom.

Harzing, A. W. 2007. *Publish or Perish*. https://harzing.com/resources/publish-or-perish. Accessed 8 March 2022.

Heinrich, P., and Schwabe, G. 2014. "Communicating Nascent Design Theories on Innovative Information Systems through Multi-grounded Design Principles," in *Proceedings of the 9th International Conference on Design Science Research in Information Systems and Technology*, Miami: USA, pp. 148-163.

Herwix, A., and zur Heiden, P. 2021. "Context in Design Science Research: Taxonomy and Framework," in *Proceedings of the 55th Hawaii International Conference on System Sciences*, Hawaii: USA.

Hevner, A. 2007. "A Three Cycle View of Design Science Research," *Scandinavian journal of information systems* (19:2), pp. 87-92.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly: Management Information Systems* (28:1), pp. 75-105 (doi: 10.2307/25148625).

Houkes, W. 2009. "The Nature of Technological Knowledge," in *Philosophy of Technology and Engineering Sciences : Handbook of the Philosophy of Science*, A. Meijers (ed.), Amsterdam: North-Holland, pp. 309-350.

Iivari, J., Hansen, Perlt, Magnus Rotvit, and Haj-Bolouri, A. 2018. "A Framework for Light Reusability Evaluation of Design Principles in Design Science Research," in *Proceedings of the 13th International Conference on Design Science Research in Information Systems and Technology*, Chennari: India.

Iivari, J., Rotvit Perlt Hansen, M., and Haj-Bolouri, A. 2020. "A proposal for minimum reusability evaluation of design principles," *European Journal of Information Systems*, pp. 1-34 (doi: 10.1080/0960085X.2020.1793697).

Järvinen, P. 2007. "Action Research is Similar to Design Science," *Quality and Quantity* (41:1), pp. 37-54 (doi: 10.1007/s11135-005-5427-1).

Kim, H. 2010. "Effective organization of design guidelines reflecting designer's design strategies," *International Journal of Industrial Ergonomics* (40:6), pp. 669-688 (doi: 10.1016/j.ergon.2010.08.002).

Kuipers, T. A. F. 2013. "Philosophy of Design Research," in *New Challenges to Philosophy of Science*, H. Andersen, D. Dieks, W. J. Gonzalez, T.

Uebel and G. Wheeler (eds.), Dordrecht: Springer Netherlands, pp. 457-466.

Kundisch, D., Muntermann, J., Oberländer, A., Rau, D., Roeglinger, M., Schoormann, T., and Szopinski, D. 2021. "An Update for Taxonomy Designers - Methodological Guidance from Information Systems Research," *Business & Information Systems Engineering* (doi: 10.1007/s12599-021-00723-x).

Maedche, A., Gregor, S., Morana, S., and Feine, J. 2019. "Conceptualization of the Problem Space in Design Science Research," in *Extending the Boundaries of Design Science Theory and Practice – 14th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2019, Worcester, MA, USA, June 4–6, 2019, Proceedings. Ed.: B. Tulu*, Springer International Publishing, Cham, pp. 18-31.

Markus, M. L., Majchrzak, A., and Gasser, L. 2002. "A Design Theory for Systems That Support Emergent Knowledge Processes," *MIS Quarterly: Management Information Systems* (26:3), pp. 179-212.

McAdams, D. A. 2003. "Identification and Codification of Principles for Functional Tolerance Design," *Journal of Engineering Design* (14:3), pp. 355-375 (doi: 10.1080/0954482031000091095).

Meth, H., Mueller, B., and Maedche, A. 2015. "Designing a Requirement Mining System," *Journal of the Association of Information Systems* (16), pp. 799-837 (doi: 10.17705/1jais.00408).

Möller, F., Guggenberger, T., and Otto, B. 2020. "Towards a Method for Design Principle Development in Information Systems," in *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry*, S. Hofmann, O. Müller and M. Rossi (eds.), Cham: Springer International Publishing.

Möller, F., Schoormann, T., and Otto, B. 2021. "'Caution - Principle Under Construction' - A Visual Inquiry Tool for Developing Design Principles," in *DESRIST 2021*, Chandra Kruse, L., Seidel, S. (ed.), Kristiansand: Norway, 223-235.

Nickerson, R. C., Varshney, U., and Muntermann, J. 2013. "A Method for Taxonomy Development and its Application in Information Systems," *European Journal of Information Systems* (22:3), pp. 336-359 (doi: 10.1057/ejis.2012.26).

Niiniluoto, I. 1993. "The aim and structure of applied research," *Erkenntnis* (38:1), pp. 1-21.

Niiniluoto, I. 2014. "Values in design sciences," *Studies in History and Philosophy of Science Part A* (46), pp. 11-15 (doi: 10.1016/j.shpsa.2013.11.002).

Nowack, M. L. 1997. *Design guideline support for manufacturability*.

Purao, S., Bush, A., and Rossi, M. 2001. "Problem and Design Spaces During Object-oriented Design: An Exploratory Study," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Hawaii: USA, 1-9.

