Artifact Types in Information Systems Design Science – A Literature Review

Philipp Offermann, Sören Blom, Marten Schönherr, and Udo Bub

Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{philipp.offermann,soeren.blom,marten.schoenherr,
udo.bub}@telekom.de

Abstract. Many information systems researchers designate their work as design science referring to the term “IT artifact” and the categorization systems that have emerged under this label. Alas, there is no consensus at this point as to what the research output in design science is and what types of artifacts exist. Using a widely accepted artifact typology would strengthen the scientific discussion and ease the categorization of contributions. Based on a literature review of all DESRIST publications and a special MISQ issue on design science, we derived such a typology. We identified eight relevant artifact types and related our typology to existing ones. With this contribution, we hope to enable a discussion about what legitimate design science outputs and their main types are.

Keywords: Design science, research output, IT artifact, typology, literature review.

1 Introduction

Design science has been established as a research paradigm in information systems for many years. According to the current consensus, design science is about designing IT artifacts [1, 2]. However, not even in the superordinate discipline of information systems research (ISR) has a common understanding of the IT artifact evolved. Current discussions in design science do not provide any grounding in existing publications, but are derived from analogies with natural or social science. Additionally, different views on the IT design artifact exist, making it difficult for the researcher to know which view to follow when classifying their own design research.

The purpose of this paper is to identify which types of IT design artifacts exist. Thereby, we hope to create a more uniform understanding, facilitating design output comparison. Additionally, we aim to reduce confusion about what a design output is and what it is not. To achieve this, we performed a literature review of design science publications. As literature, we selected the 2006-2009 DESRIST publications and the 2008 MIS Quarterly vol. 32 no. 4 special issue on design science, and qualitatively analyzed the statements about the design artifacts in those articles. The goal of the analysis was to identify artifact types that can be found in existing publications.

The paper is structured as follows: First, the current understanding of the IT artifact in design science is introduced. Then, our literature review and the design outputs found are presented. Following, results of the qualitative analysis are given and implications discussed. Finally, a conclusion is drawn.

R. Winter, J.L. Zhao, and S. Aier (Eds.): DESRIST 2010, LNCS 6105, pp. 77–92, 2010.

2 Related Work and Background

The discussion about the object of research in information systems has been going on for some time [3-10], as summarized by Benbunan-Fich and Mohan [11] and Alter [12]. The sub-discipline of design science has developed more recently and accordingly offers less existing work on its research output. The link between information systems as behavioral science and design science is the overlap regarding their research objects. While related, the implications of the discussion about the "IT artifact" differ for the two approaches: Behavioral research usually produces theories about information systems, with information systems, or more general "IT artifacts" being the object of research. On the other hand, the output of design science are not theories as known from behavioral research but predominantly designs, so that the research output oftentimes is the same as the research object. This recursive relationship is discussed in other design disciplines as well, often with reference to Gadamer's "hermeneutical circle" [13].

For design science, Vaishnavi and Kuechler Jr. [14] identify the two perspectives on the research output from March and Smith [2] and from Purao [15]. March and Smith [2] state: "Design science products are of four types, constructs, models, methods, and implementations." For them, constructs or concepts are conceptualizations that "form the vocabulary of a domain". They see a model as "a set of propositions or statements expressing relationships among constructs". It represents "situations as problem and solution statements". Methods are based on constructs and models. They are "a set of steps (an algorithm or guideline) used to perform a task." Finally, an instantiation realizes "an artifact in its environment".

Purao [15], following Gregg et al. [16], has a perspective different from March and Smith. In his viewpoint, the most visible output of design research is "the situated implementation of an invention (artifact) as software or system". The implementation ensures that the design is feasible. However, for Purao, "two other kinds of outputs are more important for design research". The first are operational principles or reproducible knowledge. They are abstractions to explicate the intended behavior in accepted forms. The second kind of output is the "metaphorical understanding of how the artifact supports or controls the phenomenon of interest". Thereby, "the expected behavior of the phenomenon […] is articulated".

In addition to these two perspectives, Winter [17] and Bucher and Winter [18] extend the classification of March and Smith [2] by *theory*. Winter [17] argues: "Although theory building is not design science research, theories as 'intermediate' artifacts need to be included in the system of relevant artifacts for IS design science research."

Walls et al. [19] have a different view on theory. They propose "Information System Design Theory" (ISDT) to present research outputs of design science. An ISDT describes a *design product*, consisting of *meta-requirements* and a *meta-design*, and a *design process*. It combines different types of the categorization of March and Smith and includes additional information about the design.

