Towards a Framework for Evidence-Based and Inductive Design in Information Systems Research*

Robert Winter

University of St. Gallen, Institute of Information Management
Müller-Friedberg-Strasse 8, CH-9000 St. Gallen
robert.winter@unisg.ch

Abstract. Discussions about design science research as an alternative or at least complementary approach to the dominant descriptive research paradigm have not only taken place in information systems research, but also in organizational sciences, accounting, operations, and other business research disciplines. In contrast to the descriptive research paradigm that can be taken over from sociology and psychology in a very mature state, the problem-solving paradigm is comparably new to business research. Not only have different variants of this approach (e.g. design as search, evidence-based design, emergent design) been proposed and applied that appear to be incompatible at first sight. Descriptive research and design science research also appear to have no common ground and no synergy potentials. As a consequence, not only seem improvement and change ('design and engineering') often detached from phenomenon analysis and theory building. The role of 'un-grounded', innovative practices is also not clear. In order to provide a common ground and support a better integration of descriptive and design-oriented research in information systems, we propose a framework that is not only organized along the well-known 'descriptive vs. prescriptive' dimension, but also introduces a generality dimension. The four resulting quadrants 'operations', 'explanations', 'technologies' and 'solutions' allow not only to position all central objects of research, but also to position and better integrate research activities and iterations. This extends not only to 'deductive' design (solution search based as well as evidence-based), but also to 'inductive' design.

1 Introduction

Information systems (IS) can be studied from two fundamentally different perspectives. The descriptive perspective aims at analyzing, explaining and / or at least partially predicting technology use by organizations and organizational actors as empirical phenomena. This research perspective dominates in social sciences and humanities. Its dominant outcomes are "theories for analysis, explanation and / or prediction" [2]. Examples from IS research are studies that explain why IS are used or

* The proposed framework is regarded to be applicable for information systems research as well as in related fields. In [1] the author describes the framework's application for organizational design and engineering. Most figures are identical.

M. Helfert et al. (Eds.): EDSS 2013, CCIS 447, pp. 1–20, 2014.

not continuously used [e.g. 3]. Such explanations may or may not include predictions of continuous use under changed conditions.

In contrast, the problem-solving, 'design' or 'engineering' perspective aims at improving IS use. Its dominant outcomes are "theories for design and action" [2]. Baskerville and Pries-Heje [4] present Walls et al.'s design theory for vigilant EIS [5] as an exemplar because it shows that certain templates, directives, etc. will ensure a consistent vision that allows executives to deal with their broad, diverse and variable issues. We will use the term design to characterize design as well as engineering aspects in the following. In contrast to the descriptive perspective, the design perspective is not restricted to analyzing existing empirical phenomena and is not purpose-free. Instead, it aims at creating 'better worlds', the ends for which effective means are proposed.

While the descriptive perspective dominates in important IS research communities (e.g. in the United States), the design perspective dominates in others [6]. Due to different positions, goals and outputs, it is not surprising that descriptive research and design research are considered to be disjunctive approaches [7, 8]. As an example, constructs in descriptive research are usually identified and validated in a completely different way than their counterparts in design research, leading to knowledge components that might be incompatible so that the findings built on such constructs cannot be combined or integrated. An exemplary 'mismatch' is the incompatibility between technology acceptance model constructs – like intention to use – that describe by what factors acceptance can be captured in a causal model on the one hand, and constructs that are used to describe methods for IS introduction on the other hand.

Missing integration between descriptive and design research impede cumulative research within and between communities, not only in IS research. This article therefore deals with the conceptual integration of descriptive and design research. A central challenge is to identify or propose a common conceptual basis that serves as a foundation for both research perspectives. Thus the first research goal of this article is to analyze related work on research frameworks regarding their integration suitability.

A second research question is related to the form in which generalized problem solutions are created in design research. Often a deductive approach is regarded as essential, i.e. proposed solutions should be based on generalized descriptive knowledge. In their "anatomy of a design theory", Gregor and Jones demand to always specify a "justificatory" or "kernel" theory, i.e. "the underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design" [9]. Design, however, not always needs to be done in such an 'evidence-based' manner only. The inherently inductive approach of identifying and reusing patterns has not only been proposed in civil engineering [e.g. 10], but also in software engineering [e.g. 11] and organizational engineering [see e.g. 12, 13] – i.e. in reference disciplines of design-oriented IS research. The apparent benefit of inductive design is that yet 'unexplained' or 'un-grounded', innovative practices can be generalized and reused without have to take the 'detour' of descriptive theorizing. It is however unclear how inductive design should be structured from a conceptual research process perspective and which components of inductive design can be integrated with traditional design-oriented or with descriptive research activities. Thus the second research question of this article is how to integrate inductive design into the proposed conceptual framework.

Based on a common foundation of artifact types in social sciences and their use on the one hand, and generality levels of artifacts on the other, a two-dimensional model is introduced in the next section ‘Common Foundation’. The proposed model’s four quadrants ‘operations’, ‘explanations’, ‘technologies’ and ‘solutions’ allow not only to position all main objects of all mentioned research perspectives, but also to position and compare research processes and iterations in section ‘Design and Engineering Activities’. In section ‘De-Contextualization and Emergence’, inductive design activities are characterized and integrated into the proposed framework. The concluding section discusses the proposal’s contribution and suggests avenues for further research.

2 Common Foundation

In design science research for IS (DSR-IS), March and Smith’s [14] differentiation of constructs, models, methods and instantiations as artifact types is commonly accepted [15]. Hevner et al. characterize these artifact types as follows: “Constructs provide the language in which problems and solutions are defined and communicated […]. Models use constructs to represent a real world situation – the design problem and its solution space […]. Methods define processes. They provide guidance on how to solve problems, that is, how to search the solution space. […] Instantiations show that constructs, models, or methods can be implemented in a working system.” [16]. Since models can not only represent problem solution requirements or problem solutions (means or ends, problem-solving paradigm), but also represent the phenomena under analysis (descriptive paradigm), we differentiate between ‘problem or solution models’ and ‘descriptive models’.

As stated by Winter [6] it is important to understand the artifact types not as disparate concepts, but as an interlinked system. Chmielewicz’s [17] taxonomy may serve as a foundation to explain such linkages. He differentiates between four fundamentally different research approaches in social sciences which build upon another: (1) ontology building, (2) theory building, (3) technology building and (4) judgment. The respective research outcomes in Chmielewicz’s system are

- ontological facts (foundational concepts, e.g. constructs of a causal relationship, constructs of a problem requirements specification or constructs of a solution)
- theoretical statements (cause-effect relations, e.g. explanatory theories)
- technological statements (means-end relations, e.g. solution methods or solution models) and
- normative statements (object-value relations, e.g. evaluations of solution models).

