Journal of the Association for Information Systems (2020) **21**(5), 1237-1278
doi: 10.17705/1jais.00636

RESEARCH ARTICLE

ISSN 1536-9323

Method Engineering as Design Science

Göran Goldkuhl[1], Fredrik Karlsson[2]

[1]Linköping University, Sweden / Uppsala University, Sweden, goran.goldkuhl@liu.se
[2]Örebro University, Sweden, fredrik.karlsson@oru.se

Abstract

In this paper, we motivate, devise, demonstrate, and evaluate an approach for the research-based development of information systems development methods (ISDMs). This approach, termed "method engineering as design science" (ME-DS), emerged from the identified need for scholars to develop ISDMs using proper research methods that meet the standards of both rigor and relevance. ISDMs occupy a position of central importance to information systems development and scholars have therefore invested extensive resources over the years in developing such methods. The method engineering (ME) discipline has developed different frameworks and methods to guide such development work and, for that purpose, they are well-suited. Still, there remains a need for applications and evaluations of ISDMs based on the demands for knowledge justification. Unfortunately, in many cases, scholars come up short with regard to how ISDMs are generated and empirically validated. While design science (DS) stresses knowledge justification, prominent DS approaches seem to be biased toward the development of IT artifacts, making this approach ill-suited for the development of method artifacts. We therefore propose eight principles that marry ME and DS, resulting in a process model with six activities to support research-based development of ISDMs. We demonstrate and evaluate ME-DS by assessing three existing research papers that propose ISDMs. These retrospectives show how ME-DS directs attention to certain aspects of the research process and provides support for future ISDM development.

Keywords: Method, Information Systems Development Method, Design Science, Research Approach

Jan Pries-Heje was the accepting senior editor. This research article was submitted on September 13, 2018 and underwent one revision.

1 Introduction

Information systems development methods (ISDM) are important in information systems (IS) practices and there is great interest in such methods within the IS discipline. These methods can take the form of *objects* of IS research—e.g., in the evaluation of their applications. However, ISDMs can also take the form of *results* of IS research, a type of research that can be pursued in different ways. As a response to demands for well-thought-out ISDM development, a group of approaches exists under the concept of method engineering (ME) (Bergstra, Jonkers, & Obbink, 1985; Brinkkemper, 1996). ME is intended to be used for developing and adapting ISDMs in both research and practice.

The ME discipline has developed different frameworks and methods (often called metamethods) to guide the development and adaptation of ISDMs and ME metamethods are well suited for such purposes. Nevertheless, in ME research, there is still a need for applications and evaluations of ISDMs that are based on demands for knowledge justification, which entails the use of proper empirical research methods. However, Section 2.3 shows that, to a limited extent, research literature claiming to develop or adapt ISDMs appears to have used both ME metamethods and proper research methods for empirical inquiries. Obtaining high-quality ISDMs that are properly evaluated in terms of their practical applicability requires adequate knowledge support, such as ME metamethods and empirical research methods.

The introduction of the design science (DS) approach has had a profound effect on IS research (Hevner et al., 2004; March & Smith, 1995). However, the DS approach seems to be mainly oriented toward the development of new IT artifacts, although DS advocates also propose methods, models, and constructs as possible artifact outcomes (Hevner et al., 2004; March & Smith, 1995). Thus, the research-based development of ISDMs should be seen as one type of DS in IS. However, as discussed in Section 3.3, DS frameworks and process models appear to be mainly focused on the development of IT artifacts and are not specifically adjusted for ISDMs. This might explain why few ISDM development studies self-identify as DS, as will be further explored in Section 2.3.

Bucher and Winter (2008) investigated early DESRIST conference papers (2006-2007) and did not find many papers oriented toward the design of ISDM as the principal research outcome, and further found the influence of ME on DS studies to be minimal. They concluded: "According to our perception, method engineering is closely related to the [design research] paradigm insofar as it represents a distinct research field of design research for information systems. We strongly believe that both [design research] for IS (as a superordinate research paradigm) and ME (as a subordinate research field to DR for IS) stand to benefit from a two-way comparison and mutual transfer of knowledge" (Bucher & Winter, 2008, p. 46). Although this study is over a decade old, their conclusions are still valid. An explicit integration of ME and DS is still lacking, regardless of the existing work seeking to relate the two areas (e.g. Offermann, Blom, Bud et al., 2010).

There seems to be potential for applying ME frameworks and methods in DS studies; similarly, there appears to be potential for applying DS frameworks and methods in ME. A marriage between these two domains would be beneficial and fruitful to both disciplines and this paper contributes to this endeavor. We elaborate a research approach building on both ME and DS, aiming to detail how to conduct development and adaption of ISDMs through research. We thus integrate knowledge from both disciplines into one congruent research approach. Beyond ME as DS, ME also includes practitioner-driven ME. Likewise, DS goes beyond DS as ME and also comprises the design of other kinds of artifacts. A thorough discussion of different artifacts will be pursued in Sections 3.2 and 3.3. We label our proposed research approach "method engineering as design science" (ME-DS) because ISDMs and ME are the core objects focused on in this paper.

As a practice, ME-DS would produce ISDMs grounded in both empirical data and a scholarly knowledge base through extant theories and other knowledge artifacts. This means that ME-DS complies with the principles of multigrounded DS (Goldkuhl, 2004; Goldkuhl & Lind, 2010; Kuechler & Vaishnavi, 2012). Thus, ME-DS represents a knowledge contribution to both the ME and DS domains. Specifically, it contributes to ME through explicating how such endeavors can be pursued as DS studies. Similar to many DS-approaches we present ME-DS as an open structure, which, in our case, would be capable of being populated with more specific ME strategies. ME-DS contributes to DS through clarifying how to design ISDMs as method artifacts. It also contributes to DS by demonstrating how the design of an ISDM differs from the design of an IT artifact.

We conduct our research as an inquiry in the pragmatist sense (Dewey, 1938). It begins with a problematic situation and progresses toward a proposed solution through knowledge development. As discussed above, the problematic situation indicates that (1) an explicit DS approach is seldom applied in ME studies, and (2) DS does not seem to be sufficiently adapted to the development of ISDM as a resulting artifact. These concerns will be further investigated in Sections 2 and 3. In Section 2, we clarify the notions of ISDM and ME and investigate existing ISDM development research that employs research methods, which indicates the possibility of developing ME oriented toward DS. In Section 3, we turn to DS and investigate the artifact notion as a foundation for DS-oriented research and problematize different types of artifacts in DS. We show that the main DS models appear to have a bias toward the IT artifact as the primary design outcome, which suggests the possibility of developing DS oriented toward ME. Section 4 presents the ME-DS approach through eight principles and a process model. Section 5 demonstrates ME-DS retrospectively based on three existing and published cases of ISDM development research, with the aim of testing the applicability of the ME-DS principles and their usefulness for assessment. A diagnostic subpurpose of the demonstration, discussed in this section, illustrates how these cases could be improved by ME-DS. The paper concludes with a discussion and conclusions presented in Section 6. In this section, we also justify how our pragmatic inquiry, as a research approach, can be seen as an application of DS methods.

2 Research on Information Systems Development Methods

2.1 Information Systems Development Methods

Over the years, many scholars have tried to define the concept of ISDMs as used within the IS field (e.g. Brinkkemper, 1996; Checkland, 1981; Goldkuhl, Lind, & Seigerroth, 1997; Rumbaugh, 1995; Russo & Stolterman, 2000). Although they all define the concept slightly differently, Karlsson and Ågerfalk (2004) conclude that there seems to be a common understanding that an ISDM comprises three interrelated parts. First, an ISDM includes a process description that informs developers of which activities should be carried out. For example, if developers create a state chart, they need to identify the initial and final states of the object and identify the stages the object might undergo during its life. Second, an ISDM includes some sort of notation documenting the results of the activities. For example, a rectangle with rounded corners represents a state in a state chart. Finally, an ISDM includes a set of concepts used to describe the problem domain and the method itself. For example, a start stage defines the first step of a process in an IS.

A more inclusive view of ISDM acknowledges that scholars occasionally include other parts within the ISDM concept. Jayaratna, Holt, and Wood-Harper (1999) argue that in order for an activity set to be considered as a method, the rationale of those activities cannot be implicit. We find similar thoughts presented in Brinkkemper (1996) and Russo and Stolterman (2000), suggesting that the method's rationale is an important part of the method (Rossi et al., 2004). Goldkuhl et al. (1997) identified framework and cooperation form as two method parts. A framework contains the method's overall structure, such as the waterfall (Royce, 1970) or the spiral model (Boehm, 1988); the cooperation form involves how actors collaborate during development work through, for example, interviews, seminars, and workshops (Lind, 2001). Furthermore, Nilsson (1995) included the interest group model as a method part. We do not claim that these method parts represent a complete inventory; rather, they serve to illustrate that ISDMs can take slightly varied forms and levels of coverage, compared to the core discussed above.

2.2 Method Engineering

The discipline of developing ISDMs has been recognized in its own right (Rossi et al., 2004) as something distinct from but related to information systems development (ISD). Instead of focusing on ISD through the use of an ISDM, this discipline focuses on the artifacts that support ISD, such as ISDMs. This metalevel of ISD, which was first introduced by Bergstra et al. (1985). Later, Kumar and Welke (1992) referred to this discipline as methodology engineering. However, while van Slooten and Brinkkemper (1993) argue that this discipline should be called method engineering, Brinkkemper (1996, p. 276) defines it as "the engineering discipline to design, construct, and adapt methods, techniques and tools for the development of information systems." However, Henderson-Sellers et al. (2014) clarify that method engineering is now a generally accepted term.

Of course, there is no silver bullet for a task as complex as ME and scholars have proposed different ME metamethods for developing and adapting ISDMs (e.g. Bajec, Vavpotič, & Krisper, 2007; Cameron, 2002; Cervera, 2015; Harmsen, 1997; Karlsson & Ågerfalk, 2009; Ralyté & Franch, 2018; Ralyté & Rolland, 2001; Sandkuhl & Seigerroth, 2019). Although these metamethods differ, they share some fundamental ideas. One of the central ideas is the method part, a small part of an existing method or method-to-be (Henderson-Sellers et al., 2014), which is used to construct, extend, or reduce an ISDM. The collection of these parts is called a method base (Punter & Lemmen, 1996) and is often stored in a repository. The method part implies that standardized formats are used, such as method fragment (Harmsen, Brinkkemper, & Oei, 1994), method chunk (Rolland & Prakash, 1996), or method component (Karlsson & Wistrand, 2006). Method parts can be elicited from existing ISDMs, reconstructed from practice, or generated from existing theories and new ideas. Rolland and Prakash (1996) stress that to design, construct and adapt ISDMs, it is necessary to describe the ISD situation in which a method part is relevant. For that purpose, previous studies have proposed the use of method rationale (Oinas-Kukkonen, 1996; Rossi et al., 2004; Ågerfalk & Wistrand, 2003).

ME is carried out, implicitly or explicitly, in many organizations when developing or adapting ISDMs to current ISD project needs. In these circumstances, ME often exists as part of organizational learning (Henderson-Sellers et al., 2014) and might therefore be described as an activity integrated with ISD, which has been previously characterized as evolutionary ME (Rossi et al., 2004; Tolvanen, 1998). ME is also carried out as a scientific activity, which we focus on in this paper. As a scientific activity, produced ISDMs must meet scientific requirements of rigor and relevance. Hence, the use of research methods in ME should be of great importance when generating and validating the proposed designs.

2.3 Research Methods Used in Development of Information Systems Development Methods

To contextualize this study, we analyzed 53 of the most cited studies relating to the development of ISDMs published in international journals, at international conferences, and in book chapters between 1992 and 2017. The analysis identified the research methods that were used to generate and validate ISDMs in existing studies proposing ISDMs. We analyzed the authors' description of the research method used and interpreted the actual research method used based on the content of the paper. We present the details of our analysis in Appendix A and offer an overview in Table 1. The left-hand column presents the research method, using an extended version of Mingers's (2003) classification framework, the middle columns show how scholars described their research method, and the right-hand columns show our interpretation of the actual research method used.

Our first observation is that few studies have applied DS to generate and validate ISDMs. In total we identified two studies (D'Aubeterre, Singh, & Iyer, 2008; van de Weerd et al., 2006); in both cases the described research method and the actual research method aligned. We identified three additional studies (Champion & Stowell, 2002; Pilemalm & Timpka, 2008; Vidgen, 2012) that employed action research, a research method that shares several basic tenets with DS. This indicates that the vast majority of the studies are based on research methods other than DS for generating and validating designs of ISDMs.

Our analysis of this latter group of papers shows that in five papers (Cossentino & Seidita, 2005; Rahimian & Ramsin, 2008; Spanoudakis & Moraitis, 2008; van de Weerd et al., 2006; Weigand & de Moor, 2003) the authors described employing (situational) ME metamethods to generate ISDMs. The actual use of ME metamethods seems to be somewhat higher. We identified nine papers (Cossentino & Seidita, 2005; DeLoach & Valenzuela, 2007; Georg et al., 2015; Kavakli et al., 2006; Rahimian & Ramsin, 2008; Spanoudakis & Moraitis, 2008, 2011; van de Weerd et al., 2006; Weigand & de Moor, 2003) that used metamethods to generate ISDMs. Hence, we conclude that although metamethods focus solely on the development and adaption of ISDMs, these methods have had little impact on work within the research discipline itself.

Most of the papers—44 of them (e.g. Hirschheim & Klein, 1994; Mouratidis & Giorgini, 2007; Reinhartz-Berger, Dori, & Katz, 2002; Scerbo et al., 2011)—include no description of the research method that generated the ISDM. This is problematic because it makes it difficult to assess, compare, and replicate studies. Further investigation of what was actually done revealed that most of these ISDMs were generated based on a subjective argumentative research approach—that is, these ISDMs are based on reasoning. We found that 37 studies proposed an ISDM in this manner. Examples of such studies include Ahituv and Neumann (2002) and Ayed et al. (2010).

Table 1. Overview of Described and Actual Research Method

	Described research method		Actual research method	
Research method	**Generating**	**Validating**	**Generating**	**Validating**
Action research	3	3	3	3
Case study	0	21	1	14
Consultancy	0	0	0	0
Critical theory	0	0	0	0
Design science	2	2	2	2
Ethnography	0	0	0	0
Experiment	0	2	0	16
Grounded theory	0	0	0	0
Interviews	0	1	0	0
Literature review	0	0	0	0
No research method mentioned	44	25	n/a	n/a
Observation	0	0	0	0
Qualitative content analysis	0	1	0	1
Simulation	0	0	0	1
(Situational) ME method	5	0	9	0
Subjective/argumentative	0	0	37	0
Survey	0	1	0	1
Unable to classify	n/a	n/a	6	18

Note: a research study can employ more than one research method. Therefore, the number of research methods does not equal the number of studies

Furthermore, Table 1 shows that 25 papers lack ISDM validation and hence a description of the research method to assess the ISDM's usefulness (e.g. Bayuk & Horowitz, 2011; Fernandez, Cholmondeley, & Zimmermann, 2007; Soylu & De Causmaecker, 2009; Zhuge, 2003). This is problematic because the consequences of the ISDM were not tested in these papers. Nevertheless, these scholars seem to prioritize validation of the ISDM over how they arrived at the design, as 25 papers lack a description of how the ISDMs were validated, compared to the 44 papers lacking a description of how the ISDMs were developed. Of the studies that include validations, the case study method was by far the most common type of validation, with 21 studies claiming its use (e.g. Ge et al., 2006; Georg et al., 2015; Lee & Lee, 2008; Savage & Mingers, 1996). When investigating actual use, however, we found that only 14 studies were indeed case studies (e.g. Beynon-Davies & Holmes, 1998; Galal & Paul, 1999; Georg et al., 2015; Lee & Lee, 2008) in which the validation of the ISDM was linked to an actual context such as an organization or a project. Artificially generated situations are commonly described as case studies, even though it is more appropriate to classify them as experiments. As shown in the right-hand column of Table 1, we found that 16 studies employed experiments (e.g. Gustas & Gustiené, 2008; Ingham et al., 2006; Osmundson, 2000; Siau & Tan, 2005) as the actual research method to validate the ISDM.

To summarize, existing high-impact studies that propose an ISDM as the main research outcome pay relatively scant attention to research methods. Indeed, as Table 1 shows, the development of ISDMs can almost be described as amethodical. We conclude that proposed ISDMs have been validated in empirical settings to only a limited extent. However, what is even more striking is the common black-boxing scholars employ regarding how they arrive at the ISDM, especially considering the fact that several methods for ME have been suggested that focus on this aspect. This may be explained by the fact that although the debate on research methods is very much alive (e.g. Venkatesh, Brown, & Bala, 2013; Ågerfalk, 2013), research methods capable of *both* generating and validating ISDMs that support both rigor and relevance have received limited attention. This is somewhat surprising, given that DS, which emphasizes both the generating and validating aspects of design, has been the subject of increasing attention in the IS discipline. However, our review clearly indicates that DS as a research approach has been explicitly used for proposing ISDMs to only a limited extent.

3 Artifacts in Design Science

The artifact concept is central to DS, as it is seen as the key outcome (Hevner et al., 2004). In this section, we investigate what an artifact is and how a method such as an ISDM can be seen as an artifact.