Purao, S., Chandra Kruse, L., and Mädche, A. 2020. "The Origins of Design Principles: Where do… they all come from?" in *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry*, S. Hofmann, O. Müller and M. Rossi (eds.), Cham: Springer International Publishing, pp. 183-194.

Ritchey, T. 2014. "On a Morphology of Theories of Emergence," *Acta Morphologica Generalis* (3:3), pp. 1-17.

Romme, A., and Endenburg, G. 2006. "Construction Principles and Design Rules in the Case of Circular Design," *Organization Science* (17:2), pp. 287-297 (doi: 10.1287/orsc.1050.0169).

Romme, G. 2003. "Making a Difference: Organization as Design," *Organization Science* (14:5), pp. 558-573 (doi: 10.1287/orsc.14.5.558.16769).

Roozenburg, N. F. M., and Eekels, J. 1995. "Product design: fundamentals and methods,"

Schoormann, T., Möller, F., and Hansen, M. R. P. 2021. "How do Researchers (Re-)Use Design Principles: An Inductive Analysis of Cumulative Research," in *DESRIST 2021*, Chandra Kruse, L., Seidel, S. (ed.), Kristiansand: Norway, 188-194.

Schryen, G., Wagner, G., Benlian, A., and Pare, G. 2020. "A Knowledge Development Perspective on Literature Reviews: Validation of a new Typology in the IS Field," *Communications of the Association for Information Systems* (46), pp. 134-186 (doi: 10.17705/1CAIS.04607).

Seidel, S., Chandra Kruse, L., Székely, N., Gau, M., and Stieger, D. 2017. "Design Principles for Sensemaking Support Systems in Environmental

Sustainability Transformations," *European Journal of Information Systems* (27:2), pp. 221-247 (doi: 10.1057/s41303-017-0039-0).

Simon, H. A. 1995. "Problem Forming, Problem Finding and Problem Solving in Design," *Design & systems* (3), pp. 245-257.

Simon, H. A. 1996. *The Sciences of the Artificial*, Cambridge, USA: MIT Press. Third Edition. ISBN: 0-262-69191-4.

Szopinski, D., Schoormann, T., and Kundisch, D. 2020. "Visualize Different: Towards Researching the Fit Between Taxonomy Visualizations and Taxonomy Tasks," in *Proceedings of the 15th International Conference on Wirtschaftsinformatik*, Potsdam: Germany.

van Aken, J. 2001. "Improving the relevance of management research By developing tested and grounded technological rules," *Eindhoven Centre for Innovation Studies Working Paper* (01:19).

van Aken, J., Chandrasekaran, A., and Halman, J. 2016. "Conducting and Publishing Design Science Research: Inaugural Essay of the Design Science department of the Journal of Operations Management," *Journal of Operations Management* (47-48), pp. 1-8 (doi: 10.1016/j.jom.2016.06.004).

van Aken, J. E. 2004. "Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules," *Journal of Management Studies* (41:2), pp. 219-246 (doi: 10.1111/j.1467-6486.2004.00430.x).

van Aken, J. E. 2005a. "Management research as a design science: Articulating the research products of mode 2 knowledge production in management," *British journal of management* (16:1), pp. 19-36.

van Aken, J. E. 2005b. "Valid knowledge for the professional design of large and complex design processes," *Design Studies* (26:4), pp. 379-404 (doi: 10.1016/j.destud.2004.11.004).

van den Akker, J. 1999. "Principles and Methods of Development Research," in *Design approaches and tools in education and training*, J. van den Akker, R. M. Branch, K. Gustafson, N. Nieveen and T. Plomp (eds.), Kluwer Academic Publishers, pp. 1-14.

van Strien, P. J. 1997. "Towards a Methodology of Psychological Practice: The Regulative Cycle," *Theory & Psychology* (7:5), pp. 683-700 (doi: 10.1177/0959354397075006).

Vom Brocke, J., Winter, R., Hevner, A., and Maedche, A. 2020. "Accumulation and Evolution of Design Knowledge in Design Science Research - A Journey Through Time and Space," *Journal of the*

Association for Information Systems (21), pp. 520-544 (doi: 10.17705/1jais.00611).

Wache, H., Möller, F., Schoormann, T., Strobel, G., and Petrik, D. 2022. "Exploring the Abstraction Levels of Design Principles: The Case of Chatbots," in *Proceedings of the 17th International Conference on Wirtschaftsinformatik (WI)*, Nürnberg: Germany.

Walls, J. G., G, J., Widmeyer, R, G., and El Sawy, O. A. 1992. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), 36-59 (doi: 10.1287/isre.3.1.36).

Webster, J., and Watson, R. T. 2002. "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Quarterly: Management Information Systems* (26:2), pp. xiii-xxiii.

Wright, G. H. von 1963. *Norm and Action: A Logical Enquiry*, Routledge and Kegan Paul.