Due to their conceptual differences, these types of outcomes can be regarded as fundamentally different artifact types. Descriptive models (theoretical statements) use constructs (ontological facts) as their building blocks. Problem or solution models as well as solution methods (technological statements) should use theory as explanatory justification. Actual solutions (model or method instantiations) are instantiated from technologies based on specific choices (judgment).

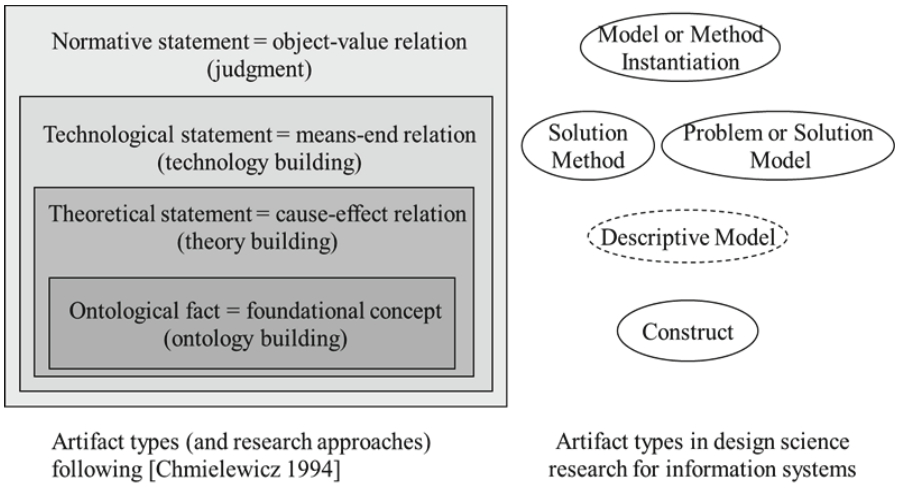

Fig. 1. Artifact types / research approaches in social sciences (left) and design science research for IS (right) [Based on 6, 18]

Fig. 1 relates the Chmielewicz taxonomy of artifact types and research approaches in social sciences (left) to the artifact types of the "Sciences of the Artificial" (DSR-IS, right). Foundational concepts can be related to constructs, theoretical statements can be related to descriptive models, means-end relations can be related to solution methods and problem / solution models, and object-value relations (= technologies chosen to achieve certain goals) can be related to model / method instantiations. The comparison supports three interesting insights [6]:

Firstly, the systems of research outcome types (and thus research activities) in social sciences and 'sciences of the artificial' seem to be more compatible than expected – given the fundamentally different perspectives of these approaches and the apparent lack of established common frameworks.

Secondly, descriptive models can and should be incorporated into the set of DSR-IS artifact types 'between' technological statements and ontological facts. This claim is supported by many authors [e. g., 5, 9, 19, 20] who argue that technology design should be informed by kernel / justificatory knowledge and, as a consequence, both should be based on the same conceptual foundation.

Thirdly, problem or solution models and solution methods are more closely related to each other than to other DSR-IS artifact types. It has in fact been argued that problem / solution models and solution methods are "two views of the same thing" [21]. While problem / solution models focus on design inputs and outputs – and imply procedural aspects –, solution methods focus on procedural aspects – and imply design outcomes. Some authors therefore propose to represent procedural aspects and outcomes in a more integrated forms, e. g. by process deliverable diagrams (for an exemplary application cf. [22]).

2.1 Descriptive vs. Prescriptive Artifacts

Descriptive models (including their constructs) are different from solution methods, problem / solution models and instantiations (including their constructs): Descriptive artifacts exist independently from any valuations or goals. As a consequence, (explanatory and / or predictive) theory building is aiming to propose primarily valid – and not necessarily useful – results. In contrast, solution methods as well as problem / solution models (including their constructs) are always related to certain (problem solution) goals, and instantiations are always created based on certain valuations and choices. As a consequence, technology development is aiming to propose primarily useful – and not necessarily valid - results.

Traditionally, only descriptive models that represent explanatory and / or predictive relations between constructs, have been designated as theory [cf. e. g. 23]. Since the term theory is claimed by all research paradigms, generic solution models and solution methods have also been designated theories in the context of DSR-IS [cf. e. g. 2, 9, 24, 25]. According to Gregor, the distinctive feature of a design theory is that it makes explicit prescriptions (e.g., construction guidelines, principles of form and function) for an artifact. Based on this specific feature, design theories can be understood as means-end relations according to the Chmielewicz taxonomy, as opposed to (explanatory) theories that are included as cause-effect relations in the taxonomy.

The question whether a design theory is just "effective practice" or has components whose validity can be proven, has been investigated by Baskerville and Pries-Heje [4]. They propose to separate a design theory into an explanatory and a practice component, designated as "explanatory design theory" and "design practice theory", respectively. From an explanatory point of view, design theory is "…a general design solution to a class of problems that relates a set of general components to a set of general requirements" [4] – this comes very close to Chmielewicz' understanding of generic means-end relations. Certain solution requirements can be interpreted as reasons for corresponding solution components. Certain solution components can be justified by corresponding solution requirements. While the explanatory design theory provides functional explanations for prescriptive artifacts, the design practice theory gives explicit prescriptions on how to design and develop an artifact, e.g. by applying solution methods and / or re-using (reference) solution components (e.g., patterns).

Theory is an important constituent for research, from a descriptive perspective as well as from a design perspective. It should be carefully differentiated whether "theory-type" statements relate cause and effect (explanatory and / or predictive theory) or relate means and ends (design theory). This line separates two 'worlds', the world of descriptive artifacts and the world of prescriptive artifacts.

2.2 Artifacts on Different Abstraction Levels

In the light of the huge amounts of highly diverse artifacts that are created both in design research and in design practice, the differentiation of descriptive and prescriptive artifacts seems not to be sufficient for a precise differentiation of research processes and outcomes. We therefore propose to additionally differentiate artifacts on different levels of abstraction. While instantiations represent one situated artifact implementation in context and time (e. g. a specific project plan or a specific workflow instance or a specific algorithm at a certain point in time), all other artifact types such as solution methods, solution models, descriptive models, or constructs can be instantiated by a set of more or less complex artifacts that are linked to more or less diverse goals, subject to more or less diverse contexts, valid in more or less points in time, etc.