3.1 What is an Artifact?

An artifact is a human-made object, in contrast to a natural object. Etymologically, "artifact" has its origins in the Latin "*arte*" ("by skill") and "*factum*" ("thing made", from "*facere*" = "to make, do"). The concept is used in many disciplines, such as archaeology (Binford, 1962), anthropology (Henare, Holbraad, & Wastell, 2007), engineering (Maier & Fadel, 2009), design studies (Crilly, 2010; Rosenman & Gero, 1998) and philosophy (Dipert, 1995; Verbeek, 2005). In these disciplines, artifacts are mainly, and often implicitly, conceived of as physical things. Many contemporary artifact inquiries are oriented toward technical artifacts as a subclass of physical artifacts (e.g. Franssen et al., 2014). Fundamental to artifacts is their functional or instrumental character (Crilly, 2010; Dipert, 1995; Rosenman & Gero, 1998). Artifacts are created to support people in reaching their purposes (Crilly, 2010; Dipert, 1995; Rosenman & Gero, 1998; Simon, 1996); they are considered to be objects and, as such, they have their own separate existence.

Most artifact theorists, for example those referenced above, seem to focus on physical artifacts and do not include other types of objects in their definition. That said, other scholars (e.g. Beckman, 2002; Simon, 1996) have included semiotic objects as one subclass of artifacts, acknowledging works in which language is the primary medium, like books, papers, recordings, etc. These symbolic expressions rely on some materiality, such as paper, magnetic tape, or optical discs, for their existence but these physical properties do not represent their primary function. Rather, their primary function, as signs, is to inform/entertain readers/listeners. These objects are sometimes called sign artifacts, although this concept is not widely used in linguistics and semiotics—Bernard (2009) proposes the concept of a signifact. Although sign objects may not be primarily physical objects, they fulfill the criteria of being a typical artifact; they have a separate existence and are intentionally created by humans based on some purpose and some intended social use.

There are also even broader conceptualizations of artifacts that encompass anything created by humans (Dahlbom, 2002), rather than merely "artificial" things. In his rather inclusive view of artifacts, Beckman (2002, p. 56), however, restricts the artifact to "a humanly designed, socially objectified vehicle of functional meaning". He demands an artifact be "socially objectified," meaning that it needs to be recognizable to more than one person and based on some intersubjective meaning concerning function and use. According to this view, an artifact is an object, but not necessarily a physical object. An inclusive artifact notion creates opportunities for theories and ideologies as artifacts (Beckman, 2002; Dahlbom, 2002).

More typical artifact		Less typical artifact
Fixated (distinct, separate, recognizable)	↔	Fluid
Made through deliberation	↔	Occurring serendipitously
Designed, manufactured	↔	Evolving
Purpose-given (functional meaning)	↔	Unclear and varying uses
Inter-subjectively recognized	↔	Personally held
Enduring, reusable	↔	Temporary
Entity-like (typical noun)	↔	Process-like (typical verb)
Material	↔	Immaterial

Figure 1. Artifact Characteristics

It is possible to take a position that an artifact can be immaterial, in the sense of being essentially knowledge, if it fulfills most accepted criteria of artifacts, i.e., a deliberately constructed and intersubjectively identifiable object with enduring existence and with clear purposes and intended uses. For example, many theories and methods could fulfill such criteria. Thus, beyond physical artifacts and sign artifacts, we can add knowledge artifacts as a third type, also termed mentefacts by Bernard (2009). Nevertheless, not just any type of knowledge counts as an artifact. A knowledge artifact must be an intersubjective and clearly demarcated knowledge assemblage based on deliberate design rather than serendipitous evolution.

Figure 1 summarizes what we interpret as criteria for something to be considered an artifact, based on the artifact-theoretical literature discussed above. Generic artifact properties are described in the figure's left part; the right part contains their opposites. These polarities characterize artifacts in the following way: the more properties on the left, the more typical the artifact; the more properties on the right, the less typical the artifact. Introducing these characteristics imbues the notion of artifact with a more palpable meaning and aids in positioning ISDM as a knowledge artifact.

This analysis of the artifact concept results in three artifact types—physical artifacts, sign artifacts, and knowledge artifacts—i.e., the ideal artifact types. However, artifacts may also be a mixture of these types. Moreover, these three types are related to each other. Many physical/technical artifacts have semiotic inscriptions, such as instructions for use. These should not be taken as key properties of physical artifacts but as auxiliary properties for guiding users of the artifact. Physical artifacts are created through knowledgeable actions; in a sense, they encapsulate knowledge but are not knowledge as such. To establish a knowledge assemblage as an intersubjectively recognized knowledge artifact with an enduring existence, such artifacts need to be expressed in ways that promote the artifact's existence, i.e., through the use of communication in sign artifacts.

Some objects are mixtures of technical and sign artifacts—exemplified, for example, by an IT artifact, comprising hardware, software, and digitalized information. The hardware is the technical foundation for such an artifact. The software of the IT artifact has a dual nature. While to the designer the software embodies signs, with the source code being a set of rules to create desired IT artifact behavior, as translated machine code, the software also comprises a direct part of the technical artifact in that it operates as a program for the behavior of the IT artifact. Thus, the IT artifact's software can be said to have a dual nature. Finally, digitalized information contents in digital storage systems and on user interfaces are sign artifacts carried by the information technology.

We characterize a method as basically a knowledge artifact. As described in Section 2.1, an ISDM comprises process descriptions, notational rules, and a set of concepts. All these parts represent clear examples of knowledge. However, methods as socially recognized knowledge artifacts must build on method knowledge representations in appropriate documentation—that is, there is a need for sign artifacts, or ISDM descriptions, that encapsulate the method knowledge. In the following section, we explore ISDMs as artifacts and DS study outcomes.

3.2 The Artifact Concept in Design Science

Through a close and hermeneutic reading of pertinent DS publications, we have attempted to articulate a clearer view of how the artifact notion is conceived within the DS community. This work has been challenging because of obscurities and confusion in the literature. Iivari (2015, p. 107) accused "the scientific discourse on DSR [of being] in a state of conceptual confusion." Two of the most prominent DS advocates have directly acknowledged this kind of confusion: "We contend that ongoing confusion and misunderstandings of DSR's central ideas and goals are hindering DSR from having a more striking influence on the IS field" (Gregor & Hevner, 2013, p. 338).

As discussed earlier, there are four types of artifacts identified as DS main outcomes: constructs, models, methods, and instantiations (Hevner et al., 2004; March & Smith, 1995). *Constructs* are defined as "concepts" and "conceptualizations" (March & Smith, 1995) and as "vocabulary and symbols" and "the language in which problems and solutions are defined and communicated" (Hevner et al., 2004, p. 78). March and Smith (1995, p. 256) explicitly position constructs within an ISD context: "Such constructs may be highly formalized as in semantic data modelling."

Models, according to Hevner et al. (2004, p. 78-79): "use constructs to represent a real world situation—the design problem and its solution space. … Models aid problem and solution understanding and frequently represent the connection between problem and solution components enabling exploration of the effects of design decisions and changes in the real world." The model concept covers obviously different kinds of models produced during ISD.

A *method* is defined by March and Smith (1995, p. 257) in an unfortunate and ambiguous way as "a set of steps (an algorithm or guideline) to perform a task" (March & Smith, 1995, p. 257). March and Smith further locate the functions of algorithms in an automated context and they exemplify ISDMs as process guidelines. As also described by Hevner et al. (2004, p. 79), this means that the method concept covers both ISDMs and automated processes: "Methods define processes. They provide guidance on how to solve problems, that is, how to search the solution space. These can range from formal, mathematical algorithms that explicitly define the search process to informal, textual descriptions of 'best practice' approaches, or some combination." We interpret the use of "method" as designating rules and guidelines for the performance of some kind of process, which is in line with the etymological meaning of the word "method" as "following a way," derived from the ancient Greek "methodos." However, in an IS context, not clearly distinguishing between an ISDM and the algorithms expressed in software products creates ambiguities, which will be further discussed below.

Instantiation is probably the most salient artifact type. Hevner et al. (2004, p. 82) define an instantiation as an IT artifact. An instantiation covers the other three artifact types described by March and Smith (1995, p. 258): "instantiations operationalize constructs, models, and methods." Similarly, according to Hevner et al. (2004, p. 79): "Instantiations show that constructs, models, or methods can be implemented in a working system." This implies that the main artifact from DS, the IT artifact expressed as an instantiation, comprises the other three types of artifacts. The four artifact types could, of course, be interpreted as *subclasses* of the artifact concept. However, there is also another type of relationship between these concepts; construct, model and method are all *subparts* of an instantiation. Accordingly, they do not represent four subclasses on a similar footing. The instantiation type, i.e., the IT artifact, *encapsulates* the other three.

Given the above discussion, as an artifact type, the ISDM represents a deviation of sorts because it does not fit neatly into the typology of March and Smith (1995) and Hevner et al. (2004). In addition, studies by both Sein et al. (2011) and Kuechler and Vaishnavi (2012), place the IT artifact in the foreground when reasoning about the resulting DS artifact; they explicitly refer to the ensemble view of the IT artifact as conceptualized by Orlikowski and Iacono (2001).

We were unable to find any theoretical or empirical grounding for the four artifact types proposed by March and Smith (1995). They seem to be purely idea based. Offermann, Blom, Schönherr et al. (2010) investigated the use of artifact types in DS studies and concluded that many scholars seem to use the March and Smith (1995) typology in a fairly unreflective way. Moreover, Offermann, Blom, Schönher et al. identified a set of artifacts that differ from this typology; their alternative typology was inductively generated from conducted and published DS studies. They propose the following artifact types: system design (description), requirement (statement), method, algorithm, pattern, guideline, language/notation, and metric. These new constructs have not been mapped onto the typology of March and Smith but there are important issues to note regarding ISDMs.

Offermann, Blom, Schönherr, et al. (2010) distinguish between method and algorithm, which seems important. Above, we noted the conflation in the March and Smith (1995) typology of method and algorithm, with which Bucher and Winter (2008) also confer. There are other artifacts in the Offermann, Blom, Schönherr, et al. typology that directly relate to the method notion. A guideline, as "a generalized suggestion about system development" (Bucher & Winter, 2008, p. 84), should be viewed as a type of method or at least as part of a method. Language/notation as "modeling elements and rules how these elements can be related" (Bucher & Winter, 2008, p. 84), should definitely be taken as one element of a method. A metric, as a "model that is used to evaluate aspects of a system design" (Bucher & Winter, 2008, p. 84), could also be interpreted as a special type of method. Also, Wieringa (2014, p. 29) offers a broad list of possible artifact outcomes of DS: "algorithms, methods, notations, techniques, and even conceptual frameworks."

As such, Offermann, Blom, Schönherr, et al.'s (2010) attempt to offer alternative artifact conceptions in DS is important. They acknowledge the specific characteristics of methods, characteristics that the often referred to, but overly restricted typology by March and Smith (1995) misses.

3.3 Design Science Process Models and Resulting Artifacts

Several process models exist for DS research (e.g. Kuechler & Vaishnavi, 2012; Peffers et al., 2007; Sein et al., 2011). These models are primarily adapted to developing an IT artifact as the main option. Peffers et al.'s (2007, p. 55) contention that "a design research artifact can be any designed object in which a research contribution is embedded in the design" seems to have a specific IT artifact in mind: "this activity includes determining the artifact's desired *functionality* and its *architecture* and then creating the actual artifact" (Peffers et al.; our emphasis).

DS process models appear to be generally adapted from ISD models and partially from general design models (e.g. Takeda et al., 1990), with some added research "ingredients." For example, in Peffers et al.'s (2007) model, there are two steps (evaluation and communication) that relate more clearly to research than design. Evaluation is concerned with assessing system characteristics, often based on some demonstration. Communication involves disseminating results to scholars and other audiences. The other steps can be mapped fairly well onto ordinary ISD processes.

The process models mentioned above can be characterized as *linear* and *one-layered models* acknowledging the iterative nature of the design research process. The Kuechler and Vaishnavi (2012) model is part of a larger framework including theory development but the interaction between the designing and theorizing processes is not made explicit in their DS process model. We view their addition of a theory development process as being in opposition to the original DS position of March and Smith (1995) and Hevner et al. (2004) who place the development and justification of theories outside of DS. However, there are many scholars who acknowledge the importance of theory development in relation to DS (Gregor & Jones, 2007; Lee, Pries-Heje, & Baskerville, 2011; Venable, 2006), and several two-layered models (e.g. Goldkuhl & Lind, 2010; Lee et al., 2011; Winter, 2014) stress a continual alternation between an abstract knowledge layer and a concrete design layer. Such models show not only a *progression* (as in one-layered linear models) but also an *alternation* between abstract and concrete layers.

The bias toward IT artifacts as the primary artifact outcome is found in several other influential DS publications. In an *MIS Quarterly* editorial, Goes (2014) reviewed published articles in DS, calling for more efforts in publishing such research. He characterizes design science in an IT-artifact-centric manner: "the research paradigm is about problem solving; it is about presenting solutions through systems and IT artifacts" (Goes, 2014, p. iv). In a *Journal of the Association for Information Systems* editorial, Baskerville et al. (2018) elaborate on the balancing between artifact and theory in DS studies and publications, making a strong positioning statement: "a novel IT artifact must be built and evaluated in a DSR project" (p. 359). In a *European Journal of Information Systems* editorial, Peffers, Tuunanen, and Niehaves (2018) suggest the use of different genres to describe and evaluate DS research. They propose five such DS genres, which can, however, be grouped into two main genres: one design theory oriented and one focused on concrete development artifacts; none of these genres has a specific focus on ISDMs. Iivari (2015) identifies two research strategies in design science: one laboratory oriented and one practice based. While the results generated by these two strategies may differ, Iivari denotes results from both strategies as "a real system implementation" (p. 110).

There are several examples of DS processes given in the literature. However, most of these case descriptions appear to be IT-artifact-centric examples. This is clearly the case for the studies by Hevner et al. (2004) and Sein et al. (2011). Peffers et al. (2007) present four examples: three are IT-artifact-centric cases and one case is from the development of an IS planning method. Although they offer one example from the ISDM domain, their process model does not seem to be adapted to this type of artifact but rather takes IT artifact as the key type of artifact. The dominant role models for artifact types in DS are thus IT artifacts rather than ISDMs.

We conclude that DS process descriptions are not specifically adapted to ISDMs as resulting artifacts. Instead, conceptualizations of DS tend to be IT-artifact-centric. This also corroborates Bucher and Winter's (2008) conclusions following their investigation of the first two years of DESRIST conference papers (p. 46):

> *we were surprised to see that the "method" artifact takes the center stage of DR publications less frequently than any other DR artifact. Moreover, we were amazed at the nature and characteristics of the methods which are described in those DESRIST publications which do take up this type of design product. Almost all papers that fall into the method category, describe the design and development of some kind of algorithm or mathematical/statistical technique*

This does not represent the usual character of many ISDMs and the main focus of ME studies.

To the conclusions above, we add that several DS publications exist with outcomes (e.g. ISDMs) other than IT artifacts. Influential DS publications (several which are mentioned above) are inclusive concerning artifact types, even though the primary focus does seem to be on IT artifacts. In the next section, we elaborate on what a DS process with a specific focus on ISDMs would look like.

4 How to Conduct Method Engineering as Design Science

4.1 How to View ISDM Research

Hevner et al. (2004) conceptualize IS research as interactions with a business environment and a knowledge base. Figure 2 presents a simplified version of their IS research model. The environment has three subparts: people, organizations, and technology. Hence, in their model knowledge is not defined as an explicit subpart; rather, knowledge seems to be assembled within the knowledge base. However, their discussion indicates that the knowledge base comprises scholarly knowledge; practical knowledge directly related to business practices is not part of this model.

ISDMs can take the form of research instruments and research objects in IS research endeavors and can also represent outcomes of such research. An ME study can use method parts from both practitioner-generated ISDMs and scholarly developed ISDMs. The generated method of an ME study may be aimed at practitioners or scholars, or at both groups. These distinctions are hard to detect in Figure 2. Therefore, Figure 3 proposes an alternative model.

In this alternative model, the knowledge base comprises two parts: a scholarly knowledge base (SKB) and a business practice knowledge base (BPKB). The SKB is generated by scholars through established scientific quality assurance processes, while practitioners generate knowledge considered useful and effective for business practice purposes. There is, however, an overlap between these knowledge bases: knowledge that is considered to be both of scholarly quality and useful for practical purposes. ISDMs are one example of such type of intersectional knowledge. In Figure 3, this model has been populated with ISDMs of different origins and intended uses. As discussed above, we treat ISDMs as knowledge artifacts. The different types of ISDMs, as intersubjective spheres, are elements in such knowledge bases and they must therefore be supported by proper documentation (as sign artifacts).

The five types of methods in Figure 3 have different origins and intended targets. Some ISDMs are developed by practitioners with intended use in business practices (Categories 1 and 2). Method Category 1 lacks any interference from scholars. If such an ISDM is studied by scholars and evaluated in some way, then it belongs to ISDM Category 2; however, this does not mean that the method has changed. Rather, the ISDM is a research object in such a research process and the research outcome is thus evaluative knowledge about this ISDM. In Category 3, the generation of an ISDM is pursued through cooperation between practitioners and scholars and this ISDM is intended for business practice. The role of practitioners participating in development work should be to ensure a practice-oriented ISDM. In contrast, the ISDM knowledge is part of the scholarly knowledge base since this should be the axiomatic target of scholarly work.

Scholars can also engage in the development of ISDMs with intended use in business practice without practitioners being involved in the development process (Category 4). Such an ISDM also belongs to both spheres of the knowledge base (BPKB/SKB). Historically, methods of this kind were often considered to be pure scholarly objects. This is the case for Method Category 5, in which the ISDM development is considered to be a kind of "internal academic exercise." ISDMs are generated by scholars but, at this stage, they have no broader ambitions than to contribute to a scholarly discourse. A Category 5 ISDM can later become an ISDM of Category 4 or even Category 3 if further development involves practitioners.

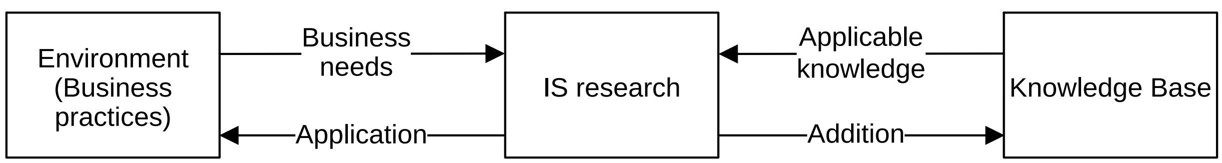

Figure 2. A View of IS Research (Simplified from Hevner et al., 2004, p. 80)

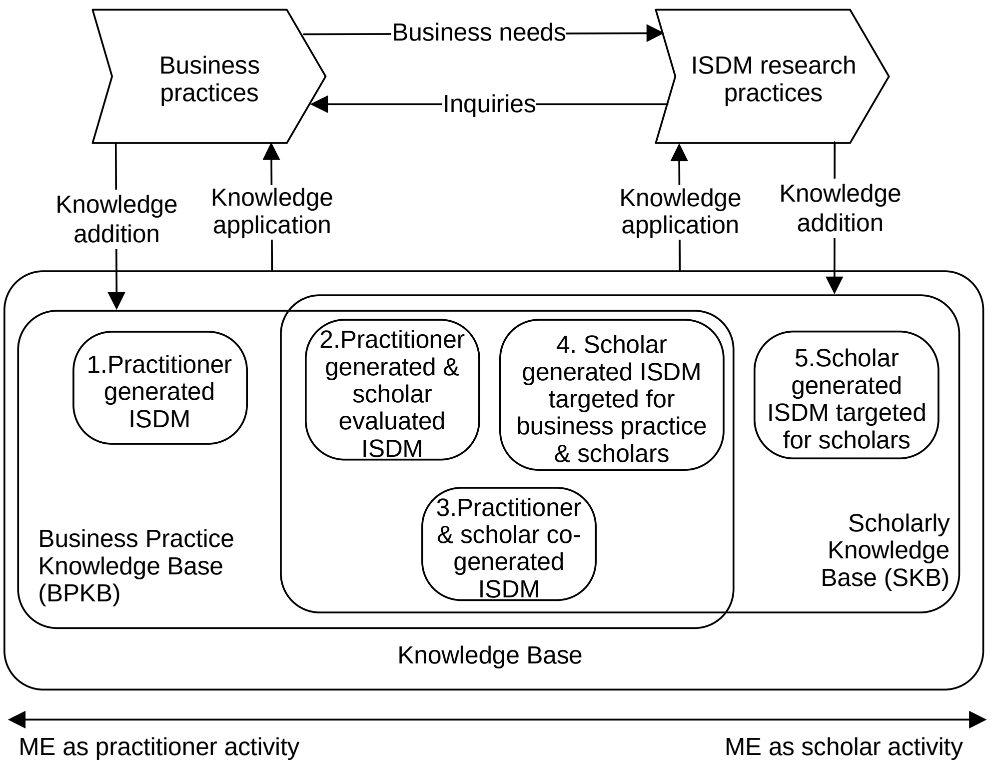

Figure 3. IS Research Model with a Knowledge Base and Five Types of ISDMs

4.2 Principles and Process Model for Method Engineering as Design Science

As knowledge artifacts produced by scholars, ISDMs should have a place in the SKB. However, merely presenting an ISDM is not a sufficient research process result. To earn its place in the SKB, an ISDM needs to be complemented by proper arguments. We term this complement the "method rationale." As discussed above, the concept of method rationale has been used by several ME scholars (e.g., Ågerfalk & Wistrand, 2003) as an explication of values and goals behind a method; it has also been used by Rossi et al. (2004) to indicate method changes and the reasons behind these changes. Following the concept of design rationale as arguments for a proposed design (MacLean, Young, & Moran, 1989), we use method rationale with an inclusive meaning comprising the arguments for a presented ISDM—from external sources such as the knowledge base and empirics and from internal sources, namely the ISDM itself. This conceptual determination of method rationale is based on the general meaning of rationale as the justification of something via the explanation of its underlying reasons. Arguments within a method rationale are generated, formulated, assessed, and documented together with the ISDM as intermediaries and final outcomes of a proper research process. The outcomes of an ISDM research process thus comprise an ISDM complemented with a method rationale.

Based on our discussion in previous sections, we have developed eight ME-DS principles that can act as a foundation for a research process. These principles are presented in Table 2. The left-hand column shows the name of the principle, the second column shows the parts of the ISDM design that the principle emphasizes, the third column presents the activities in the ME-DS process model, and the right-hand column offers references to earlier discussions in the paper. The eight principles emphasize the need for a research process that comprises a continual evaluation and justification of the ISDM design that "grounds" the method design. The first two principles concern relevance grounding, in which the practical relevance and the research relevance of developing a new ISDM are proposed. The third principle states that the ISDM should be grounded in an explicit set of values since an ISDM is a normative artifact. The fourth principle argues for the importance of grounding the ISDM in well-established concepts and conceptual frameworks. The fifth principle states that the ISDM should be in line with functional patterns in the existing knowledge base that have proven useful. The sixth principle stresses the importance of grounding the ISDM's applicability by proving its usefulness in ISD environments. The seventh principle concerns the internal grounding of the ISDM to ensure that the method parts constitute a harmonious whole. Finally, the eighth principle states that the ISDM should have a well-defined audience.

Table 2. ME-DS Principles

ME-DS principle	ISDM characteristics	ME-DS activity	References
Principle 1: Practical relevance	The ISDM should be a well-motivated response to practical ISD problems and needs.	Activity 1: Identify ISDM problem and motivate	Section 4.1
Principle 2: Research relevance	The ISDM should be a well-motivated response to research problems and needs.	Activity 1: Identify ISDM problem and motivate	Section 4.1
Principle 3: Value compliance	The ISDM should, as a normative artifact, be based on a perspective and its explicit values.	Activity 2: ISDM theorizing	Sections 2.2 and 3.3
Principle 4: Conceptual concordance	The ISDM should be in line with established concepts and conceptual frameworks in the knowledge base or be well-motivated deviations from them.	Activity 2: ISDM theorizing	Sections 2.1, 2.2, and 3.3
Principle 5: Functional pattern concordance	The ISDM should be in line with established functional patterns in the knowledge base or be a well-motivated deviation from them.	Activity 2: ISDM theorizing	Sections 2.1, 2.2, and 3.3
Principle 6: Practical applicability and usefulness	The ISDM's applicability and usefulness should be demonstrated and evaluated in relation to intended purposes.	Activity 4: ISDM demonstration, Activity 5: Formal evaluation	Section 2.3
Principle 7: Intelligible and harmonious whole	The ISDM should, as a knowledge artifact, have a well-thought-out structure, in which the collection of the method parts constitutes an intelligible and harmonious whole.	Activity 3: Method engineering	Sections 2.2, 3.1, and 4.1
Principle 8: Target audiences	The ISDM should be communicated to both scholars and practitioners (i.e., intended for SKB and BPKB).	Activity 6: Communication	Section 4.1

Based on these ME-DS principles, we designed a process model, illustrated in Figure 4, that can serve as a structure for accomplishing ME-DS. Building on influential prior research in ME and DS, we elicit appropriate process and product elements to implement these principles. Rather than emphasizing the nuanced differences in existing research, we sought to synthesize—that is, integrate and harmonize, process and product elements in existing approaches.

Table B1 in Appendix B contains process and product elements from influential ME and DS research. As seen in Appendix B, the ME and DS literatures have different strengths. While ME focuses more on the theorizing and the development of ISDMs and clearly distinguishes the design of an ISDM as a metalayer in relation to ISD, DS emphasizes the need to identify a research problem, to demonstrate the artifact, and to evaluate and communicate the results, revealing less about how to accomplish the actual design activity. As a synthesis, we suggest an iterative process model comprising six activities.

Activity 1: Identify ISDM Problem and Motivate. This activity, found on the left side of Figure 4, implements ME-DS Principles 1 and 2. Hevner et al. (2004) argue that DS should address relevant and important problems. Hence, there is a need to justify the research problem. Some scholars (Nunamaker, Chen, & Purdin, 1991; Walls, Widmeyer, & El Sawy, 1992) stress applied problems when justifying research problems; others (Archer, 1984; Eekels & Roozenburg, 1991) emphasize the theoretical aspects. ME-DS Principles 1 and 2 thus clearly suggest that the development of an ISDM should be justified from both practical and research perspectives. As shown in Figure 4, this also means that the idea of an ISDM can have two different starting points; it can be based on practical needs or on theoretical ideas. Regarding practical needs, ISD problems, needs, and ideas from ISD practice are used to motivate an ME-DS effort, where empirical data has been collected about current ways of working with ISD. These needs can be turned into a set of method requirements (Ralyté & Rolland, 2001) that represent an important part of the method rationale.

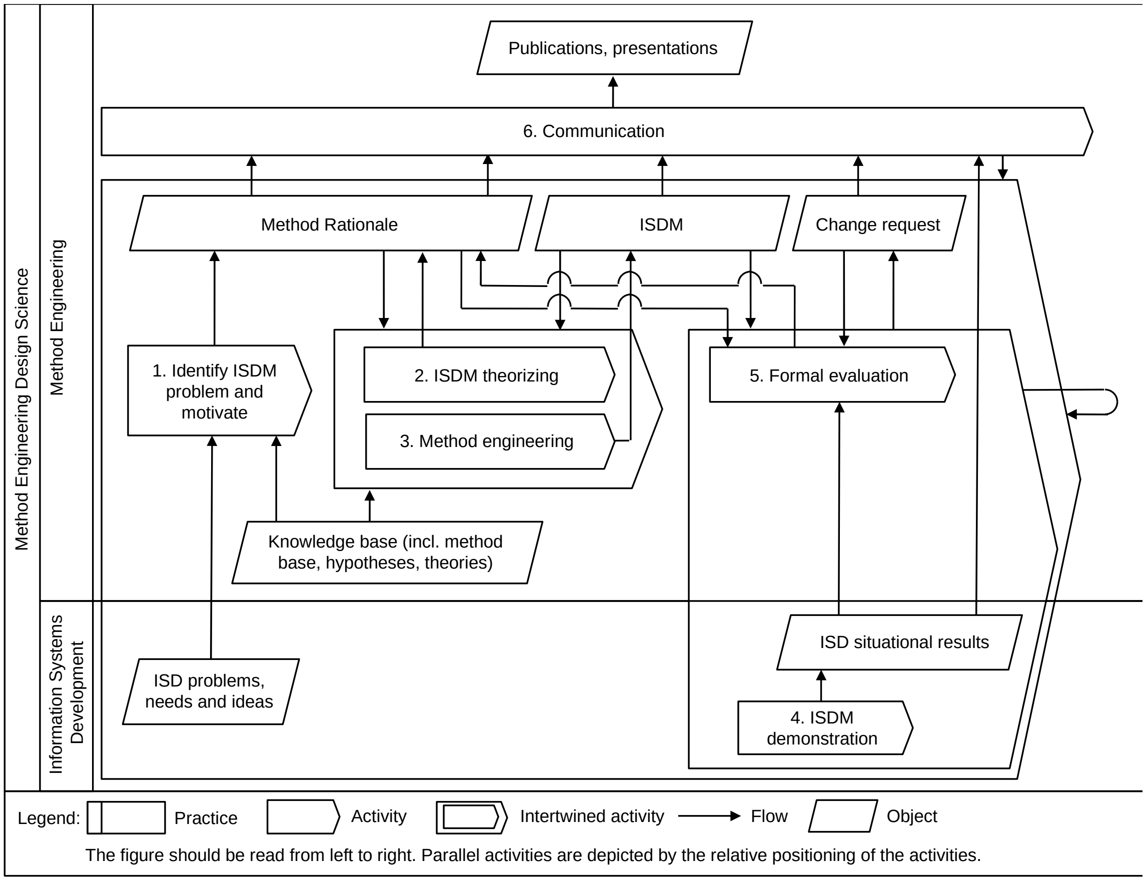

Figure 4. ME-DS Process Model

However, according to ME-DS Principle 2, merely identifying practical needs does not sufficiently show that a research problem exists. There may already be ISDMs that fulfill the method requirements, i.e., an existing solution to the identified ISD problems and needs. As shown in Figure 4, the knowledge base needs be consulted and assessed to determine if any extant ISDMs meet the method requirements (Karlsson & Ågerfalk, 2007). A structured literature review (e.g. Webster & Watson, 2002) would be a useful tool toward achieving this end.

If theoretical ideas are the starting point for a new ISDM, discussions, hypotheses, and/or theories in the existing knowledge base drive the method requirements and one or more of the research questions. Of course, it is equally important to justify such an ISDM from both a practical and a research perspective. Hence, in order to fulfill ME-DS Principles 1 and 2, it is necessary to identify the kind of ISD problems and needs that such an ISDM would resolve, in addition to showing the lack of such methods in the knowledge base. Together, the elicited method requirements and research questions become part of the method rationale, which is the output of the Identify ISDM Problem and Motivate activity in Figure 4.

The next two activities—ISDM Theorizing and Method Engineering—are highly intertwined, as shown in Figure 4. Although we present them in sequence below, it is important to note that they are often carried out in an iterative pattern in which they inform each other. Thus, it is possible to begin with either one of them. The iterative pattern is necessary in order to achieve a theoretical grounding (Goldkuhl, 2004) in which the method rationale and ISDM are mutually adjusted.

Activity 2: ISDM Theorizing. This activity implements ME-DS Principles 3, 4, and 5. As shown in Figure 4, it contributes to the method rationale of the developed ISDM. DS stresses the need to define the goals of the solution and the principal means for achieving this end (Peffers et al., 2007). Thus, it is necessary to infer what goals the new ISDM intends to achieve and how the new ISDM is expected to support solutions for the identified ISD problems that have not previously been addressed in a satisfactory way. To do this, we start with ME-DS Principle 3, which expresses that an ISDM is a normative artifact that should be based on a perspective with relevance for the type of

ISD problems needing to be solved (Brinkkemper, 1996; Jayaratna, 1994; Russo & Stolterman, 2000). This perspective could, for example, follow from the discussions, hypotheses, theories, and/or empirical insights that act as a base for the ISDM. The values proposed by this perspective can be used to define goals (Ågerfalk & Wistrand, 2003), expressing what the ISDM should accomplish in relation to the method requirements.

ME-DS Principles 4 and 5 address the principal means to achieve these goals. ME-DS Principle 4 states that the ISDM should include well-established concepts and conceptual structures from the knowledge base when available. Future developers will use these concepts to understand the ISD domain, typically by asking questions during systems development processes. ME-DS Principle 5 states that the knowledge base should be consulted regarding the kind of functional patterns that can be used by future developers to operationalize the concepts and the conceptual structure in a fruitful way. For example, if a theory drives the perspective, this theory could inform about relevant functional patterns. As shown in Figure 4, the knowledge base therefore provides input into ISDM Theorizing. If concepts, conceptual structures and/or functional patterns are not available in the knowledge base, such parts should be developed.

It is important to note that concepts, conceptual structures, and functional patterns should be the relevant principal means to accomplish the ISDM goals. The values promoted by the perspective are intended to help future developers focus on important aspects of the ISD domain through the use of relevant concepts, conceptual structures, and functional patterns. Hence, ME-DS Principle 3 guides the implementation of ME-DS Principles 4 and 5 and what are considered relevant principal means. Taken together, these arguments constitute input for the method rationale. The method rationale is important because it offers a general point of departure for devising ISDMs that is similar to the method suggested in the paper by the reporting researchers.

Activity 3: Method Engineering. This activity implements ME-DS Principle 7. Developing an artifact is central to DS (Hevner et al., 2004; Peffers et al., 2007). We focus here on the development of an ISDM, which is the output from the method engineering activity in Figure 4. The developed ISDM should be in line with the emerging method rationale, which acts as an input along with the previous version(s) of the ISDM and the knowledge base. The actual development can draw on a wide variety of ME strategies (see Henderson-Sellers & Ralyté, 2010; Henderson-Sellers et al., 2014; Ralyté, Deneckère, & Rolland, 2003) to achieve ME-DS Principle 7. According to both Henderson-Sellers et al. (2014) and Ralyté et al. (2003) there are two basic ME strategies: tailoring an existing ISDM and constructing a new ISDM. The first strategy targets situations where an ISDM exists that may be applicable but the method "requires some adaptations" (Ralyté et al., 2003). Specific strategies have been suggested for this purpose (e.g. Bajec et al., 2007; Karlsson & Ågerfalk, 2009). The second strategy targets situations in which there is no good fit between existing ISDMs and the situation, i.e., a new ISDM must be constructed, which can be done through a variety of strategies that can broadly be categorized as (1) instantiation of meta-models (Cervera, 2015; Tolvanen, 1998); (2) selection of existing method parts (e.g. Brinkkemper, 1996; Harmsen, 1997; Ralyté & Rolland, 2001); and (3) the development of method parts from scratch (Henderson-Sellers et al., 2014).