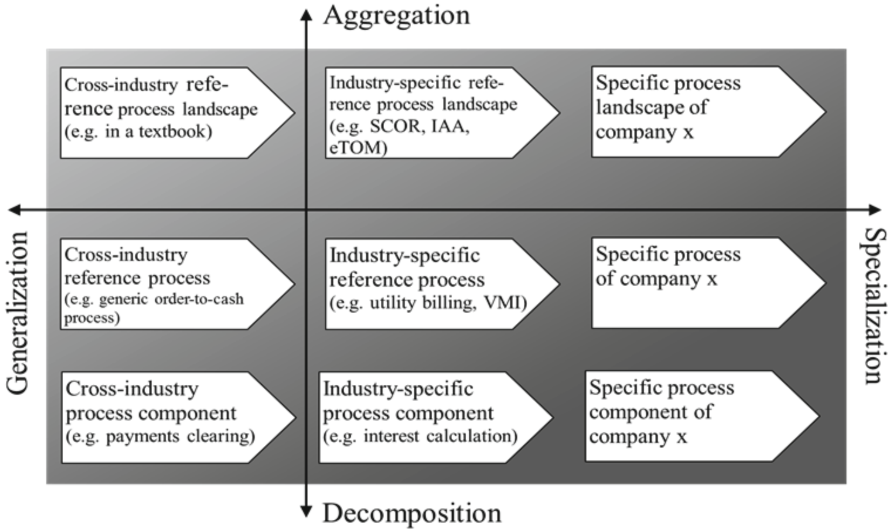

Fig. 2. Process models of different abstraction levels

In order to specify “more or less” abstraction, we refer to traditional data management approaches [e.g. 26] that differentiate at least a generalization / specialization and an aggregation / decomposition sub-dimension: While the level of generality indicates how many different instantiations the artifact allows, the level of aggregation indicates into how many components the artifact can be decomposed. MIT’s process compass [27] is a nice example to illustrate that generalization / specialization and aggregation / decomposition are orthogonal sub-dimensions which specify the abstraction level of – in this case – a process model. Fig. 2 illustrates the process compass idea. A lighter background color indicates more general and / or more aggregate process models. A darker background color indicates more specific and / or more decomposed process models. Some exemplary process models are positioned in Figure 2 to illustrate not only their different degree of generalization / specialization and of aggregation / decomposition, but also to show that these dimensions are independent and all combinations exist.

The proposed two-dimensional abstraction model cannot only be applied to solution models (like process models). Exemplified by Business Process Management (BPM), typical abstraction levels for a solution method are

- Generic: Generality level is “one size fits all”, i. e. the method is applicable to all processes in all organizations in all existing or possible worlds. Aggregation level is “one method covers all”, i. e. the method is comprised of process analysis, process control, continuous process development and maybe even more components and thus covers modeling, performance management, change management, etc. BPM methods on this abstraction level are e.g. found in textbooks or method handbooks.

- Archetypal: The method is applicable for all problem situations that share certain properties (e.g. process type, organization type, project type/goals, available resources and/or skills). Usually a small number of problem archetypes is differentiated that represent important, relevant *design problem classes* like e. g. small enterprises, a certain industry, or certain BPM goals like speed or throughput. BPM methods on this generality level might be derived by adapting abstract methods to the problem class at hand and / or by selecting certain components of abstract methods.
- Configurable: Based on either a refinement of archetypes or on a classification of real-world problems, a large number of problem configurations is differentiated whose solutions are created from reusable modules by configuration or aggregation. BPM methods on this level of generality might either be inductively created from “best practices” or constructed as adaption of more general, e.g. archetype-specific methods. For BPM, a configurable method has been proposed by Bucher and Winter [28, 29]. They differentiate four archetypes of BPM, five resulting BPM project types, and show how three important BPM project types can be aggregated from a set of 17 reusable method fragments.
- Situated: Generality level is “one of a kind”, i.e. the method is applicable only in a specific organization for a specific process at a specific point in time. Aggregation level is “specific technique”, i. e. only selected BPM aspects are covered. BPM methods (or better approaches) on this level of abstraction are either individually developed ‘on the fly’, or are instantiated from more abstract methods.

Theoretically, artifacts of every type can be represented on a literally unlimited number of abstraction levels. The generality and aggregation levels of constructs, descriptive models, problem or solution models, or solution methods are implied by specifying the respective scope or problem class, e. g. by focusing on design goals, application areas, problem characteristics, etc. In order to discover relevant focus dimensions, an empirical technique like the one proposed by Winter [30] can be used. By using principal component analysis on data of 47 BPM projects, Winter yielded four relevant focus dimensions for BPM: performance measurement maturity, process orientation maturity, process manager impact, and methodology and standard maturity [29]. By choosing more or less restrictive ranges for these four focus dimensions, a BPM problem class is defined for which respective descriptive or solution artifacts can be constructed. If every observed BPM approach in a company is represented in the four dimensional room spanned by the four discovered focus dimensions, a cluster analysis can be carried out to determine a reasonable number of clusters, i. e. design problem classes. The higher number of clusters is chosen, the larger the set of problem classes will be, and the less abstract will be respective descriptive and solution artifacts. Fig. 3 is a typical dendrogram-like tree diagram that results from agglomerative clustering and illustrates how artifacts on different abstraction levels are related.

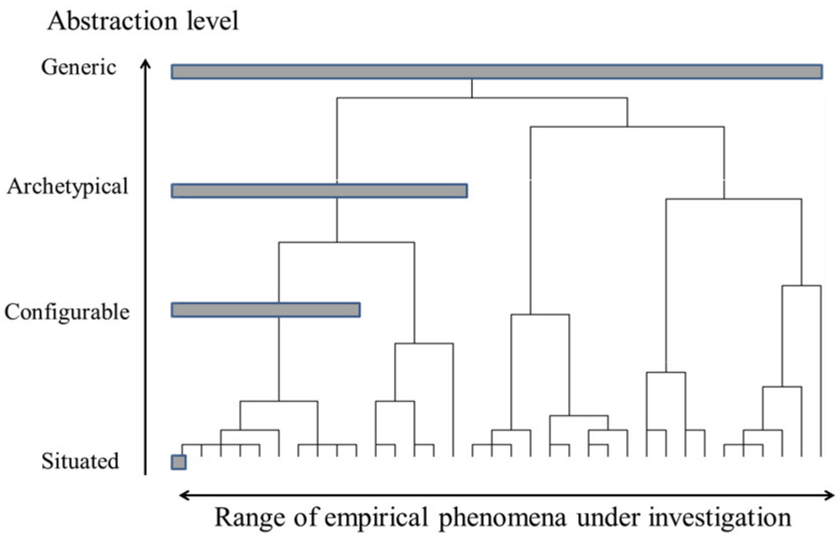

Fig. 3. Ultrametric tree visualization of artifact generality [adapted from 31]

The vertical dimension in Fig. 3 can be illustrated by characterizing four exemplary levels of abstraction of an solution model: The situated artifact's scope is limited to exactly one empirical phenomenon, e. g. a solution instance in a specific organization at a specific point in time. The configurable artifact's scope covers a certain range of phenomena delimited by a reusable set of description or solution components, e. g. a certain type of decision problems that can be solved by a parameterized algorithm. The archetypal artifact's scope covers a larger range of phenomena defined by a problem class context and certain analysis / design goals, e. g. BPM in large discrete manufacturing companies. The generic artifact's scope is the largest, covering an entire class of phenomena, e. g. performance management in commercial organizations. The (dis)similarity of two artifacts corresponds to the generality level of their link. If two artifacts are very similar, their link is represented on a low level of generality – and vice versa.

2.3 Four Artifact 'Worlds'

We have argued that artifacts can be differentiated regarding whether (1) they are descriptive or prescriptive and (2) regarding their level of abstraction – which can be expressed by their degree of aggregation and generalization. Since these two dimensions are sufficiently independent, their combination yields four different artifact 'worlds':

1. The *world of explanations* (quadrant E in Fig. 4) is the quadrant where artifact use is analyzed, explained and/or predicted on a general level. The most important artifacts in this quadrant are descriptive models including their conceptual base (construct definitions). An example for E-artifacts is the Technology Acceptance Model which explains/predicts IS acceptance by end users through (a) reconstruction of constructs like 'intention to use' or 'IS acceptance' and (b) empirical validation of a hypothetical dependency between these constructs that can be interpreted as causality ('acceptance of x by y is dependent to extent z on intention of y to use x'). E-artifacts are primarily created by descriptive research using social science techniques. Validity is the most desirable property of descriptive models. Among equally valid E-artifacts, those are usually higher valued that are more general and / or more comprehensive.
2. The *world of technologies* (quadrant T in Fig. 4) is the quadrant where solution models are related to problem models. The most important artifacts in this quadrant are design theories which, e. g. in the form of 'technological rules' or patterns or methods, link solution components (i. e. components of solution models or solution activities) to requirements (i. e. components of problem models). An example for T-artifacts is Activity Based Costing, a means to enable an organization to make appropriate (e. g. pricing, order acceptance) decisions in the presence of complex service processes, unsteady capacity usage and large indirect costs. T-artifacts are primarily created by problem-solving research using engineering techniques. Researchers might take an observer role, but can also be directly involved into solution design (action design research [32]). 'Effectiveness' is the most desirable property of technologies. Among equally effective T-artifacts, those are usually higher valued that are more general and / or more comprehensive.
3. The *world of solutions* (quadrant S in Fig. 4) is the quadrant where specific organizational design problems are addressed (and hopefully solved) by suitable artifacts. In contrast to quadrant T, such artifacts are not abstract any more, but adapted, configured/composed and/or implemented for solving a specific problem of a specific organization at a specific point in time – yet not implemented. The content of this quadrant can be characterized as '(concrete) problem solution' with specific problem-solving power being its most desirable property. Examples of S-artifacts are concrete process workflows to handle a business transaction (= instantiated process models) or concrete project plans (= instantiated solution methods) to achieve a business goal. S-artifacts are created in practice.
4. The *world of operations* (quadrant O in Fig. 4) is the quadrant where artifact application and use are described on an instance level. In contrast to quadrant S which covers constructed artifacts, implemented O-artifacts are 'in action'. In contrast to quadrant E, artifact use and its consequences are described individually on an instance level and not generalized. The content of this quadrant can be characterized as concrete day-to-day operations of organizations, with performance relative to the respective business goals being its most desirable property. Examples of O-artifacts are descriptions of the actual handling of a business transaction or the actual execution of a project. O-artifacts are created in practice.

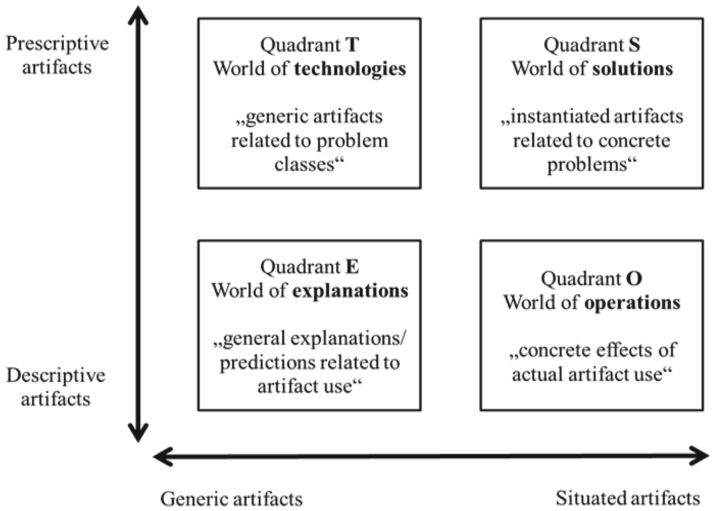

Fig. 4. Artifact 'worlds' quadrant model

3 Theory Building and Theory Application Activities

The four-world model is providing one common reference frame for representing abstract descriptive knowledge (quadrant E, e. g. explanatory theories), generalized solution knowledge (quadrant T, e. g. solution methods and solution models), concrete descriptive knowledge (quadrant O, e. g. observations of actual IS use) and concrete solutions (quadrant S, e. g. concrete workflows and plans). A common framework should however not only allow integrating all relevant artifact types, but also all activities and processes that create and use such artifacts. In the following, we therefore characterize 'intra-world' and 'inter-world' activities based on the proposed quadrant model and link the findings with existing reference process models from DSR-IS.

The most obvious activities are those that use and create artifacts within a world:

1. Within quadrant E, the body of (analytical / explanatory / predictive) theory knowledge can always be extended by combining or refining theories. Without data input from quadrant O (e.g. observations of innovative practices), new aspects of the phenomenon cannot be theorized. As a consequence, the significance of processes within quadrant O is limited to incremental progress.
2. Within quadrant T, the body of technologies can be extended by combining or refining problem / solution models or solution methods. Without input from quadrant S (e.g. analyzing novel solutions from practice) or quadrant E (e.g. applying new justificatory theory), however, the significance of processes within quadrant T is also limited to incremental progress (like e.g. improved modularization of a method).