It is beyond the scope of this paper to provide an extensive account of how to choose and combine ME strategies. However, Ralyté et al. (2003) offer a detailed account of how to choose between ME strategies using goals, meaning that the combination of ME strategies should be based on the most effective way of reaching the goals that are set during different stages of method engineering. For example, if the goal is to target a new type of development situation that has previously not been supported by an ISDM, constructing a new ISDM would likely be the overarching ME strategy. The next step is to consider the most effective ME strategy for constructing a new ISDM. One fundamental idea of (situational) method engineering is reuse, which should act as a guiding principle when choosing ME strategies. As a new development situation is approached, a metamodel may not yet exist for instantiation. Depending on the requirements, it may be possible to use an assembly-based strategy (Ralyté & Rolland, 2001) that selects method parts from existing methods stored in a method base repository instead of developing method parts from scratch. If the selected method parts overlap with each other, the product and the process models should be integrated (Ralyté & Rolland, 2001). Irrespective of the chosen strategy, it is important that the ISDM is internally grounded (Goldkuhl, 2004) in order to fulfill ME-DS Principle 7, i.e., the ISDM should not be contradictory or have any loose ends.

The next two activities, ISDM Demonstration and Formal Evaluation, have an important dependency. As shown in Figure 4, the Formal Evaluation activity requires empirical results from ISDM Demonstration as input. It is therefore important to collect data during ISDM Demonstration that are relevant for the evaluation. Hence, Formal Evaluation should begin before ISDM Demonstration to develop an evaluation plan. This plan should guide what kind of demonstration is needed and which aspects of the ISDM should be focused on, which would also suggest that Formal Evaluation guides the data collection during ISDM Demonstration.

Activity 4: ISDM Demonstration. This activity implements one aspect of ME-DS Principle 6 to reveal the practical applicability of the ISDM. DS research (Peffers et al., 2007; Walls et al., 1992) has stressed the importance of proving that the suggested solution works. In our case, this involves demonstrating how the ISDM contributes to solving one or more of the problems that exist in the ISD practice. This could involve a wide variety of different types of demonstrations, such as expert assessment (Tremblay, Hevner, & Berndt, 2010), experiment (D'Aubeterre et al., 2008), case study (Ayed et al., 2010), or any other appropriate activity.

When choosing the type of demonstration and the aspects of the ISDM to demonstrate, the ISDM's maturity is important to consider. During early ME-DS iterations, the ISDM may be immature and expert assessment or experiment might be wise choices for demonstrating a selected aspect of the ISDM. For example, using an expert assessment suggests that knowledgeable person(s) in a particular field would reason about the ISDM's possibilities to solve the targeted ISD problems. When initial flaws are removed, i.e., in later ME-DS iterations, case study approaches might be more appropriate to demonstrate the full potential of the ISDM in an organizational setting. Based on how the demonstration is set up, the collected data—ISD Situational Results in Figure 4—will take different forms. Still, it is important that the collected data fit the purpose of the Formal Evaluation.

Activity 5: Formal Evaluation. This activity implements one aspect of ME-DS Principle 6 to evaluate the ISDM's usefulness. Hence, the activity is a formal and systematic determination of the ISDM's merits regarding the ISD problems, needs, and ideas that were initially located in the ISD practice and/or the theoretical ideas. As discussed above and illustrated in Figure 4, this activity is related to ISDM Demonstration. However, scholars (e.g. Peffers et al., 2007) have also acknowledged that demonstration is an activity in its own right. Formal evaluation implies comparing the actual results to a standard—for example, comparing the ISDM's performance to the method's requirements and/or goals. It is therefore important to develop a plan for how to collect data about the ISDM's performance during ISDM Demonstration.

Depending on the characteristics of the chosen standard(s), data collection and analysis could take many forms, including objective quantitative performance measures (Petersen, 2011) such as costs or number of met deadlines, interviews (Middleton, 1998), experiments (D'Aubeterre et al., 2008), or surveys (Ham et al., 2004). Theoretical frameworks that support the assessment of the chosen standard can also drive evaluations. For example, Karlsson and Hedström (2013) used design boundary object theory to assess end-user development as a requirements-engineering technique during ISD. As shown in Figure 4, the Formal Evaluation activity results in lessons learned that are included in the method rationale. In addition, this activity can result in change requests that serve as input for possible ISDM redesign during the next ME-DS iteration.

Activity 6: Communication. This activity implements ME-DS Principle 8. Communication concerns the dissemination of resulting knowledge to scholars and practitioners or other audiences when appropriate (Peffers et al., 2007). It is also an important activity in the ME-DS process for receiving feedback and criticism from peers on the work carried out. As illustrated in Figure 4, this activity therefore exists in parallel with the other ME-DS activities to enable continuous diffusion of the ISD problem and its importance, the current ISDM design, its method rationale, and its novelty and effectiveness. Communication needs to be tailored to the audience. Scholarly publications and presentations may cover different parts of the process presented in Figure 4, depending on how much work has already been done. Presentations and publications directed toward practitioners might preferably focus on the ISDM itself and on achieved results.

The research process detailed above contributes to different types of results: ISDM, method rationale, change requests, publications, and presentations. Method rationale may be the most complex of these results, as it includes all arguments underlying the ISDM (being justificatory knowledge). Hence, it should be stressed that method rationale is an aggregate of eight elements: method requirements, research questions, perspective, goals, concepts, conceptual structures, functional patterns, and lessons learned.

5 Demonstration

5.1 Design of Retrospective Demonstration

To demonstrate the use of ME-DS, we have applied it retrospectively to existing ISDM development papers. Retrospective demonstration is an established way of assessing proposed research approaches in DS. It is often used in connection with an initial presentation of a new research approach. Retrospective demonstration has been used in several prominent DS papers (e.g. Baskerville & Pries-Heje, 2010; Gregor & Hevner, 2013; Hevner et al., 2004; Kuechler & Vaishnavi, 2012; Peffers et al., 2007) as well as in related areas of IS (e.g. Gregor, 2006). Retrospective demonstration is defined as applying elements of a proposed research approach to one or several already published cases. We investigated the above-mentioned publications for purposes and selection, as well as presentation principles, and summarize the detailed results of this investigation in Appendix C, Table C1.

In Table C1, the above-mentioned authors give different purposes for conducting a retrospective demonstration. One purpose is to *illustrate and illuminate*, i.e., to show templates for possible users. We also found obvious applicability tests, in which a knowledge framework is applied to cases to try out its conceptualization. This corresponds to the purpose of the *proof-of-concept test* introduced by Gregor and Hevner (2013) with reference to Nunamaker and Briggs (2011). *Evaluation* has also been cited as a purpose (Peffers et al., 2007), although there is some hesitation regarding this ambition (Hevner et al., 2004). For example, in the meeting of a proposed knowledge framework and a selected case, what, if anything, is evaluated? We argue that evaluation can work in both directions when two knowledge artifacts are confronted. A proposed knowledge framework (type level) is applied to a selected case (instance level). This application implies a case description but it may also imply an assessment of the case following concepts and possible ideals in the knowledge framework. However, such an application can also serve as an evaluation of the knowledge framework to gauge how useful it is for description, reasoning, and assessment (a proof-of-usefulness for evaluation).

We use retrospective demonstration for all three purposes. First, our demonstration serves the purpose of *illustrating* our ME-DS proposal, i.e., to clarify and exemplify possible uses. Second, it also serves a *proof-of-concept* purpose, i.e., to see if it is possible and meaningful to apply the ME-DS concepts and principles to relevant cases. Third, the use of these concepts and principles also includes some *evaluation* of each case. This is not an activity undertaken for the primary sake of assessing these published cases; rather, this activity mainly serves the purpose of testing if it is possible to use the concepts and ideals of ME-DS for describing, reasoning, and assessing a case; i.e., a *proof-of-usefulness for evaluation*.

The cases we study are *published research studies* on ISDM development. Publications are the standard way of packaging scholarly knowledge and these are the natural study objects for an inquiry of conducted research. Using published research offers other scholars the opportunity to validate and replicate the analysis. The number of studied cases varies among the above-mentioned papers from two cases to thirteen cases (see Table C1 in Appendix C). Consequently, there are differences in the depth of the analyses and presentations of these cases and they range from detailed assessments (several pages) to short notes in table cells.

Based on the above, we have chosen to use three papers from our literature review described in Section 2.3 (detailed in Appendix A). This choice was driven by the need for adequate depth in analysis and presentation to provide a basis for appropriate applicability and assessment tests and the desire to nevertheless offer some variation among cases. Our selection process prioritized studies that employ research approaches closely related to ME-DS. In concrete terms, it meant focusing on studies that used research approaches from ME and DS, since it is from such studies that ME-DS borrows the most ideas. This selection strategy enabled a close assessment of how ME-DS can contribute to each of the selected studies. We selected the following three papers because our literature review revealed that they represent good and clear cases of ME, DS, and the combination of ME and DS:

- Rahimian and Ramsin (2008) developed an agile ISDM for the area of mobile software development using ME.
- van de Weerd et al. (2006) developed an ISDM for web applications using a combination of ME and DS.
- D'Aubeterre et al. (2008) incorporated security as functional requirements in an existing ISDM using DS.

5.2 An Agile Methodology for Mobile Software Development: Rahimian and Ramsin

Rahimian and Ramsin (2008) propose an agile ISDM for the area of mobile ISD. At the time of their study, there was rapid growth of wireless networks that transformed mobile phones into important platforms for IS. The authors argue that mobile systems have unique requirements and constraints that affect ISD. The ISDM drew upon ideas of risk-based, architecture-centric, and test-based development. They also incorporated ideas from the domains of adaptive software development and new product development.

5.2.1 Principle 1: Practical Relevance

Rahimian and Ramsin offer a general impression of practical needs. They did not collect any specific data from industry to justify the practical needs of a new ISDM. Instead, they based their argumentation on a set of challenges identified in existing research. However, the authors could have strengthened their practical relevance arguments by providing more up-to-date references.

The authors propose seven traits, or method requirements that they "believe a development method should have in order to be efficiently employed for mobile software development" (Rahimian & Ramsin, 2008, p. 339). However, the authors could have been more transparent in their elicitation of the method requirements; they did not explicitly show how these traits were anchored in practical challenges or in the existing knowledge base.

5.2.2 Principle 2: Research Relevance

Rahimian and Ramsin argue that scholars have made few attempts to devise ISDMs for mobile ISD. They claimed that work performed has mainly focused on implementation-oriented aspects of ISD while method-oriented aspects have not been "properly addressed" (Rahimian & Ramsin, 2008, p. 337). The authors discuss method-oriented research on ISDM that targets mobile ISD. However, their review is presented as a general discussion and is difficult to decipher the extent to which existing ISDMs fulfilled any of the method requirements. The authors could have provided a more systematic analysis of existing research to show research relevance.

5.2.3 Principle 3: Value Compliance

Rahimian and Ramsin base their ISDM on a number of different starting points. They propose agile development as the most central aspect, claiming that "agile characteristics" are "of [the] highest importance in the context of mobile software development" (Rahimian & Ramsin, 2008, p. 339). They also base their ISDM on a generic waterfall-based ISD life cycle, software product lines, and architecture-based and risk-based development. These starting points are presented as part of the method requirements, revealing an intertwined elicitation of requirements and design. The authors did not explore whether agile development, the generic ISD life cycle, software product lines, and architecture-based development contain conflicting values nor did they describe any prioritization of values. Instead, these starting points were treated as black boxes and it remains unclear which values actually motivated the design decisions.

5.2.4 Principle 4: Conceptual Concordance

Rahimian and Ramsin base their ISDM on a generic ISD life cycle, which is modified according to agile development, software product lines development, and architecture-based development. For example, they borrow high-level concepts, such as the phases of "idea generation" and "market testing" from the domain of new product development. Still, it is difficult to assess how they used more detailed concepts and conceptual frameworks from the knowledge base. The authors could have provided more details about the ISDM's conceptual content and its origin.

5.2.5 Principle 5: Functional Pattern Concordance

The authors base their ISDM on a number of existing high-level functional patterns and clearly detail the patterns that have been used for specific parts of the ISDM design. Their starting point is a waterfall-based functional pattern based on the generic ISD life cycle. This ISD life cycle was extended with a functional pattern to create prototypes during the design phase in order to minimize risk when specifying architectures. In this case, the authors draw upon software product lines and architecture-based development. Finally, the authors implement an iterative functional pattern regarding the implementation and test phases, borrowing from the domain of adaptive software development.

5.2.6 Principle 6: Practical Applicability and Usefulness

This paper does not present any ISDM demonstration or formal evaluation, meaning that ME-DS Principle 6 was not implemented in this study and the ISDM's practical applicability and usefulness remain unproven. Thus, at this stage, the artifact is a theoretical product. The authors discuss demonstration of the ISDM as future research, which would implicitly seem to include evaluation.

5.2.7 Principle 7: Intelligible and Harmonious Whole

Rahimian and Ramsin used hybrid methodology design, an ME metamethod, to systematically devise the ISDM as constituting an intelligible and harmonious whole. The design work was carried out over three iterations and included a couple of different ME strategies. First, the ME started with a choice and instantiation of a metamodel, the generic ISD life cycle. In later iterations, the authors used available functional patterns and integrated them into the new ISDM. As this study does not include any ISD demonstration and formal evaluation, it is not exactly clear what motivated a division of the development work into iterations.

5.2.8 Principle 8: Target Audiences

This work was presented at a scientific conference and mainly targets a scholarly audience. The paper focuses on the traits driving the ME activity. Furthermore, the paper is aimed at an audience familiar with ME and the ISDMs from which the authors borrowed concepts and functional patterns. Systems developers and project managers are provided with a high-level view of the proposed ISDM. While the authors offer references to the metamodels and ISDMs on which they build, their ISDM would be more accessible to systems developers if it included more details about the method's content. This is something the authors addressed as a potential for future research. In addition, the fact that the ISDM's practical applicability and usefulness have not been assessed lessens the value for practitioners.

5.3 A Method for Web-Based Content Management System Applications: Van de Weerd et al.

In the early years of the Internet, websites were developed by programmers who coded each web page. As websites grew into web applications, this became an inefficient way of working. The solution was found in content management systems (CMS), which provide an environment for creating, controlling, and publishing web content. Van de Weerd et al. (2006) devised an ISDM—GX WebEngineering Method—for implementing web-based CMS applications. The research project was carried out as a DS project together with an assembly-based ME strategy. They reused method parts of the following ISDMs: Unified Process (UP), UML-based Web Engineering (UWE), and GX development process (an in-house ISDM at the organization where the research was carried out). The final ISDM contains three possible routes for using the method, one for standard projects, one for complex projects, and one for migration projects.

5.3.1 Principle 1: Practical Relevance

Van de Weerd et al. (2006, p. 525) conclude that there is "a need for methods for implementing web-based CMS applications." They base this conclusion on a general argumentation that existing ISDMs failed to target web application development and that this kind of development differed from traditional ISD. According to the authors, traditional ISD starts "from scratch or from an existing legacy system" (van de Weerd et al., 2006, p. 522). When developing a web application, a situational ISDM is needed because system developers are building a system based on a preconfigured product. The authors stress that ISDMs available for this purpose were vendor specific and not freely available. They could have strengthened their argument by providing empirical data from industry to justify that such methods are not available.

5.3.2 Principle 2: Research Relevance

Van de Weerd et al. (2006) argue that there were no suitable ISDMs in the knowledge base. They reviewed a number of existing ISDMs, focusing on methods specifically designed for web applications. The review is presented as a general discussion describing the methods' intentions and the aspects of web application they cover. The authors could have made the analysis more concrete by assessing the extent to which the reviewed ISDMs met the method requirements they elicited.

5.3.3 Principle 3: Value Compliance

The authors combined UP and UWE with the GX development process. The consulting organization was familiar with the UP and UML and the new ISDM needed to deliver results that were "understandable to the customer and informative for the stakeholders at GX" (van de Weerd et al., 2006, p. 528). UWE is UML-based and "combines the strengths of the Unified Process with several web-specific characteristics" (van de Weerd et al., 2006, p. 528). Hence, the familiarity value was important.

Because it borrowed method parts from UP and UWE, the new ISDM included subsets of the value bases that shaped these methods. One such example is the decision to include use-case modeling in the complex route but not in the standard route of the method; in the latter case, it was viewed as too time-consuming. However, the values driving the design choices are to a large extent implicit. A more explicit articulation of this part of the method rationale would have been helpful for other scholars and practitioners designing similar ISDMs.

5.3.4 Principle 4: Conceptual Concordance

Van de Weerd et al. (2006) explicitly defined the concepts and conceptual framework they borrowed from UP, UWE, and GX development process. For example, they reused the concepts of actor, use case, and use case models that are found both in UP and UWE and presented the conceptual structures using metamodeling. The authors did not provide definitions of these concepts because they assumed reader knowledge about these ISDMs.

They also introduced a few new concepts, some of which are related to the different routes that exist in the ISDM: standard, complex, and migration. These concepts were clearly defined in relation to the scope of the ISDM presented in the paper that focuses on the definition phase of the web application.