3. Within quadrant S, the body of artifact instantiations can be extended by combining or refining solutions, or by applying existing technologies to new problems. Without input from quadrant T (e.g. new technologies) or quadrant O (e.g. observations of innovative practices), however, neither alternative, hopefully more effective solutions for existing problems can be found, nor can empirical evidence be used to enhance the effectiveness of solutions. A significant source of novel artifacts can however result from innovative solutions that have not been instantiated from existing technologies, but applied an invention outside our model, e. g. a new information technology or a new idea for structuring a task. It might also be possible to identify novel technologies but search.
4. Within quadrant O, the amount of knowledge about artifact implementation can be extended by collecting additional observations from the real world.

Hence the most important intra-world activities seem to be found in T (non-evidence-based solution innovation) and in O (exploration of innovative practices).

In a next step, we characterize activities that connect different worlds:

- From operations to explanations: Theory building is the process of generalizing observations (O-artifacts) in order to add generic descriptive analyses / explanations / predictions to the world of explanations, i. e. to create new E-artifacts from O-artifacts. An example is to collect a large number of actual IS acceptance observations in order to validate a general hypothesis about IS acceptance. We designate this activity O⇒E as it connects quadrant O to quadrant E.
- From explanations to technologies: DSR-IS nis the process of creating innovative, generic problem solutions (T-artifacts) that can be added to the world of technologies, ideally based on general descriptive analyses / explanations / prescriptions from the world of explanations (E-artifacts). This process is not a mere transformation, but requires to specify design goals, differentiate design situations, validate effectiveness / utility claims, etc. An example is to transform the Technology Acceptance Model into design guidelines for IS that avoid certain acceptance problems. We designate this activity E⇒T as is connects quadrant E to quadrant T.
- From technologies to solutions: Solution engineering means to situate, adapt, instantiate and maybe extend generic solutions from the world of technologies (T-artifacts) to create or improve concrete solutions to concrete design problems (S-artifacts). An example is to identify, prioritize and apply certain design principles, to identify and instantiate solution methods and/or to identify and adapt reference solution models in IS development. We designate this activity T⇒S as is connects quadrant T to quadrant S.
- From solutions to operations: Implementation / introduction means to put concrete project plans, concrete enterprise models (S-artifacts) etc. in action in a specific organization at a specific point in time to solve a specific problem (i. e. to create O-artifacts). An example is to run a project, to implement an IS or to execute a process. We designate this activity S⇒O as is connects quadrant S to quadrant O.

The above mentioned activities are illustrated based on the proposed artifact framework in Fig. 5.

When linked together, E⇒T, T⇒S and S⇒O can be interpreted as ‘evidence-based’ design. Similar to evidence-based medicine or evidence-based management [33], this means that solutions are systematically based on justificatory knowledge (cause-effect relations) as well as applicable technologies (means-end relations) – in contrast to purely ‘search-based’ design [e.g. 34].

O⇒E (theory building, the core activity of descriptive research) is the ‘missing link’ to complete a process cycle within the proposed framework. The resulting research process might start with making real-world observations (O), finding explanations (O⇒E), transforming these into innovative technologies (E⇒T), apply such innovations to real-world problems (T⇒S), implement these solutions (S⇒O), and finally evaluate how they perform in order to extend/revise explanations (O⇒E), enhance technologies (E⇒T), and so on. This chain of processes comes very close to a combination of widely accepted process models for DSR-IS (e.g. [35]) with the classical process of theory-building in social sciences (e.g. [23]).

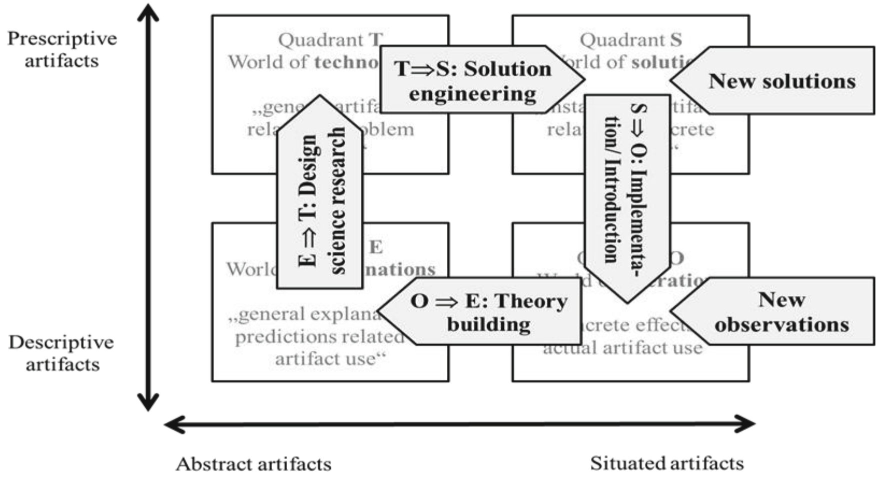

Fig. 5. Theory building and theory application in the proposed artifact framework

Our first research question aimed at integrating descriptive and design research activities. If O⇒E represents descriptive research and (E⇒T; T⇒S; S⇒O) represents design research, the proposed framework provides a conceptual foundation for connecting DSR-IS to theory-building in information systems research.

4 De-contextualization and Emergence

Does the ‘intertwining’ of theory building with DSR-IS only work in a ‘forward-engineering’, evidence-based way? Can innovative solutions only be created by situating, adapting, instantiating and maybe extending general technologies that rely on justificatory foundations – that themselves have been validated by observing existing phenomena? How can innovation be explained when there is no innovative design without justificatory explanations, and when explanations rely on observations of 'applied theory'?

An alternative understanding of design has been proposed by van Aken and others who endorse inductive design by empirical research on multiple case studies to determine existing best practice. E. g., van Aken and Nagel [36] use seven case studies to identify technological rules which solve the problems associated with the 'fuzzy front end' of the product development process. As Davies [37] summarizes this approach, "once a rule has been identified, it is then tested in a range of contexts, with adjustments being made when needed, until 'theoretical saturation' is reached and additional cases do not add anything to knowledge about when and how the rule works".

The induction of 'technologies' (i.e. means-end relations) has also been proposed under the 'pattern' label not only in civil engineering [e.g. 10] or software engineering [e.g. 11], but also in the context of organizational design and engineering [see e.g. 12, 13]. The apparent benefit of inductive design is that yet 'unexplained', innovative practices can be generalized and reused without have to take the 'detour' of descriptive theorizing. It is however unclear how inductive design should be structured from a conceptual research process perspective and which components of inductive design can be integrated with evidence-based design or with descriptive research activities.