5.3.5 Principle 5: Functional Pattern Concordance

The authors provide a detailed view of the activities in the definition phase, clearly articulating the original ISDM in which a specific functional pattern originated. "A checked pattern indicates that this method fragment originated from the old method at GX; grey indicates that it is a UWE fragment; and, finally white indicates a Unified Process origin" (van de Weerd et al., 2006, p. 529). For example, in the ISDM's complex route, the authors borrow the functional pattern for domain modeling from UWE, use case modeling from UP, and application modeling from the GX development process.

5.3.6 Principle 6: Practical Applicability and Usefulness

The authors prove the practical applicability and usefulness of the ISDM's definition phase through an expert assessment and two case studies, exemplifying how the complexity of an ISDM demonstration can be incremented gradually. Before demonstrating the definition phase in real projects, it was reviewed by experts—in this case, consultants and project managers at the organization where the research took place. The case studies aimed at evaluating the method's standard and complex routes in real settings.

The authors did not describe how the data were collected during the expert assessment nor did they describe the standards used during evaluation. During both case studies, data were collected using questionnaires and interviews. However, few empirical details are presented. The authors could have strengthened their arguments by using quotes from the interviews instead of short summaries. Beyond the general description of why the experts were positively inclined toward the new ISDM, the authors could have used their rich case study data to supplement the method rationale with a more detailed explication of lessons learned, which would have increased the value of this study for future method users.

5.3.7 Principle 7: Intelligible and Harmonious Whole

Van de Weerd et al. (2006) employed an assembly-based ME strategy. They divided the original ISDMs into method fragments that were stored in a method base and later reused. They used process-data diagrams to trace how process and product fragments are connected to create a harmonious whole. As discussed above, the authors provide few explicit values for their method design. Hence, the ME activity focused more on the choices made rather than why these choices were made.

5.3.8 Principle 8: Target Audiences

Van de Weerd et al. (2006) address a limited portion of the proposed ISDM—the definition phase. However, they provide an extensive overview of this phase, including how they arrived at this design. Hence, while the paper contains guidance for systems developers, systems developers will still need to consult the original ISDMs to get more detailed instructions about the activities and deliverables.

The authors also have scholars as a target audience, and thus provide an extensive description of how they devised and assessed the ISDM. Their work would have been even more valuable had they provided a stronger motivation concerning the ISDM's practical relevance and had they used the method requirements to analyze existing ISDMs. In addition, the design would have been more accessible to this target audience had they been more explicit about the values driving the selection of method parts. This would have provided a more generic description of the method design that could be useful for designing other ISDMs for web-based CMS applications.

5.4 Incorporating Security as a Functional Requirement in the Analysis and Modeling of Business Processes: D'Aubeterre et al.

In ISD, security requirements are often treated as an afterthought in the nonfunctional requirements of IS and thus separated from the analysis of functional requirements. D'Aubeterre et al. (2008) propose an ISDM addition called secure activity resource coordination (SARC), which integrates security considerations into business process modeling and is based on their prior work elaborating a semantic approach to secure collaborative interorganizational e-business processes. They explicitly refer to their use of a DS approach following Hevner et al. (2004).

5.4.1 Principle 1: Practical Relevance

The authors motivate their study by claiming that there is a risk of IT vulnerability if security requirements are not well integrated with functional IT requirements. While they seem to base this on a general impression of needs, they give no specific report of data collection from practice concerning such needs, which would have strengthened the authors' arguments.

5.4.2 Principle 2: Research Relevance

The authors refer to discussions in the research literature calling for further method development and the addition of security requirements as an integral part of the functional requirements of an IS. While there are published accounts of this kind of addition and integration, D'Aubeterre et al. (2008) conclude that there is a knowledge gap in the specific area of business process development: "the lack of appropriate security controls on information exchanged among business activities in a business process can leave organizations vulnerable to information assurance threats" (D'Aubeterre et al., 2008, p 529). The authors' arguments would have benefitted from a clarification of how general ISD differs from business process-oriented ISD, especially in terms of security requirements.

5.4.3 Principle 3: Value Compliance

D'Aubeterre et al. (2008) rely on a set of values, including the sharing of knowledge and information across interorganizational business processes and the sustainability of security of such information resources. This implies a controlled balancing of these potentially conflicting values. The developed ISDM addition, SARC, should be interpreted as a proposal for how to operationally balance these values in the design of a business process-oriented IT artifact.

5.4.4 Principle 4: Conceptual Concordance

SARC applies security concepts based on role-based access control. Concerning business process management, concepts are obtained from coordination theory. The authors made certain conceptual simplifications regarding dependencies between business activities. SARC integrates security concepts and business process coordination concepts. From an ME perspective, a more structured description and analysis of used and integrated concepts would have been desirable.

The rationale for D'Aubeterre et al.'s work is a demarcated focus on IS for business processes. This focus on one subpart of IS is based on recurrent discussions of "eBusiness processes in an extended enterprise" (D'Aubeterre et al., 2008, p. 530). However, their presentation of and motivation for SARC would have benefitted from a deeper conceptual analysis and clarification of this central concept.

5.4.5 Principle 5: Functional Pattern Concordance

The development of SARC is based on a general hypothesis on the benefits of the integration of security requirements within specific functional requirements. It is clear that the presented method is based on this hypothesized functional pattern.

One claimed benefit of a SARC application may be greater security awareness among IS analysts. This notion of security awareness derives from situational awareness theory. This theory functions as a kind of background for motivating the ISDM and the demonstration of its possible benefits, as described in the next section.

5.4.6 Principle 6: Practical Applicability and Usefulness

Applicability and usefulness have mainly been demonstrated in an artificial setting. The authors used case material from an application in a real organization. The original case material was simplified and models were created using SARC and a rival modeling notation (enriched-use cases). An experimental test situation was developed using enrolled IS students and comparative data were collected concerning students' understanding of "elements of security in the business models presented to them including authorization constraints, security policies, and security violations" (D'Aubeterre et al., 2008, p 535). D'Aubeterre et al. (2008) summarize the outcome of this experimental test: "Our results show that SARC artifacts help analysts develop greater security awareness in requirement specifications, modeling, and analysis of business processes" (p. 529). One could object that this is an exaggerated claim, based on the empirical material, since the test only measured students' knowledge based on interpretations of the SARC models at hand. They did not study the process of development of requirement specifications nor did they investigate real analysts applying this ISDM. In fact, the authors did not discuss the need for real-life demonstration of the method at all, save one suggestion that future research should advance the statistical testing in artificial settings.

5.4.7 Principle 7: Intelligible and Harmonious Whole

The basic idea of this paper seems to center around an explication of "modeling concepts and grammar for SARC secure business processes" (D'Aubeterre et al., 2008, p. 531). While the authors discuss concepts and present examples of SARC models, they include no ME-based description of the developed ISDM. A more structured and comprehensive description and analysis of SARC would be desirable, both from a scholarly view and to guide its practical use.

The method presented in this paper is an integration of concepts and views from two areas: security management and business process management. The integration of concepts from these two areas is described in the paper but, from an ME-perspective, a more thorough conceptual discussion would be desirable.

5.4.8 Principle 8: Target Audiences

D'Aubeterre et al.'s (2008) paper was published in the *European Journal of Information Systems* and has a clear orientation toward research rigor—for example, an emphasis on experimental design regarding ISDM demonstration and formal evaluation—and thus clearly targets a scholarly audience. While examples of models are included in the paper, they are not accompanied by any clear descriptions of notational rules that could guide method users (practitioners) in employing the method.

5.5 Lessons Learned

We summarize experiences and conclusions derived from demonstrating ME-DS using the three papers discussed in detail above. These cases were influenced by the research approaches (ME, DS) employed, which would naturally exclude influence by ME-DS. However, our analysis of the published research cases follows the structure of the eight ME-DS principles we introduce. We interpreted the cases and reasoned about them based on ME-DS concepts and how the cases complied with the normative ideals of ME-DS principles. For our analysis, we divided the paper reviews between us. After writing the separate reviews, we compared them and found similar reasoning across cases and authors. We made minor adjustments to reach consensus regarding how assessments were made across cases. For most principles, we found the analysis of these cases to be fairly straightforward. In some situations, it was more challenging and required in-depth reading and interpretation, which was likely based on a lack of explicit attention to certain ME-DS aspects within these papers.

We were able to identify shortcomings and suggest possible improvements in each assessed case following the ideals of ME-DS. For example, the analysis of ME-DS Principle 2 revealed a shortcoming in terms of the motivation of research relevance; the analyzed papers provide only general discussions of existing research to justify the need for a new ISDM, resulting in often shallow arguments. Another example is related to the analysis of ME-DS Principle 3. Our analysis revealed that the authors of these papers did not sufficiently define the values driving their design choices (i.e., they were black-boxed), which is important because ISDMs are normative artifacts. Thus, based on this retrospective demonstration, we assert that ME-DS, has been applicable and useful for case evaluation and for proposing potential enhancements to these studied cases. We have reached a stage of knowledge development regarding ME-DS, with insight into the structure (i.e., different and complementary principles), content, and possible uses of ME-DS, that could be communicated to a larger audience.

6 Discussion and Conclusions

In this paper, we propose a marriage between method engineering (ME) and design science (DS), resulting in the research approach “method engineering as design science” (ME-DS). ME-DS has emerged from the identified need for scholars to develop ISDMs using proper research methods that meet the standards of both rigor and relevance, which grounds the practical relevance of ME-DS. ME-DS also answers Bucher and Winter’s (2008) call for the integration of ME and DS. Furthermore, our theorizing of the ISDM artifact shows that this type of artifact differs from the IT artifact, a point that has been largely ignored in existing DS approaches, which demonstrates the research relevance of our work. Our method theorizing and engineering of ME-DS resulted in eight principles that guide the integration of ME and DS. Through assessing three previously published research papers on ISDM development, we retrospectively demonstrate the usefulness and applicability of ME-DS. Our evaluation, integrated with the demonstrations, illustrates what ME-DS adds to research-based ISDM development. As shown by the discussion above, the development of ME-DS is an inquiry in the pragmatist sense (Dewey, 1938) following a design science pattern.

Synthesizing ME and DS is not a straightforward process, as both disciplines contain many nuanced differences. There are several different ME strategies that emphasize different aspects of the ISDM concept. For example, concerning the difference between how method fragments (Harmsen, 1997) and method components (Karlsson & Wistrand, 2006) address goals and values, the latter might make it easier to implement the ME-DS principle of value compliance, i.e., the perspective and its values should drive the ME activity in ME-DS. Our strategy in designing ME-DS treats it as an open structure, making it possible to choose between different ME strategies. We also conclude that there are different varieties of DS that, for example, vary in terms of how much the theoretical aspect of the proposed artifact is emphasized (cf. Kuechler & Vaishnavi, 2008). In ME-DS we have chosen to emphasize this aspect of ISDM design—method theorizing—because it is important for developing method rationale, which is central to ISDM development (Brinkkemper, 1996; Jayaratna et al., 1999; Russo & Stolterman, 2000). Thus, we do not claim that the suggested process model is the only possible way to implement ME-DS principles and structure ME-DS.

Many of the presented ideas about ME-DS are not new and have been drawn from a considerable literature on ME and DS, i.e., they are well grounded. The proposed process model uses method parts (process and product elements) from existing ME and DS approaches. As such, the ME-DS process model and activities are the result of a method engineering activity paralleling the theoretical work. Nevertheless, the selection and integration of these ideas have led to a research approach that differs from existing ME and DS approaches. We argue that both ME and DS have strengths and weaknesses regarding the research-based development of ISDM. The contributions from our research should therefore be assessed in terms of the existing shortcomings in ME and DS, respectively, regarding such knowledge development.

6.1 Contributions to the Method Engineering Discipline

This paper contributes to existing ME metamethods (e.g. Bajec et al., 2007; Cameron, 2002; Harmsen, 1997; Karlsson & Ågerfalk, 2009; Ralyté & Rolland, 2001; Sandkuhl & Seigerroth, 2019) by addressing the relevance and the usefulness of the proposed ISDM and reducing the black-boxing of development work. We show that much of the published high-impact research on ISDM development comes up short regarding research methods, specifically in terms of how the proposed ISDM is generated and empirically validated. It is striking that research papers in this area often seem to lack transparent accounts for why new ISDMs are needed or for the applicability and usefulness of the proposed ISDMs. Therefore, the ME-DS principles of practical relevance, research relevance, and practical applicability and usefulness contribute significantly to ME. ME-DS provides scholars with an explicit structure that stresses the need for anchoring a new or modified ISDM both in practice *and* in research. This structure also includes steps for validating the proposed ISDM through combining ISDM demonstration and formal evaluation. Since ME tends to focus on demonstration, the relationship between these two activities has not been addressed sufficiently in existing ME-approaches. ME-DS highlights that scholars need to sort out how these two activities should interact in their specific research projects.

Furthermore, the ME-DS principles of value compliance, conceptual concordance, functional pattern concordance, and an intelligible and harmonious whole are important for reducing black-boxing. Since the ideas behind these principles can be found in ME (Henderson-Sellers & Ralyté, 2010; Henderson-Sellers et al., 2014), the principles themselves are therefore not a contribution. However, we specifically draw on how DS separates defining requirements and objectives from constructing the artifact, which led us to operationalize these principles as two distinct activities (ISDM theorizing and method engineering). ME-DS contributes to ME by showing that scholars need to combine theoretical work with existing ME strategies, such as construction-based strategies (Henderson-Sellers et al., 2014; Ralyté & Rolland, 2001), when carrying out research-based ISDM development.

Finally, the ME-DS principle of target audience represents a (minor) contribution to ME. Although communicating resulting knowledge is a focus of most research, it has not been a traditionally emphasized aspect of knowledge development in ME. It should be noted that we view communication as an activity conducted in parallel with the other ME-DS activities. Thus, this differs from how communication is treated by, for example, Peffers et al. (2007), who featured it as the last activity of each design cycle. We believe that treating communication as a continuous activity better supports the need for a cumulative approach to knowledge development discussed above. In addition, it aids somewhat in mitigating the problem of limited space associated with presenting research results. ME-DS provides scholars with a structure to refer to when publishing studies focusing on selected parts of the research process or when addressing a selected number of ME-DS principles in a research paper. Thus, ME-DS can help scholars clearly express how their research paper is situated within the overall research strategy.

6.2 Contributions to the Design Science Discipline

We contribute to DS by pinpointing the specific characteristics of ISDMs and how they impact knowledge developed using DS. We respond to Peffers et al.'s (2007, p. 74) call for "a methodology to support the design of methods for requirements analysis," as such a method for requirements analysis is a subtype of an ISDM. By theorizing ISDMs, we show that ISDMs are knowledge artifacts that are documented as sign artifacts, which differ from the IT artifact that often exists the foreground of extant DS approaches (e.g. Hevner et al., 2004; Kuechler & Vaishnavi, 2012; Peffers et al., 2007; Sein et al., 2011). These approaches build on the artifact typology of March and Smith (1995), which does not present a useful characterization of the method artifact. In this paper, we build on ME and existing artifact theorists (e.g. Beckman, 2002; Dipert, 1995), i.e., on the existing knowledge base, to make our contribution. We acknowledge that the specific characteristics of ISDMs have implications for (1) the execution of defining requirements and objectives for the proposed ISDM, and (2) carrying out the development work. Thus, the ME-DS principles of value compliance, conceptual concordance, functional pattern concordance, and an intelligible and harmonious whole represent contributions to DS. They result in a process model that stresses continuous iteration between ISDM theorizing and method engineering. In addition, by theorizing ISDM, we provide theoretical anchoring of the empirically and extended DS artifact typology proposed by Offermann, Blom, Schönherr, et al. (2010).

Significantly, providing an ISDM without any proper arguments does not qualify as ME-DS. For an ISDM to constitute scholarly work, it must include a coherent method rationale. The ME-DS principles of value compliance, conceptual concordance, functional pattern concordance, and practical applicability and usefulness provide a secure basis for providing such a method rationale. The ME-DS process model and activities offer scholars a supporting structure to develop method rationale and can help scholars pursuing the theoretical, internal, and empirical grounding of ISDMs. Thus, our intention is similar to Peffers et al. (2007), in that it provides scholars with a research approach and a mental model for presenting DS research. Where Peffers et al. (2007, p. 73), like many other DS approaches, focus on "one general methodological guideline for effective DS research," we focus specifically on scholars pursuing ISDM development. Hence, our work is a natural complement to existing DS frameworks.

Existing DS approaches (Hevner et al., 2004; B. Kuechler & Vaishnavi, 2008; Peffers et al., 2007), that place the IT artifact in the foreground, stress the necessity to assess the extent to which the new artifact is superior to existing artifacts. This often involves demonstrating utility by comparing artifact performance. However, demonstrating the utility of knowledge artifacts such as ISDMs often invokes a specific challenge. Rigorous comparison of ISDMs is often difficult to achieve in organizational settings because all ISD projects are unique. It is not possible to set up the same project with the same ISD team twice because teams learn from previous experiences. We therefore concur with Offermann, Blom, Bud, et al. (2010, p. 301) in their conclusion that the performance of ISDMs "cannot be easily be measured." It is also not possible to compare how two different teams perform on the same project task because each team brings different background knowledge into the project, which that affects how an ISDM is adopted (Fitzgerald, Russo, & Stolterman, 2002). This makes results from, for example, case studies difficult to compare. Experimentation, as used in D'Aubeterre et al. (2008), might be an option for comparisons between groups, making it possible to control for different levels of background knowledge. The drawback of such methods is that they tend to be based on artificial settings. This reveals the complexity of demonstrating and evaluating ISDMs. Thus, ISDMs necessitate a cumulative approach to knowledge development in which different types of demonstration are employed and provide pieces of the jigsaw puzzle.

6.3 Contributions to Practitioners

The primary audience for ME-DS is scholars; however, it may also be a useful method for practitioners developing or tailoring ISDMs in organizations. ME-DS provides practitioners with a vehicle to conduct evolutionary ME (Rossi et al., 2004) in which each project would provide natural boundaries for a design iteration. As ME-DS focuses on the continuous evaluation of the ISDM and the need to provide a complementary method rationale, it could help organizations continuously improve their ISDMs. The developed method rationale would provide practitioners with a track record regarding different solutions to methodological problems and how they have worked out thus far; ME-DS would also provide support for building on the existing knowledge base. Nevertheless, practitioners, or scholars for that matter, should not apply the ME-DS process model and its activities using a rigid orthodoxy. Instead, it is important to acknowledge that the situational and local character of knowing means that the activities need to be adapted to the current situation, often referred to as method-in-action (Fitzgerald et al., 2002). In terms of ME-DS, we believe that the important aspect is acknowledging the importance of the proposed eight principles.

6.4 Limitations

Alongside our retrospective demonstration, we evaluated ME-DS by example in this paper. However, because the three studies we assessed were not conducted using ME-DS, they should not be viewed as exemplars of its application. They nevertheless contribute to knowledge because our retrospectives analysis of these papers illustrates some of the shortcomings in existing research on ISDM development. While our analysis shows how ME-DS can direct attention to certain aspects of the research process and can therefore support future ISDM development, the reader should be aware of the limitation of retrospective demonstration based on existing papers. The validity of such a demonstration relies on how well the described research processes correspond to the actual activities in the research projects. Of course, any research publication will confront difficulties in presenting complex research processes within limited space constraints, meaning that there may be deviations between the communication and the actual execution of the research project.

6.5 Future Research

Our study opens up avenues for future research. ME-DS needs to be demonstrated and evaluated in actual research-based ISDM development going beyond the retrospective demonstration and evaluation carried out in this paper. Such work would add important lessons learned to ME-DS and its method rationale. As discussed above, ME-DS boasts a rather open structure, which would enable researchers to select both different ME strategies and ways to model and present ISDMs and their method rationale. This open structure may be viewed as a kernel, a common ground for research-based ISDM development, making it possible to contribute by offering more detailed guidelines on how to operationalize ME-DS activities in relation to specific ME strategies.

Furthermore, ISDMs represent one type of knowledge artifact. Other types of methods exist in other areas of the information systems field, such as in information security management (e.g. Beckers et al., 2013; Kolkowska, Karlsson, & Hedström, 2017) and IT governance (e.g. Bin-Abbas & Bakry, 2014; Simonsson, Johnson, & Wijkström, 2007). The concept of knowledge artifact is applicable to several of the categories outlined in March and Smith (1995) as well as Offermann, Blom, Schönherr, et al. (2010). For example, models, patterns, and guidelines share many of the artifact characteristics of a knowledge artifact. ME-DS can likely offer guidance in the development of other types of knowledge artifacts as well. Nevertheless, the reader should be aware that ME-DS is based on the extensive ME research on the modeling of ISDM. While other types of knowledge artifacts may have other unique characteristics, this opens up the possibility for exploring the applicability of ME-DS for the development of other types of knowledge artifacts.

References

Ågerfalk, P. J. (2013). Embracing diversity through mixed methods research. *European Journal of Information Systems, 22*, 251-256.

Ågerfalk, P. J., & Wistrand, K. (2003). Systems development method rationale: A conceptual framework for analysis. *Proceedings of the 5th International Conference on Enterprise Information Systems.*

Agh, H., & Ramsin, R. (2016). A pattern-based model-driven approach for situational method engineering. *Information and Software Technology, 78*, 95-120.

Ahituv, N., & Neumann, S. (2002). A system development methodology for ERP systems. *Journal of Computer Information Systems, 42(3),* 56-66.

Archer, L. B. (1984). Systematic method for designers. In N. Cross (Ed.), *Developments in design methodology* (pp. 57-82). Wiley.

Ayed, M. B., Ltifi, H., Kolski, C., & Alimi, A. M. (2010). A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain. *Decision Support Systems, 50*(1), 64-78.

Bajec, M., Vavpotič, D., & Krisper, M. (2007). Practice-driven approach for creating project-specific software development methods. *Information and Software Technology, 49*(4), 345-365.

Balmelli, L., Brown, D., Cantor, M., & Mott, M. (2006). Model-driven systems development. *IBM Systems Journal, 45*(3), 569-585.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. *Journal of the Association for Information Systems, 19*(5), 358-376.

Baskerville, R., & Pries-Heje, J. (2010). Explanatory design theory. *Business & Information Systems Engineering, 2*(5), 271-282.

Bayuk, J. L., & Horowitz, B. M. (2011). An architectural systems engineering methodology for addressing cyber security. *Systems Engineering, 14*(3), 294-304.

Beckers, K., Côté, I., Faßbender, S., Heisel, M., & Hofbauer, S. (2013). A pattern-based method for establishing a cloud-specific information security management system: Establishing information security management systems for clouds considering security, privacy, and legal compliance. *Requirements Engineering, 18*(4), 343-395.

Beckman, S. (2002). The nature of artifacts. In B. Dahlbom, S. Beckman, & G. B. Nilsson (Eds.), *Artifacts and artificial science*. Almqvist & Wicksell.

Ben Ayed, M., Ltifi, H., Kolski, C., & Alimi, A. M. (2010). A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain. *Decision Support Systems, 50*(1), 64-78.

Bergstra, J., Jonkers, H., & Obbink, J. (1985). A software development model for method engineering. In J. Roukens & J. Renuart (Eds.), *ESPRIT '84: status report of ongoing work*. North-Holland.

Bernard, J. (2009). Ten theses on perception in terms of work: A Rossi-Landian/Wittgensteinian point of view. *Semiotica, 173*, 155-167.

Beynon-Davies, P., & Holmes, S. (1998). Integrating rapid application development and participatory design. *IEEE Proceedings: Software, 145*(4), 105-112.

Biffl, S., Moser, T., Mordinyi, R., & Wahyudin, D. (2008). *Ontology-supported quality assurance for component-based systems configuration*. Paper presented at the 6th International Workshop on Software Quality.

Bin-Abbas, H., & Bakry, S. H. (2014). Assessment of IT governance in organizations: A simple integrated approach. *Computers in Human Behavior, 32*, 261-267.

Binford, L. R. (1962). Archaeology as anthropology. *American Antiquity, 28*(2), 217-225.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *IEEE Computer, 21*(5), 61–72.

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology, 38*, 275-280.

Bucher, T., & Winter, R. (2008). *Dissemination and importance of the "method" artifact in the context of design research for information systems*. Paper presented at the 3rd International Conference on Design Science Research in Information Systems and Technology.

Cameron, J. (2002). Configurable development Processes. *Communications of the ACM, 45*(3), 72-77.

Carter, C. J., Whyte, I. G., Birchall, S. A., & Swatman, D. (1997). Rapid integration and testing of business solutions. *BT Technology Journal*, *15*(3), 37-47.

Cervera, M. (2015). *A model-driven approach for the design, implementation, and execution of software development methods* (Doctoral dissertation). Universitat Politecnica de Vallencia, Spain.

Cervera, M., Albert, M., Torres, V., & Pelechano, V. (2012). A model-driven approach for the design and implementation of software development methods. *International Journal of Information System Modeling and Design*, *3*(4), 86-103.

Cervera, M., Albert, M., Torres, V., & Pelechano, V. (2015). On the usefulness and ease of use of a model-driven Method Engineering approach. *Information Systems, 50*, 36-50.

Champion, D., & Stowell, F. (2002). Navigating the Gap Between Action and a Serving Information System. *Information Systems Frontiers, 4*(3), 273–284.

Checkland, P. B. (1981). *Systems thinking, systems practice*. Wiley.

Cossentino, M., & Seidita, V. (2005). Composition of a new process to meet agile needs using method engineering. In R. Choren, A. Garcia, C. Lucena, & A. Romanovsky (Eds.), *Software engineering for multi-agent systems III: Research issues and practical applications* (pp. 36-51). Springer.

Crilly, N. (2010). The roles that artefacts play: Technical, social and aesthetic functions. *Design Studies, 31*(4), 311-344.

D'Aubeterre, F., Singh, R., & Iyer, L. (2008). Secure activity resource coordination: Empirical evidence of enhanced security awareness in designing secure business processes. *European Journal of Information Systems, 17*(5), 528-542.

Dahlbom, B. (2002). The idea of an artificial science. In B. Dahlbom, S. Beckman, & G. B. Nilsson (Eds.), *Artifacts and artificial science*. Almqvist & Wiksell.

DeLoach, S. A., & Valenzuela, J. L. (2007). An Agent-Environment Interaction Model. In L. Padgham & F. Zambonelli (Eds.), *Agent-oriented software engineering VII: 7th InternationalWorkshop, AOSE 2006, Hakodate, Japan, May 8, 2006 Revised and Invited Papers* (pp. 1-18). Springer.

Dewey, J. (1938). *Logic: The theory of inquiry*. Henry Holt.

Diguet, J.-P., Eustache, Y., & Gogniat, G. (2011). Closed-loop-based self-adaptive hardware/ software-embedded systems: Design methodology and smart cam case study. *ACM Transactions on Embedded Computing Systems, 10*(3), Article 38.

Dipert, R. R. (1995). Some issues in the theory of artifacts: Defining 'artifact' and related notions. *The Monist, 78*(2), 119-135.

Edwards, M., & Green, P. (2000). *An object oriented design method for reconfigurable computing systems.* Paper presented at the Design, Automation and Test in Europe Conference and Exhibition 2000.

Eekels, J., & Roozenburg, N. F. M. (1991). A methodological comparison of the structures of scientific research and engineering design: Their similarities and differences. *Design Studies, 12*(4), 197-203.

Fernandez, E. B., Cholmondeley, P., & Zimmermann, O. (2007). *Extending a secure system development methodology to SOA*. Paper presented at the 18th International Workshop on Database and Expert Systems Applications.

Fernandez, E. B., & Yuan, X. (2007). *Securing analysis patterns*. Paper presented at the 45th Annual Southeast Regional Conference, ACMSE 2007.

Fitzgerald, B., Russo, N. L., & Stolterman, E. (2002). *Information systems development: Methods in action*. McGraw-Hill.

Franch, X., Ralyté, J., Perini, A., Abelló, A., Ameller, D., Gorroñogoitia, J., . . . Susi, A. (2018). A Situational Approach for the Definition and Tailoring of a Data-Driven Software Evolution Method. *Proceedings of the Advanced Information Systems Engineering -30th International Conference, CAiSE 2018*.

Franssen, M., Kroes, P., Reydon, T. A. C., & Vermaas, P. E. (Eds.). (2014). *Artefact kinds: Ontology and the human-made world*. Springer.

Galal, G. H. (2001). From contexts to constructs: the use of grounded theory in operationalising contingent process models. *European Journal of Information Systems, 10*(1), 2-14.

Galal, G. H., & Paul, R. J. (1999). A qualitative scenario approach to managing evolving requirements. *Requirements Engineering, 4*(2), 92-102.

Ge, X., Paige, R. F., Polack, F. A. C., Chivers, H., & Brooke, P. J. (2006). Agile Development of secure web applications. *Proceedings of the 6th International Conference On Web Engineering*.

Georg, G., Mussbacher, G., Amyot, D., Petriu, D., Troup, L., Lozano-Fuentes, S., & France, R. (2015). Synergy between activity theory and goal/scenario modeling for requirements elicitation, analysis, and evolution. *Information and Software Technology, 59*, 109-135.

Goes, P. B. (2014). Design science research in top information systems journals. *MIS Quarterly, 38*(1), iii-viii.

Goldkuhl, G. (2004). Design theories in information systems: A need for multi-grounding. *Journal of Information Technology Theory and Application, 6*, 59-62.

Goldkuhl, G., & Lind, M. (2010). A Multi-Grounded Design Research Process. In R. Winter, J. L. Zhao, & S. Aier (Eds.), *Global perspectives on Design Science Research: DESRIST 2010* (pp. 45-60). Springer.

Goldkuhl, G., Lind, M., & Seigerroth, U. (1997). Method integration as a learning process. In N. Jayaratna, B. Fitzgerald, J.-M. Larrasquet, P. Checkland, E. Mumford, & T. Wood-Harper (Eds.), *Training and education of methodology practitioners and researchers: 5th International Conference of the British Computer Society Information System Methodologies Specialist Group* (pp. 15-26). Springer.

Green, P. N., & Edwards, M. D. (2000). Object oriented development method for reconfigurable embedded systems. *IEEE Proceedings: Computers and Digital Techniques, 147*(3), 153-158.

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly, 30*(3), 611-642.

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

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

Griss, M. L., Favaro, J., & d'Alessandro, M. (1998). *Integrating Feature Modeling with the RSEB*. Paper presented at the Fifth International Conference on Software Reuse.

Gustas, R., & Gustiené, P. (2008). Pragmatic-driven approach for service-oriented analysis and design. In P. Johannesson & E. Söderström (Eds.), *Information systems engineering: From data analysis to process networks* (pp. 97-128). IGI Global.

Ham, D.-H., Kim, J.-S., Cho, J.-H., & Ha, S.-J. (2004). MaRMI-III: A methodology for component-based development. *ETRI Journal, 26*(2), 167-180.

Harmsen, A. F. (1997). *Situational method engineering* (Doctoral dissertation). University of Twente, The Netherlands.

Harmsen, A. F., Brinkkemper, S., & Oei, H. (1994,). *Situational method engineering for information system project approaches.* Paper presented at the IFIP WG8.1 Working Conference CRIS'94.

Henare, A., Holbraad, M., & Wastell, S. (2007). Introduction: Thinking through things. In A. Henare, M. Holbraad, & S. Wastell (Eds.), *Thinking through things: Theorising artefacts ethnographically* (pp. 1-31). Routledge.

Henderson-Sellers, B. (2002). Process metamodelling and process construction: examples using the OPEN process framework (OPF). *Annals of Software Engineering, 14*(1-4), 341-362.

Henderson-Sellers, B., & Ralyté, J. (2010). Situational method engineering: State-of-the-art review. *Journal of Universal Computer Science, 16*(3), 424-478.

Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J., & Rossi, M. (2014). *Situational method engineering*. Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

Hirschheim, R., & Klein, H. K. (1994). Realizing emancipatory principles in information systems development: The case for ETHICS. *MIS Quarterly, 18*(1), 83-109.

Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems, 24*(1), 107-115.

Ingham, M. D., Rasmussen, R. D., Bennett, M. B., & Moncada, A. C. (2006). Generating requirements for complex embedded systems using State Analysis. *Acta Astronautica, 58*(12), 648-661.

Jayaratna, N. (1994). *Understanding and evaluating methodologies*. McGraw-Hill.

Jayaratna, N., Holt, P., & Wood-Harper, T. (1999). Criteria for methodology choice in information system development. *The Journal of Contemporary Issues in Business and Government, 5*(2), 30-34.

Johannesson, P. (1995). Representation and communication: A speech act based approach to information systems design. *Information Systems*, *20*(4), 291-303.

Jørgensen, K. A. (1998). Information modelling: Foundation, abstraction mechanisms and approach. *Journal of Intelligent Manufacturing, 9*(5), 571-581.

Karlsson, F. (2013). Longitudinal use of method rationale in method configuration: An exploratory study. *European Journal of Information Systems*, *22*(6), 690-710.

Karlsson, F., & Hedström, K. (2013). Evaluating end user development as a requirements engineering technique for communicating across social worlds during systems development. *Scandinavian Journal of Information Systems, 25*(2), 1-26.

Karlsson, F., & Wistrand, K. (2006). Combining method engineering with activity theory: Theoretical grounding of the method component concept. *European Journal of Information Systems, 15*(1), 82-90.

Karlsson, F., & Ågerfalk, P. J. (2004). Method configuration: Adapting to situational characteristics while creating reusable assets. *Information and Software Technology*, *46*(9), 619-633.

Karlsson, F., & Ågerfalk, P. J. (2007). *Multi-grounded action research in method engineering: The MMC case.* Paper presented at the Situational Method Engineering conference: Fundamentals and Experiences

Karlsson, F., & Ågerfalk, P. J. (2009). Towards structured flexibility in information systems development: devising a method for method configuration. *Journal of Database Management, 20*(3), 51-75.

Karlsson, F., & Ågerfalk, P. J. (2012). MC Sandbox: Devising a tool for method-user-centered method configuration. *Information and Software Technology, 54*(5), 501-516.

Kavakli, E., Kalloniatis, C., Loucopoulos, P., & Gritzalis, S. (2006). Incorporating privacy requirements into the system design process: The PriS conceptual framework. *Internet Research, 16*(2), 140-158.

Kim, Y., & Yun, H. (2006). *An Approach to modeling service-oriented development process*. Paper presented at the 2006 IEEE International Conference on Services Computing.

Kolkowska, E., Karlsson, F., & Hedström, K. (2017). Towards analysing the rationale of information security noncompliance: Devising a Value-Based Compliance analysis method. *Journal of Strategic Information Systems, 26*(1), 39-57.

Kuechler, B., & Vaishnavi, V. (2008). On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, *17*(5), 489-504.