The question is whether an inductive identification of technologies is compatible with DSR-IS. For strategies that have not been formulated and implemented (Mintzberg and Waters use the term 'designed') deliberately, but instead become evident as "a pattern in a stream of decisions" (i.e. are implied by their implementation only), Mintzberg and Waters [38] coined the attribute 'emergent'. Van Burg et al. [39] apply the distinction between deliberate and emergent design to organized systems:

- *Deliberate Design*: (Descriptive) research findings (E-artifacts) are used to identify design principles (T-artifacts) which are used to construct design solutions (S-artifacts) which are in turn implemented as practices (O-artifacts) which might allow new/better research findings. Van Burg et al. [39] designate this as "a process of contextualization". It corresponds to evidence-based design as characterized above.
- *Emergent Design*: Innovative practices (O-artifacts) are generalized as design solutions (S-artifacts) which allow to infer design principles (T-artifacts) which in turn allow to infer research findings (E-artifacts). Van Burg et al. [39] designate this as "a process of de-contextualization". This process has not yet been positioned in the framework proposed here.

4.1 'Backward' Design Activities

In the light of Van Burg et al.'s [39] proposal, the evidence-based design process proposed in the preceding section is 'deliberate': E⇒T, T⇒S and S⇒O move not only upward in the Chmielewicz pyramid (theory⇒model/method⇒instantiation), but also decrease abstraction so that this process chain is 'a process of contextualization'. Can the proposed four quadrant model also be used as a foundation to illustrate de-contextualization processes in IS-DSR?

- Concrete problem solutions (S) need often be repeatedly revised or extended based on insights from their actual use (or not-use) in the world of operations (O).

An example is the revision of an IS solution to overcome user resistance that results from a not easy-to-use interface or from functional deficits. We designate this process as O⇒S as it connects quadrant O to quadrant S.

- Generic artifacts (T) need also often to be repeatedly revised or extended based on insights from applying them in the world of solutions (S). An example is the revision of a design theory to cover contexts or problem aspects that were not covered before and that become apparent during instantiation. We designate this process as S⇒T as it connects quadrant S to quadrant T.
- Finally, generic explanations (E) need sometimes to be revised or extended based on insights from trying to use them for problem-solving (as kernel theories for designing technologies T). An example is the extension of IS use theories by social networking aspects because observed technology adoption effects in the presence of social networking seems to call for new / amended explanations. We designate this process as T⇒E as it connects quadrant T to quadrant E.

When linked together, O⇒S and S⇒T form the backward (feedback) component of the build cycle, the core cycle of design science research [40]. This is illustrated by Figure 6.

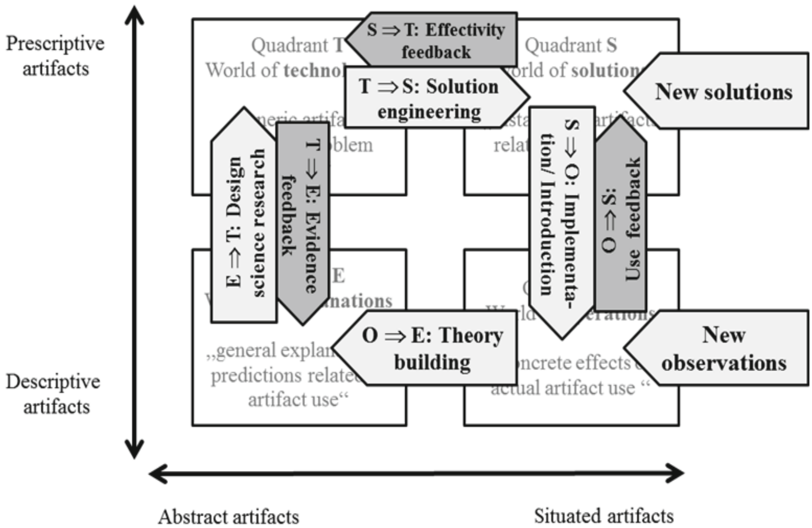

Fig. 6. The ‘build cycle’

The backward / feedback process is however not emergent, but rather a necessary component of deliberate design – and a consequence of the understanding of IS-DSR as a directed search process [34]. The same hold for T⇒E which is also has more the character of a feedback mechanism (of deliberate theory-building) than that of ‘emergent theory-building’.

4.2 Emergent Design

The question is therefore whether really ‘emergent’, de-contextualization processes can be included in the proposed framework. To that end, we characterize inductive design activities in the following:

- Solution induction: By aggregating use data over several time periods, users, use situations, or even organizational sub-structures, solution use data (O-artifacts) can be de-contextualized into solution knowledge (S-artifacts). E. g., configuration options of a concrete IS solution can be inferred by collecting data about what actual functions are used by what types of users in what use situation [41]. We designate this de-contextualization activity as O⇒SE.
- Technology induction: By pattern recognition, classification or techniques like Quantitative Case Analysis [42], technologies (T-artifacts) can be inferred from innovative concrete problem solutions (S-artifacts). Examples are the inductive design of a reference process model from observed ‘best practice’ processes, the inductive design of a maturity model from observed successful capability improvement practices, or the inductive design of a method from observed successful procedures. Depending on the desired degree of de-contextualization, various levels of generality can be realized (see sub-section on artifact generality above and examples in [30]). We designate this de-contextualization activity as S⇒TE.

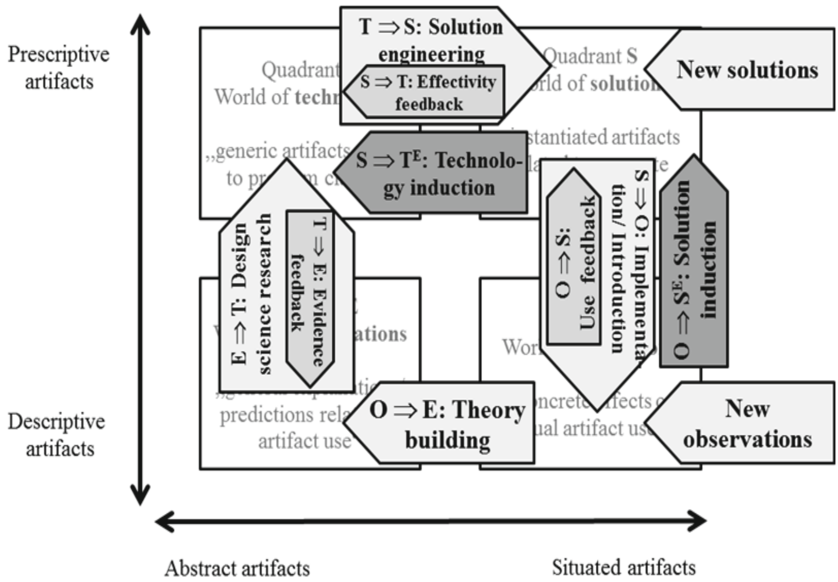

Fig. 7. Evidence-based and inductive design

Fig. 7 adds solution and technology induction to the evidence-based design activities to the already positioned build cycle and theory-building activities. This extension addresses our second research question, the inclusion of inductive design activities.