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

Kumar, K., & Welke, R. J. (1992). Methodology Engineering: A proposal for situation specific methodology construction. In W. W. Cotterman & J. A. Seen (Eds.), *Challenges and Strategies for Research in Systems Development* (pp. 257-269). Wiley.

Lee, J., & Lee, H. (2008). Strategic agent based web system development methodology. *International Journal of Information Technology & Decision Making, 7*(2), 309-337.

Lee, J. S., Pries-Heje, J., & Baskerville, R. (2011). Theorizing in design science research. *Proceedings of DESRIST 2011: Service-Oriented Perspectives in Design Science Research*.

Lind, M. (2001). *Från system till process: Kriterier för processbestämning vid verksamhetsanalys* (Doctoral Thesis). Linköping University, Sweden.

MacLean, A., Young, R. M., & Moran, T. P. (1989). Design rationale: the argument behind the artifact. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Maier, J. R. A., & Fadel, G. M. (2009). Affordance based design: a relational theory for design.

Research in Engineering Design, 20(1), 13–27.

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

Michaelides, R., & Kehoe, D. (2007). *Internet communities and open innovation: An information system design methodology*. Paper presented at the 6th IEEE/ACIS International Conference on Computer and Information Science.

Middleton, P. (1998). Software quality by administration. *Software Quality Journal, 7*(3-4), 261-275.

Mingers, J. (2003). The paucity of multimethod research: a review of the information systems literature. *Information Systems Journal*, *13*, 233-249.

Mouratidis, H., & Giorgini, P. (2007). Security attack testing (SAT): Testing the security of information systems at design time. *Information Systems*, *32*(8), 1166-1183.

Mouratidis, H., Giorgini, P., & Manson, G. (2004). *Using security attack scenarios to analyse security during information systems design.* Paper presented at the Sixth International Conference on Enterprise Information Systems.

Nilsson, A. G. (1995). Evaluation of methodologies for information systems work, a historical perspective. In B. Dahlbom (Ed.), *The infological equation essays in honor of Börje Langefors*. Göteborg University.

Nunamaker, J. F., & Briggs, R. O. (2011). Toward a broader vision for information systems. *ACM Transactions on Management Information Systems, 2*(4), Article 20.

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

Offermann, P., Blom, S., Bud, U., & Levina, O. (2010). Proposal for components of method design theories: Increasing the utility of method design artefacts. *Business & Information Systems Engineering*, *2*(5), 295-304.

Offermann, P., Blom, S., Schönherr, M., & Bub, U. (2010). Artifact types in information systems design science: A literature review. In R. Winter, J. L. Zhao, & S. Aier (Eds.), *Global Perspectives on Design Science Research: DESRIST 2010* (pp. 77-92). Springer.

Oinas-Kukkonen, H. (1996). Method rationale in method engineering and use. In, S. Brinkkemper, K. Lyytinen, & R. J. Welke (Eds.), *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference On Method Engineering On Method Engineering* (pp. 87-93). Chapman & Hall.

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

Osmundson, J. S. (2000). A systems engineering methodology for information systems. *Systems Engineering, 3*(2), 68-81.

Peffers, K., Tuunanen, T., & Niehaves, B. (2018). Design science research genres: Introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems, 27*(2), 129-139.

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.

Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, *53*(4), 317-343.

Pilemalm, S., & Timpka, T. (2008). Third generation participatory design in health informatics: Making user participation applicable to large-scale information system projects. *Journal of Biomedical Informatics, 41*(2), 327-339.

Punter, T., & Lemmen, K. (1996). The MEMA-model: Towards a new approach for Method Engineering. *Information and Software Technology*, *38,* 295-300.

Rahimian, V., & Ramsin, R. (2008. *Designing an agile methodology for mobile software development: A hybrid method engineering approach.* Paper presented at the Second International Conference on Research Challenges in Information Science.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. In J. Eder & M. Missikoff (Eds.), *Advanced Information Systems Engineering, 15th International Conference, CAiSE 2003* (pp. 95-110). Springer.

Ralyté, J., & Franch, X. (2018). Using contextual goal models for constructing situational Methods. In J. C. Trujillo, K. C. Davis, X. Du, Z. Li, T. W. Ling, G. Li, & M. L. Lee (Eds.), *Conceptual modeling: 37th International Conference, ER 2018 Xi'an, China, October 22–25, 2018 Proceedings* (pp. 440-448). Springer.

Ralyté, J., & Rolland, C. (2001). An assembly process model for method engineering. *Proceedings of the 13th Conference on Advanced Information Systems Engineering*.

Reinhartz-Berger, I., Dori, D., & Katz, S. (2002). OPM/Web: Object-process methodology for developing web applications. *Annals of Software Engineering, 13*(1-4), 141-161.

Rolland, C. (2009). Method engineering: Towards methods as services. *Software Process Improvement and Practice, 14*(3), 143-164.

Rolland, C., & Prakash, N. (1996). A proposal for context-specific method engineering. In S. Brinkkemper, K. Lyytinen, & R. Welke (Eds.), *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering on Method Engineering* (pp. 191–208). Chapman & Hall.

Rosenman, M. A., & Gero, J. (1998). Purpose and function in design: from the socio-cultural to the technophysical. *Design Studies, 19*(2), 161-186.

Rossi, M., Ramesh, B., Lyytinen, K., & Tolvanen, J.-P. (2004). Managing evolutionary method engineering by method rationale. *Journal of Association of Information Systems, 5*(9), 356-391.

Royce, W. (1970). *Managing the development of large software systems.* Paper presented at the IEEE WESCON 26 conference.

Rumbaugh, J. (1995). What is a method? *Journal of Object Oriented Programming*, *8*(6), 10-16.

Russo, N. L., & Stolterman, E. (2000). Exploring the assumptions underlying information systems methodologies: Their impact on past, present and future ISM research. *Information Technology & People, 13*(4), 313–327.

Sandkuhl, K., & Seigerroth, U. (2019). Method engineering in information systems analysis and design: a balanced scorecard approach for method improvement. *Software & Systems Modeling*, *18*(3), 1833-1857.

Savage, A., & Mingers, J. (1996). A framework for linking soft systems methodology (SSM) and Jackson system development (JSD). *Information Systems Journal, 6*(2), 109-129.

Scerbo, M. W., Murray, W. B., Alinier, G., Antonius, T., Caird, J., Stricker, E., . . . Kyle, R. (2011). A path to better healthcare simulation systems: Leveraging the integrated systems design approach. *Simulation in Healthcare, 6*(7), 20-23.

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

Siau, K., & Tan, X. (2005). Improving the quality of conceptual modeling using cognitive mapping techniques. *Data & Knowledge Engineering, 55*(3), 343-365.

Simon, H. (1996). *Sciences of the artifical*. MIT Press.

Simonsson, M., Johnson, P., & Wijkström, H. (2007). *Model-based it governance maturity assessments with COBIT*. Paper presented at the 15th European Conference on Information Systems.

Soylu, A., & De Causmaecker, P. (2009). *Merging model driven and ontology driven system development approaches*. Paper presented at the 24th International Symposium on Computer and Information Sciences.

Spanoudakis, N., & Moraitis, P. (2008). The agent modeling language (AMOLA). In D. Dochev, M. Pistore, & P. Traverso (Eds.), *Artificial intelligence: Methodology, systems, and applications. 13th International Conference, AIMSA 2008 Varna, Bulgaria, September 4-6, 2008 Proceedings* (pp. 32-44). Springer.

Spanoudakis, N., & Moraitis, P. (2011). Using ASEME Methodology for Model-Driven Agent Systems Development. In D. Weyns & M.-P. Gleizes (Eds.), *Agent-Oriented Software Engineering XI: 11th International Workshop, AOSE 2010, Toronto, Canada, May 10-11, 2010, Revised selected papers* (pp. 106-127). Springer.

Standing, C. (2002). Methodologies for developing Web applications. *Information and Software Technology, 44*(3), 151-159.

Stylianou, A. C., Kumar, R. L., & Khouja, M. J. (1997). A total quality management-based systems development process. *Data Base for Advances in Information Systems, 28*(3), 59-71.

Tahara, Y., Ohsuga, A., & Honiden, S. (1999, 22-22 May 1999). *Agent system development method based on agent patterns*. Paper presented at the 1999 International Conference on Software Engineering.

Takeda, H., Veerkamp, P., Tomiyama, T., & Yoshikawam, H. (1990). Modeling design processes. *AI Magazine, 11*(4), 37-48.

Tolvanen, J.-P. (1998). *Incremental method Engineering with modeling tools: Theoretical principles and empirical evidence* (Doctoral thesis). University of Jyväskylä, Finland.

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). Focus groups for artifact refinement and evaluation in design research. *Communications of the Association for Information Systems, 26*, 599-618.

Vaishnavi, V., Kuechler, B., & Petter, S. (2004). Design research in information systems. http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf

van de Weerd, I., Brinkkemper, S., Souer, J., & Versendaal, J. (2006). A situational implementation method for web-based content management system-applications: Method engineering and validation in practice. *Software Process Improvement and Practice, 11*(5), 521-538.

van Hee, K., Hidders, j., Houben, G.-J., Paradaens, J., & Thiran, P. (2009). On the relationship between workflow models and document types. *Information Systems, 34*(1), 178-208.

van Slooten, K., & Brinkkemper, S. (1993). A method engineering approach to information systems development. *Proceedings of IFIP WG 8.1: Information systems development process*

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

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: guidelines for conducting mixed methods research in information systems. *MIS Quarterly, 37*(1), 21-54.

Verbeek, P.-P. (2005). *What things do. Philosophical reflections on technology, agency and design*. Pennsylvania State University Press.

Vidgen, R. (2012). Constructing a web information system development methodology. *Information Systems Journal, 12*(3), 247–261.

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

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly, 26*(2), 13-22.

Weerakkody, G., & Ray, P. (2003). CSCW-based system development methodology for health-care information systems. *Telemedicine Journal and e-Health, 9*(3), 273-282.

Weigand, H., & de Moor, A. (2003). Workflow analysis with communication norms. *Data & Knowledge Engineering, 47*(3), 349-369.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering.* Springer.

Winter, R. (2014). Towards a framework for evidence-based and inductive design in information systems research. In M. Helfert, B. Donnellan, & J. Kenneally (Eds.), *Design Science: Perspectives from Europe: EDSS 2013* (pp. 1-20). Springer.

Wood, M. F., & Deloach, S. A. (2001). An Overview of the multiagent systems engineering methodology. In P. Ciancarini & M. J. Wooldridge (Eds.), *Agent-oriented software engineering: First International Workshop, AOSE 2000 Limerick, Ireland, June 10, 2000: Revised papers* (pp. 207-222). Springer.

Zhuge, H. (2003). Component-based workflow systems development. *Decision Support Systems, 35*(4), 517-536.

Appendix A

Although the general research method of our literature review is straightforward, it was not a mechanical process; thus, several issues arose, which are detailed below. The research process was as follows:

1. We started with a trial search in Scopus and Web of Science and compared the number of papers found in both databases. During this trial search we used an initial set of keywords that was later refined. Our initial searched generated the following results in Scopus and Web of Science respectively: “systems development methodology” (514/54), “systems development method” (380/42), “systems engineering method” (316/22), “systems engineering methodology” (335/73), “information systems design method” (17/2), “information systems design methodology” (62/4). Based on this comparison we decided to use Scopus for our literature review as it generated a larger set of potential papers.
2. Table A1 shows the combination of search criteria that were used when searching in SCOPUS; search fields included paper title, abstract, and keywords. The table has two columns. The left-hand column contains the search criteria used. The next column shows the total number of papers resulting from each search. We searched for papers published between 1992 and 2017. The year 1992 was selected because Nunamaker et al. (1991) published their paper on using systems development in information systems research, which includes early ideas of design science. Hence, in 1992 it was possible to refer to this their paper when presenting the research method for a study. The use of multiple search queries resulted in a gross list of 6,486 research papers including duplicates. After eliminating duplicates, we ended up with a net list of 5,520 potential papers.
3. The net list of potential papers was sorted based on number of citations in order to focus on the most influential studies.
4. The abstract of each paper was read and an initial decision was made as to whether the research seemed related to development of information systems development method (ISDM). We continued to read abstracts until we had elicited 100 potential studies.
5. All selected papers were read in detail to (1) assess if the research actually was about the development of ISDM, and (2) to search for research methods. We analyzed both the authors’ description of the research method and interpreted the actual research method used based on what was shown in the paper. This detailed reading resulted in 53 studies; we present a detailed analysis in Table A2. The left-hand column shows the authors. The second column shows the outlet where the paper was published. The third column shows the classification of the described research method, subdivided into the research method for generating and validating the ISDM. The right-hand column shows the classification of the actual research method, also subdivided into the research method for generating and validating the ISDM.

We classified the papers using a modified version of Mingers’s (2003) classification framework. Mingers (2003) originally identified 13 types of research methods, to which we have added three. The first is design science, which has received increased attention during recent years in IS research—most notably, after Mingers’ (2003) framework was created. Moreover, Mingers (2003) did not include subjective/argumentative or literature review in his framework because he only analyzed empirical papers. In total the modified framework contains 16 types of research methods. In addition, we included two non-research method categories. First, a “no research method mentioned” category was included to capture cases where the authors did not give an account for their research method, i.e., not accounted for in a research method section or elsewhere in the paper. Thus, this category was used for categorizing how scholars have labeled their research method. Second, an “unclear method” category was included to capture cases where we were unable to classify the actual research method used, i.e., based on what steps were executed in the paper. This category focuses on what was actually done, instead of how the scholars labeled their research method (c.f. no research method mentioned). In total, our framework includes 18 classes.

Table A1. Search Criteria and Search Results

Search criteria	Number of papers
"Systems development method"	380
"Systems development methodology"	514
"Systems engineering method"	316
"Systems engineering methodology"	335
"Systems design method"	702
"Systems design methodology"	1,583
"Systems development approach"	167
"Systems engineering approach"	1,270
"Systems design approach"	582
"Method engineering" & "Systems development"	90
"Method engineering" & "Systems engineering"	66
"Situational method engineering" & "Systems development"	21
"Situational method engineering" & "Systems engineering"	13
"Methodology engineering" & "Systems development"	6
"Methodology engineering" & "Systems engineering"	3
"Situational methodology engineering" & "Systems development"	0
"Situational methodology engineering" & "Systems engineering"	0
"Method engineering" & "Software engineering"	159
"Situational method engineering" & "Software engineering"	47
"Methodology engineering" & "Software engineering"	6
"Situational methodology engineering" & "Software engineering"	0
"Method engineering" & "Software development"	127
"Situational method engineering" & "Software development"	49
"Methodology engineering" & "Software development"	4
"Situational methodology engineering" & "Software development"	0
"Method engineering" & "Method construction"	30
"Methodology engineering" & "Methodology construction"	0
"Situational method engineering" & "Method construction"	16
"Situational methodology engineering" & "Methodology construction"	0
Total sum including duplicates	*6,486*

Table A2. Details analysis of Existing Research on Development of ISDMs

Author(s)	Outlet	Described research method		Actual research method	
		Generating	Validating	Generating	Validating
Ahituv & Neumann (2002)	Journal of Computer Information Systems	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Balmelli, Brown, Cantor, & Mott (2006)	IBM Systems Journal	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Bayuk & Horowitz (2011)	Systems Engineering	No method mentioned	No method mentioned	Unclear method	Unclear method
Ben Ayed, Ltifi, Kolski, & Alimi (2010)	Decision Support Systems	No method mentioned	Case study	Subjective/argumentative	Case study
Beynon-Davies & Holmes (1998)	IEEE Proceedings: Software	No method mentioned	Case study	Subjective/argumentative	Case study
Biffl, Moser, Mordinyi, & Wahyudin (2008)	International Workshop on Software Quality	No method mentioned	Case study	Subjective/argumentative	Experiment
Carter, Whyte, Birchall, & Swatman (1997)	BT Technology Journal	No method mentioned	No method mentioned	Case study	Unclear method
Champion & Stowell (2002)	Information Systems Frontiers	Action research	Action research	Action research	Action research
Cossentino & Seidita (2005)	Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications	Method engineering	Experiment	Method engineering	Experiment
D'Aubeterre et al. (2008)	European Journal of Information Systems	Design science	Design science, Experiment	Design science	Design science, Experiment
DeLoach & Valenzuela (2007)	Agent-Oriented Software Engineering VII: 7th International Workshop	No method mentioned	No method mentioned	Method engineering	Experiment
Diguet, Eustache, & Gogniat (2011)	ACM Transactions on Embedded Computing Systems	No method mentioned	No method mentioned	Unclear method	Simulation
Edwards & Green (2000)	The Design, Automation and Test in Europe Conference and Exhibition 2000	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Fernandez et al. (2007)	18th International Workshop on Database and	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method