4.3 Design and Engineering Iterations

The proposed conceptualization of ODE artifacts and processes not only allows representing elementary activities, contextualization, de-contextualization and emergence processes. Furthermore, commonly found activity patterns can be represented as partial cycles:

- Iterations of S⇒O and O⇒S represent ‘instance improvement’: Implementation/use feedback is used to improve a solution without being reflected by enhancing generic technology design.
- Iterations of T⇒S and S⇒T around (S⇒O)(O⇒S) iterations represent ‘theory-agnostic design’: The process is a sequence of build-and-evaluate cycles which are however not explicitly founded on analytical / explanatory / prescriptive models, i.e. do not sufficiently consider (and of course not enhance) the descriptive knowledge base.
- Iterations of E⇒T and T⇒E around (T⇒S)(S⇒T) cycles represent evidence-based IS-DSR because the build-and-evaluate cycles are based on kernel theories and might contribute to their enhancement by “learning and theorizing” [43].

5 Conclusions

Based on the traditional dominance of the quest for describing and explaining the present in many natural as well as social sciences (e.g. physics or sociology), ‘science’ is often regarded as a synonym for descriptive research. For many other scientific disciplines (e.g. medicine, engineering, or architecture), the dominant quest is not understanding or explaining the present world, but changing the current world into a better or preferred one. Nevertheless, this quest is often not designated as ‘science’, but instead as a complementary concept ‘design’ [e.g. 44] or ‘engineering’ [e.g. 45]. The differentiation of ‘science = understanding / explaining / predicting’ on the one hand, and ‘design / engineering = creating / innovating / problem solving’ on the other, however, might imply a qualitative differentiation between ‘research’ activities on the one hand, and ‘consulting’, ‘clinical’ or ‘application’ activities on the other. In this regard, Simon’s seminal work on the sciences of the artificial [34] was a much needed recognition of design and engineering as a scientific activity [e.g. 46]. As long as design is understood as a primarily utility-driven, not necessarily theory-based solution search [like e.g. in 16, 34], however, doubts on its scientific nature will persist. Design and engineering therefore have to develop from ‘experience-based’ into ‘evidence-based’ activities, i. e. need to be founded on the available body of theory and technology knowledge [46]. This requirement was the starting point for our proposal of an integrative framework for descriptive and ‘designed’ (or ‘engineered’) research artifacts.

While the core product of descriptive research is a (generalized) descriptive model, the core product of DSR is a “well-tested solution concept, i.e. a generic intervention to solve a generic field problem, tested in the laboratory and in the field of its intended use.” [46] These artifacts are both abstract, i.e. apply to a large number of individual observations or solutions, respectively. If we add (individual) observations as empirical base of theory building and (individual) solutions as instantiations of generalized interventions (= technologies) to the generic artifacts, we yield the core objects of the proposed four world quadrant model. Such a framework not only helps to better characterize the nature, and identify synergies, between research activities, but also to organize the vast theory and technology knowledge base of organizational design and engineering in a way that supports reuse and integration better. An application for IS-DSR can found e.g. in [47].

Since innovation is not always based on descriptive theory and technology advancements (as well as proper solution engineering and implementation), but can also be enabled by exogenous solution innovation or observed novel practices, inductive research activities need to be integrated with traditional, evidence-based activities. We therefore included not only forward-engineering and feedback activities, but also induction activities into our analysis.

In addition to improving the systematization, access and reuse of knowledge on observations, theories, technologies and solutions in IS-DSR, we see the following challenges that require further research:

- *Understanding Abstraction*: While the semantic boundary between descriptive artifacts (related to empirical facts) and prescriptive artifacts (related to goals and contexts) clearly structures the vertical dimension of the framework, the horizontal dimension is complex, even within a single problem domain. Both the design of generalized solutions as well as the classification / abstraction of concrete practices / operations / decisions rely on a clear and common understanding of abstraction levels, construct clustering, configuration rules, etc. Compared to extreme artifact situation (e.g., cases) and to extreme artifact abstraction ("one size fits all" concepts), this sub-field of IS-DSR appears to be underrepresented and needs more attention.
- *Understanding Use and Context:* Theoretically, the grounding of means-end relations on appropriate cause-effect relations is straightforward: if the ends correspond to a desired effect, then the means is to realize the cause. In real-life design and engineering, however, 'realizing' or 'implementing' causes or linking design goals to theoretical effects is not so straightforward, in particular if use and context are important factors to consider or if explanatory factors and design factors differ. While in organizational sciences it has been proposed to extend 'technological rules' by including context and intervention [48], in IS-DSR multi-grounding [19] or the use of testable design product / process hypotheses [5] has been advocated. A thorough conceptual analysis would certainly help to bring more light into this issue. A unified model of (organizational) context would be a good starting point.
- *Different Disciplinary Culture:* A better integration of design-oriented research with descriptive research requires not only a common framework (that provides common object, dependency and activity definitions, maybe even a common understanding of context and use), but also some compatibility of competencies and disciplinary culture. The boundary between validity (as primary research goal in the descriptive realm) and utility (as primary research goal in the design realm) has fundamental consequences e. g. for evaluation processes, the role perception of researchers, or the closeness of the respective research community to practice. With a

common foundation in place, organizing research knowledge accordingly might be a starting point. The mindset barriers between understanding the world and creating a better world will however always limit the synergy potentials between these research communities.

- *Systematic Discovery:* Finally, this proposal puts emphasis on the concept of de-contextualization and inductive design. Discovery-oriented activities provide an additional path (complementary to the evidence-based path) from detecting innovative practices to better solutions and better technologies / designs. Inductive design has only rarely addressed so far. We believe that a combination of inductive and deductive design activities has a great potential because innovations are often not driven by academia, but by corporate decision makers, solution vendors and consultants in the real world, so that systematic discovery mechanisms would definitely support research that is not only rigorous, but also relevant. This would however require paying more academic attention to solution induction and technology induction.

References

1. Winter, R.: A Framework for Evidence-based and Inductive Design. In: Magalhaes, R. (ed.) Organization Design and Engineering: Co-Existence, Cooperation or Integration? (to appear 2014)
2. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30(3), 611–642 (2006)
3. Bhattacherjee, A.: Understanding Information Systems Continuance: An Expectation-Confirmation Model. MIS Quarterly 25(3), 351–370 (2001)
4. Baskerville, R.L., Pries-Heje, J.: Explanatory Design Theory. Business & Information Systems Engineering 2(5), 271–282 (2010)
5. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3(1), 36–59 (1992)
6. Winter, R.: Design Science Research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
7. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.: Memorandum on design-oriented information systems research. European Journal of Information Systems 20(1), 7–10 (2011)
8. Junglas, I., Niehaves, B., Spiekermann, S., Stahl, B.C., Weitzel, T., Winter, R., Baskerville, R.L.: The inflation of academic intellectual capital: the case for design science research in Europe. European Journal of Information Systems 20(1), 1–6 (2011)
9. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)
10. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.: A Pattern Language. Oxford University Press (1977)
11. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
12. Mintzberg, H.: Patterns in strategy formation. Management Science 24(9), 934–948 (1978)
13. Mettler, T., Eurich, M.: What is the business model behind e-health? A pattern-based approach to sustainable profit. In: Proceedings of 20th European Conference on Information Systems, Barcelona, Spain (2012)

14. March, S.T., Smith, G.F.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)
15. Vahidov, R.: Design Researcher's IS Artifact - A Representational Framework. In: Proc. DESRIST 2006, Claremont, pp. 19–33 (2006)
16. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105 (2004)
17. Chmielewicz, K.: Forschungskonzeptionen der Wirtschaftswissenschaften, 3 ed, 3rd edn. Poeschel, Stuttgart (1994)
18. Gericke, A.: Konstruktionsforschung und Artefaktkonstruktion in der gestaltungsorientierten Wirtschaftsinformatik: Ein Literaturüberblick, Research Report, Institute of Information Management, University of St. Gallen (2008)
19. Goldkuhl, G.: Design Theories in Information Systems - A Need for Multi-Grounding. Journal of Information Technology Theory and Application 6(2), 59–72 (2004)
20. Venable, J.R.: The Role of Theory and Theorising in Design Science Research. In: Proceedings DESRIST 2006, Claremont, pp. 1–18 (2006)
21. Winter, R., Gericke, A., Bucher, T.: Method Versus Model – Two Sides of the Same Coin? In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO! 2009. LNBIP, vol. 34, pp. 1–15. Springer, Heidelberg (2009)
22. van de Weerd, I., de Weerd, S., Brinkkemper, S.: Developing a Reference Method for Game Production by Method Comparison. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences. IFIP, vol. 244, pp. 313–327. Springer, Heidelberg (2007)
23. Kerlinger, F.N.: Foundations of Behavioral Research; Educational and Psychological Inquiry. Holt Rinehart and Winston, New York (1964)
24. Iivari, J.: A Paradigmatic Analysis of Information Systems As a Design Science. Scandinavian Journal of Information Systems 19(2) (2007)
25. Gehlert, A., Schermann, M., Pohl, K., Krcmar, H.: Towards a research method for theory driven design research. In: Proceedings of Wirtschaftsinformatik, pp. 441–450 (2009)
26. Smith, J.M., Smith, D.C.P.: Database abstractions: aggregation and generalization. ACM Transactions on Database Systems (TODS) 2(2), 105–133 (1977)
27. Malone, T.W., Crowston, K., Lee, J., Pentland, B.T., Dellarocas, C., Wyner, G.M., Quimby, J., Osborn, C.S., Bernstein, A., Herman, G.A., et al.: Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. Management Science 45(3), 425–443 (1999)
28. Bucher, T., Winter, R.: Project Types of Business Process Management – Towards a Scenario Structure to Enable Situational Method Engineering for Business Process Management. Business Process Management Journal 15(4), 548–568 (2009)
29. Bucher, T., Winter, R.: Taxonomy of Business Process Management Approaches: An Empirical Foundation for the Engineering of Situational Methods to Support BPM. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management, vol. 2, pp. 93–114. Springer, Heidelberg (2010)
30. Winter, R.: Construction of Situational Information Systems Management Methods. International Journal of Information System Modeling and Design 3(4), 67–85 (2012)
31. Winter, R.: Problem Analysis for Situational Artefact Construction in Information Systems in Andrea Carugati and Cecilia Rossignoli, ed. In: Emerging Themes in Information Systems and Organization Studies, pp. 97–113. Physica, Heidelberg (2011)
32. Sein, M., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action Design Research. MIS Quarterly 35(1), 37–56 (2011)

33. Rousseau, D.M.: The Oxford Handbook of Evidence-based Management. Oxford University Press, New York (2012)
34. Simon, H.A.: The Sciences of the Artificial, vol. 3. MIT Press, Cambridge (1996)
35. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
36. van Aken, J.E., Nagel, A.P.: Organising and managing the fuzzy front end of new product development, Research Report, Technical University Eindhoven (2004)
37. Davies, H.: Improving the Relevance of Management Research: Evidence-Based Management: Design Science or Both? Business Leadership Review (July 2006)
38. Mintzberg, H., Waters, J.A.: Of strategies, deliberate and emergent. Strategic Management Journal 6(3), 257–272 (1985)
39. Van Burg, E., Romme, A.G.L., Reymen, I.M.M.J., Gilsing, V.A.: Creating University Spinoffs: A Science-Based Design Perspective. Journal of Product Innovation Management 25(2), 114–128 (2008)
40. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19(2), 87–92 (2007)
41. Mayer, J.H., Bischoff, S., Winter, R., Weitzel, T.: Extending Traditional EIS Use to Support Mobile Executives Online and Offline. MIS Quarterly Executive 22(2), 87–96 (2012)
42. Ragin, C.C.: Set Relations in Social Research: Evaluating Their Consistency and Coverage. Political Analysis, 291–310 (2006)
43. Rossi, M., Sein, M.K.: Design Research Workshop: A Proactive Research Approach. In: Proc. IRIS 2003. IRIS Association (2003)
44. Romme, A.G.L.: Making a Differnce: Organization as Design. Organization Science 14(5), 558–573 (2003)
45. Magalhaes, R., Silva, A.R.: Organizational Design and Engineering (ODE), Research Report, Technical University of Lisbon (2009)
46. van Aken, J.E., Romme, A.G.L.: A Design Science Approach to Evidence-based Management in Denise. In: Rousseau, M. (ed.) The Oxford Handbook of Evidence-based Management, Oxford University Press, New York (2012)
47. Winter, R., Albani, A.: Restructuring the Design Science Research Knowledge Base - A One-Cycle View of Design Science Research and its Consequences for Understanding Organizational Design Problems in Richard Baskerville. In: Baskerville, R., de Marco, M., Spagnoletti, P. (eds.) Designing Organizational Systems: An Interdisciplinary Discourse, pp. 63–81. Springer, Heidelberg (2013)
48. Denyer, D., Tranfield, D., van Aken, J.E.: Developing Design Propositions through Research Synthesis. Organization Studies 29, 393–413 (2008)