	Expert Systems Applications				
Fernandez & Yuan (2007)	45th Annual Southeast Regional Conference	No method mentioned	No method mentioned	Unclear method	Experiment
Galal (2001)	European Journal of Information Systems	No method mentioned	Case study	Subjective/argumentative	Case study
Galal & Paul (1999)	Requirements Engineering	No method mentioned	No method mentioned	Subjective/argumentative	Case study
Ge et al. (2006)	6th International Conference on Web engineering	No method mentioned	Case study	Subjective/argumentative	Experiment
Georg et al. (2015)	Information and Software Technology	No method mentioned	Case study	Subjective/argumentative, Method engineering	Case study
Green & Edwards (2000)	IEEE Proceedings: Computers and Digital Techniques	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Griss, Favaro, & d’Alessandro (1998)	5th International Conference on Software Reuse	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Gustas & Gustiené (2008)	Information Systems Engineering: From Data Analysis to Process Networks	No method mentioned	Case study	Subjective/argumentative	Experiment
Ham et al. (2004)	ETRI Journal	No method mentioned	Survey, Qualitative content analysis	Subjective/argumentative	Survey, Qualitative content analysis
Hirschheim & Klein (1994)	MIS Quarterly	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Ingham et al. (2006)	Acta Astronautica	No method mentioned	No method mentioned	Subjective/argumentative	Experiment
Johannesson (1995)	Information Systems	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Jørgensen (1998)	Journal of Intelligent Manufacturing	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Kavakli et al. (2006)	Internet Research	No method mentioned	Case study	Subjective/argumentative, Method engineering	Case study
Kim & Yun (2006)	3rd IEEE International Conference on Services Computing	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Lee & Lee (2008)	International Journal of Information Technology & Decision Making	No method mentioned	Case study	Subjective/argumentative	Case study

Michaelides & Kehoe (2007)	6th IEEE/ACIS International Conference on Computer and Information Science	No method mentioned	Case study	Subjective/argumentative	Case study
Mouratidis, Giorgini, & Manson (2004)	6th International Conference on Enterprise Information Systems	No method mentioned	Case study	Subjective/argumentative	Case study
Mouratidis & Giorgini (2007)	Information Systems	No method mentioned	Case study	Subjective/argumentative	Case study
Osmundson (2000)	Systems Engineering	No method mentioned	No method mentioned	Subjective/argumentative	Experiment
Rahimian & Ramsin (2008)	2nd International Conference on Research Challenges in Information Science	Method engineering	No method mentioned	Method engineering	Unclear method
Reinhartz-Berger et al. (2002)	Annals of Software Engineering	No method mentioned	Case study	Subjective/argumentative	Experiment
Pilemalm & Timpka (2008)	Journal of Biomedical Informatics	Action research	Action research	Action research	Action research
Savage & Mingers (1996)	Information Systems Journal	No method mentioned	Case study	Subjective/argumentative	Experiment
Scerbo et al. (2011)	Simulation in Healthcare	No method mentioned	No method mentioned	Unclear method	Unclear method
Siau & Tan (2005)	Data & Knowledge Engineering	No method mentioned	Case study	Subjective/argumentative	Experiment
Soylu & De Causmaecker (2009)	24th International Symposium on Computer and Information Sciences	No method mentioned	No method mentioned	Unclear method	Unclear method
Spanoudakis & Moraitis (2008)	13th International Conference on Artificial Intelligence: Methodology, Systems, Applications	Method engineering	Case study	Method engineering	Case study
Spanoudakis & Moraitis (2011)	11th International Workshop on Agent-Oriented Software Engineering	No method mentioned	Case study	Subjective/argumentative, Method engineering	Case study
Standing (2002)	Information and Software Technology	No method mentioned	Interviews	Subjective/argumentative	Unclear method

Stylianou, Kumar, & Khouja (1997)	ACM SIGMIS Database: the DATABASE for Advances in Information Systems	No method mentioned	Case study	Subjective/argumentative	Experiment
Tahara, Ohsuga, & Honiden (1999)	1999 International Conference on Software Engineering	No method mentioned	No method mentioned	Subjective/argumentative	Experiment
van de Weerd et al. (2006)	Software Process Improvement and Practice	Design science, Method engineering	Design science, Case study	Design science, Method engineering	Design science, Case study
van Hee, Hidders, Houben, Paradaens, & Thiran (2009)	Information Systems	No method mentioned	Case study	Subjective/argumentative	Experiment
Vidgen (2012)	Information Systems Journal	Action research	Action research	Action research	Action research
Weerakkody & Ray (2003)	Telemedicine Journal and e-Health	No method mentioned	Case study	Unclear method	Case study
Weigand & de Moor (2003)	Data & Knowledge Engineering	Method engineering	No method mentioned	Subjective/argumentative, Method engineering	Experiment
Wood & Deloach (2001)	1st International Workshop on Agent-Oriented Software Engineering	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method
Zhuge (2003)	Decision Support Systems	No method mentioned	No method mentioned	Subjective/argumentative	Unclear method

Appendix B

This appendix presents the details behind our synthesis of influential prior research in method engineering (ME) and design science (DS). The synthesis and its details are presented in Table B1, which has six columns. The left-hand column contains our synthesis of existing research. It is shown as ME-DS elements, comprising process elements, that is activity and product elements specified as input(s) and outputs(s). The second to fourth columns show examples of process and product elements in ME and DS research on which we built our ME-DS elements. The fifth column contains the authors. Finally, the sixth column shows the origin of the reference.

Table B1. Synthesis of Process Elements from Method Engineering and Design Science

ME-DS element	Process element	Product element		Authors	Discipline
	Activity	Input	Output		
Activity: Identify ISD method problem and motivate *Input*: ISD problems, needs and ideas, and knowledge base *Output*: Method rationale (research question)	Awareness of problem	New developments in industry, identification of problems within a reference discipline	Proposal, formal or informal, for a new research effort, criteria	Vaishnavi, Kuechler, & Petter (2004)	DS
	Characterization of project	Project environment	Project characteristics	Brinkkemper (1996)	ME
	Construct a conceptual framework, develop a system architecture	-	Justification of research questions pursued, requirements	Nunamaker et al. (1991)	DS
	Describe the class of goals to which the theory applies	Class of problems	Metarequirements, kernel theories	Walls et al. (1992)	DS
	Diagnosing	Difficulties	Vision	Karlsson & Ågerfalk (2007)	ME
	Elicitation and negotiation of method requirements	Development situation	Method requirements, project characteristics	Karlsson & Ågerfalk (2012)	ME
	Framing research activities	Business needs	Relevance and importance of problem	Hevner et al. (2004)	DS
	Identifying project characteristics	Project situation	Project characteristics	Karlsson & Ågerfalk (2004)	ME
	Identify problem and motivate	Practical problems	Defined problem, justification the value of a solution	Peffers et al. (2007)	DS
	Problem characterization	Project situation	Orientation, scope	Punter & Lemmen (1996)	ME
	Problem formulation	Practitioners, end-users, researchers, existing technologies, existing theories, and/or review of prior research	Research opportunity, research questions, identified problem class, potential theoretical contribution, prior technical advances, organizational commitment, roles	Sein et al. (2011)	DS
	Mapping situational factors to requirements	Stakeholders, situational factors, Situational Factors to	Requirements model	Agh & Ramsin (2016)	ME

		Requirements method base			
	Specify method requirements	Activities that needs to be supported, engineering goals	Method requirements	Ralyté & Rolland (2001)	ME
	Specifying situational method requirements	Activities that needs to be supported, engineering goals	Method requirements specified as contextual goal model	Franch et al. (2018)	ME
Activity: ISD method theorizing *Input*: Method rationale, knowledge base *Output*: Method rationale perspective, goals, concepts, conceptual structure, functional patterns)	Construct a conceptual framework	Requirements	Conceptual framework, mathematical models, methods	Nunamaker et al. (1991)	DS
	Define the objectives for a solution	Problem definition, current solutions	Objectives of a solution	Peffers et al. (2007)	DS
	Design theorizing	General requirements	Explanatory design theory	Baskerville & Pries-Heje (2010)	DS
	Internal grounding	Design theory	Modified design theory	Goldkuhl (2004)	DS
	Method theorizing	Method	Method design theory	Offermann, Blom, Bud, et al. (2010)	DS
	Suggestion	Proposal, formal or informal, for a new research effort, existing knowledge/theory base	Tentative design	Vaishnavi et al. (2004)	DS
	Theoretical grounding	External theories	Design theory	Goldkuhl (2004)	DS
Activity: Method engineering *Input*: Method rationale, knowledge base *Output*: ISD method	Artifact construction	Metarequirements, kernel theories	Design method, metadesign	Walls et al. (1992)	DS
	Analyze, design and build	Conceptual framework, mathematical models, methods, experiences gained	Prototype	Nunamaker et al. (1991)	DS
	Assembly	Selected fragments	Method advice	Punter & Lemmen (1996)	ME
	Assembly of method fragments	Method fragments	Situational method	Brinkkemper (1996)	ME
	Associate selected method chunks	Method chunks,	ISD method	Ralyté & Rolland (2001); Rolland (2009)	ME
	Build	Meta-requirements, kernel theories	Artifact (construct, model, method, instantiation)	March & Smith (1995)	DS
	Building, Intervention, and Evaluation*	Research opportunity, research questions, identified problem class, potential	Design principles, realized design of the artifact including	Sein et al. (2011)	DS

		theoretical contribution, prior technical advances, organizational commitment, roles	refinements to artifact, participants' experiences		
	Construct a method chunk	Method requirements	Method chunk	Ralyté & Rolland (2001)	ME
	Design and development	Objectives of a solution, knowledge of theory	Artifact	Peffers et al. (2007)	DS
	Develop/build	Business needs, applicable knowledge	Artifacts	Hevner et al. (2004)	DS
	Development	Tentative design	Artifact, knowledge constraints	Vaishnavi et al. (2004)	DS
	Integrate selected method chunks	Method chunks, method requirements	ISD method	Ralyté & Rolland (2001)	ME
	Matching	Orientation, scope, method base	Selected fragments	Punter & Lemmen (1996)	ME
	Method configuration (MMC)	Base method, configuration packages, configuration templates, project characteristics	ISD method	Karlsson (2013); Karlsson & Ågerfalk (2004, 2009, 2012)	ME
	Method configuration (Model-driven)	Method model, technical fragments	Executable ISD method	Cervera (2015); Cervera et al. (2012); Cervera et al. (2015)	ME
	Method definition	SPEM 2.0 meta-model, conceptual fragements	Method model	Cervera (2015); Cervera et al., (2012); Cervera et al. (2015)	ME
	Pattern-based model-driven selection of method fragments	Requirements model, requirements to method fragments method base	High-level model of target methodology, technique-independent methodology model, technique-specific methodology model, situational method	Agh & Ramsin (2016)	ME
	Process configuration	Base method, rules	Project specific method	Bajec et al. (2007)	ME
	Select method chunk	Method requirements, method base	Method chunks	Franch et al. (2018); Ralyté & Rolland (2001)	ME
	Selection and construction	Repository of process components, needs	Project/organization specific method	Henderson-Sellers (2002)	ME
	Selection of method fragments	Project characteristics, method base	Method fragments	Brinkkemper (1996)	ME

	Building, intervention, and evaluation*	Research opportunity, research questions, identified problem class, potential theoretical contribution, prior technical advances, organizational commitment, roles	Design principles, realized design of the artifact including refinements to artifact, participants' experiences	Sein et al. (2011)	DS
Activity: ISD method demonstration *Input*: ISD method *Output*: ISD situational results	Demonstration	Artifacts	Metrics, analysis knowledge	Peffers et al. (2007)	DS
	Empirical grounding	Prescribed action	Effect	Goldkuhl (2004)	DS
	Experiment, observe and evaluate the system	Prototype system	Experiences gained	Nunamaker et al. (1991)	DS
	Project performance	Situational method	Accumulated experience	Brinkkemper (1996)	ME
	Project performance	Method advice	Refinements	Punter & Lemmen (1996)	ME
	Realization of project	Situational method	Project experience	Karlsson & Ågerfalk (2004)	ME
	Analysis of method use	Experiences	Method changes, updated method engineering criteria	Rossi et al. (2004)	ME
Activity: Formal evaluation *Input*: ISD situational results, method rationale *Output*: Method rationale lessons learned), change requests	Evaluate	Artifacts	Results, assessment of relevant quality attributes	Hevner et al. (2004)	DS
	Evaluate	Artifact, criteria	Observed performance, progress	March & Smith (1995)	DS
	Evaluation	Metrics, analysis knowledge, objectives of a solution	Disciplinary knowledge	Peffers et al. (2007)	DS
	Evaluation	Criteria	Performance measures, knowledge constraints	Vaishnavi et al. (2004)	DS
	Reflection	Experiences	New fragments, modified base method, modified rules	Bajec et al. (2007)	ME
	Reflection and learning, formalization of learning	Design principles, realized design of the artifact including refinements to artifact, participants' experiences	Reflection of design/redesign, assessed stated goals, generalized problem, generalized solution, design principles	Sein et al. (2011)	DS

	Testing	Meta-requirements, design method, meta design	Tested design product/process hypotheses	Walls et al. (1992)	DS
Activity: Communication *Input*: Research problem, method rationale, ISD method, ISD situational results, lessons learned, change requests *Output*: Publications, presentations	Conclusion	Performance measures	Results, "firm" facts, "loose ends"	Vaishnavi et al. (2004)	DS
	Communicate	Results, assessment of relevant quality attributes	Presentation to technology-oriented and management-oriented audiences	Hevner et al. (2004)	DS
	Communication	Conducted DS study	Scholarly publications	Gregor & Hevner (2013)	DS
	Communication	Disciplinary knowledge	Scholarly publications	Peffers et al. (2007)	DS
Note: * This process element covers three phases since Sein et al. (2011) treat engineering, demonstration and evaluation as one integrated phase.					

Appendix C

This appendix presents the details behind our analysis of how retrospective demonstration has been used in prominent design science/information systems papers. The analysis is presented in Table C1, which has four columns. The left-hand column shows the authors. The second column shows the outlet where the paper was published. The third column shows the purpose/motive of the retrospective demonstration. Finally, the right-hand column, shows the number of cases used in the retrospective and the selection principle.

Table C1. Overview how Retrospective Demonstration has Been Used in Design Science/Information Systems Papers

Author(s)	Outlet	Purpose/motive	Case selection
Hevner et al. (2004)	MIS Quarterly	To *illustrate* the application of the proposed DS guidelines. The goal is not a critical evaluation of the quality of research contributions, but rather to *illuminate* the DS guidelines (although assessment statements occur).	3 exemplar articles for analysis from 3 different IS journals.
Gregor (2006)	MIS Quarterly	Examples are given as *illustrations* of five proposed theory types. The examples given for each theory type are *analyzed* for evidence of all seven theory components.	5 theory papers selected, one for each theory type.
Peffers et al. (2007)	Journal of Management Information Systems	To *demonstrate the use* of the proposed Design Science Research Methodology (DSRM) and to *transfer* established DS research into a formal research framework in order to illustrate its applicability. The DSRM language is used to *interpret* the research processes actually used by the researchers to *determine* how well the DSRM *fits* with the research processes used. To use the case discussions to *evaluate* the DSRM.The case studies provide *useful templates* for researchers who want to apply DSRM	4 published IS research projects.
Baskerville & Pries-Heje (2010)	Business & Information Systems Engineering	To *examine* seven design theories in the IS literature and *explain* how each can be represented as explanatory design theories (as proposed by the authors).	Examples from 5 reference disciplines (design theories that range from highly behavioral to highly natural science-oriented disciplines). 7 prominent design theories in the IS literature.
Kuechler & Vaishnavi (2012)	Journal of the Association for Information Systems	To *illustrate* the use of the proposed theory development framework and to *demonstrate* the *potential value* of this level of theory (made through development of complete design theories following their proposal "Design-Relevant Explanatory/Predictive Theory").	2 published examples of DS research/IS.
Gregor & Hevner (2013)	MIS Quarterly	To *determine* if the knowledge claims in case articles were *classifiable* according to the proposed DS contribution framework. A *proof-of-concept demonstration* of the applicability of the proposed patterns	Sample of 13 published DS research papers from one leading IS journal (MISQ).

About the Authors

Göran Goldkuhl, PhD, is a professor of information systems at Linköping University (Sweden). He is also associated with Campus Gotland at Uppsala University, and holds an honorary doctorate from Örebro University. He serves on the AIS SIGPRAG Board. His research interests cover areas such as qualitative and pragmatic research methodologies, practice research, design science, action research, case study methodology, innovation and change management, IS theorizing, practice theory, work and IT codesign, method engineering, business process modelling, communication analysis, digital service design, digital interoperation, user-interface design, IS evaluation, e-government, values and law in digitalization. He has published in conference proceedings of ALOIS, ECIS, DESRIST, ICIS, ISD, LAP, POEM, among others. He has also published in journals such *as Australasian Journal of Information Systems, Business Process Management Journal, Communications of ACM, Communications of the Association for Information Systems, Data & Knowledge Engineering, European Journal of Information Systems, Government Information Quarterly, Information and Organization, Information Systems and e-Business Management, International Journal of Qualitative Methods, Journal of Information Technology Theory and Application, Scandinavian Journal of Information Systems*, *Semiotica,* and *Transforming Government*, among others.

Fredrik Karlsson is a professor of informatics at Örebro University, Sweden. He received his PhD degree in information systems development at Linköping University in 2005. He is currently acting as head of Örebro University's School of Business. His areas of specialization include method engineering, use of systems development methods, information security management, electronic government and how values can be used to understand different sets of behavior in complex situations. His research interests in these areas focuses on providing method and computerized tool support for different roles. He often conducts research in collaboration with business organizations. He has published in conference proceedings of CAISE, ISD, ECIS, IFIP SEC among others. His research has also appeared in a variety of information systems journals such as *Computers & Security, European Journal of Information Systems, Government Information Quarterly, Strategic Journal of Information Systems,* and *Scandinavian Journal of Information Systems*. He currently serves as associate editor for *European Journal of Information Systems.*