Looking into further publications, Vahidov [20] proposes an output classification matrix that is aligned with the Zachman framework. As categories, he proposes *structure*, *behavior*, *motivation* and *instantiation*. As perspective, he proposes *analytical*, *synthetic*, *technological* and *implementation*. Any design is categorized along both dimensions to determine which cell of the matrix it fits in.

Carlsson [21] follows van Aken [22] in distinguishing three types of designs: “1) an object-design, which is the design of the IS intervention (initiative), 2) a realization-design, which is the plan for the implementation of the IS intervention (initiative), and 3) a process-design, which is the professional’s own plan for the problem solving cycle and includes the methods and techniques to be used to design the solution (the IS intervention) to the problem.”

In summary, some statements on the design science research output exist, but most are not grounded. We have the impression that many authors reference March and Smith or some other classification building on March and Smith in their design science papers. Few authors reflect on the quality of the classification used. A discussion has not yet taken place. Especially, a common view on the use of the categorization systems has not yet emerged.

3 Literature Review Approach

To help remedy the aforementioned situation, we performed a literature review to identify types of design science research output using a literature review.

We loosely follow a qualitative literature review methodology. Our methodology is based on the recommendations from [23], the methodologies described in [24] and the example paper [25]. Effectively, we applied a mix of coding and mutual agreement. Table 1 gives an overview of this approach. The remainder of this section describes each step in more detail.

Table 1. Sequence of steps to derive this article’s typology

Activity	Description	Results
Data set definition	Determine the set of articles to review	Set of 106 articles
Filtering of design articles	Identify the articles that contribute a design that is relevant for practitioners	Subset of 62 articles
Extraction of authors’ categories	Identify if and how authors subsumed their output under any type or category of design	List of terms describing design categories
Determining individual typologies	Each researcher: abstract the identified terms into typology, grouped by similarity	Three individual typologies
Unified typologies	Unify the three individual groupings into one agreed-upon typology	Typology draft
Individual classification of articles	Each researcher: classify all design articles using the typology draft	Three classifications of design articles
Consolidation of classification	Discuss differences in classification and determine if differences affect typology	Unified classification of articles, final typology

3.1 Data Set Definition and Filtering of Design Articles

The data set contains 106 papers; 102 papers from DESRIST 2006-2009 conferences and 4 papers from the MISQ vol. 32 no. 4 special issue. Arguably, the data set is far from comprehensive. This might lead to an incomplete typology. At the same time, the selected publications had an explicit focus on design science and arguably offered a better look at the complete bandwidth of design science in ISR than publications with an otherwise specialized focus. As we are not interested in quantitative statements and consider our study to be exploratory, we feel that this data set is of sufficient size and quality.

The research authors of this paper were involved in the literature review and interpretation; the research was conducted between December 2nd, 2009 and January 4th, 2010. By using publications that are specialized on design science as our data set, we expect to cover the majority of IT artifact types designed in actual research projects.

To determine which articles were of further interest, we had to decide which papers to classify as design science and include in the review and which papers to exclude. We decided to include papers presenting prescription-driven design science according to van Aken [22] and papers prescribing design and action according to Gregor [26]. Based on these two papers, we developed two questions to classify publications: *Who uses the results?*, with *practitioners* being in scope and *researchers only* being out of scope; and *How are the results used?* with *to guide action* being in scope and *to understand the world, to inform only* being out of scope.

Each of the researchers looked at each of the 106 papers as to whether it contained a designed artifact, using the two questions presented above. In 38 cases we independently classified a paper as not presenting a designed artifact. In 12 cases a consensus could be reached after a discussion. After the discussion, 62 of the 106 papers remained that we considered presenting a design within the realms of information system design science.

3.2 Extraction of Author's Categories

For the remaining papers, we extracted statements from the papers about what the proposed design is. Additionally, we extracted what type of design artifact the paper's authors classified their design as. This step did not involve any interpretation of the terms found; they were taken from the papers unchanged. A summary of types found in the 62 papers can be seen in table 2.

3.3 Individual Categories

We continued by interpreting the information found. Our aim was to identify a set of types that would be coherent and classify all artifacts found. Hence, the next step was for each researcher to look at the terms extracted from the 62 papers to come up with his own set of types. The guideline was to identify types of design research outputs, respectively IT artifacts that are structurally different, requiring different description meta-models. We believe that types, differentiated along structure, are important because they require different descriptions, evaluation techniques and research methodologies. To expose the process of our research, the types found by each of the researchers are shown in table 3.

Table 2. Extraction of authors' categories (multiple terms per paper possible)

Category	Synonymous usage	Paper
Algorithm		[27]; [28]; [29]
Approach		[30]; [31]
Architectural style		[32]
Architecture	Tool architecture	[33]; [34]; [35]
Concept	Modeling concept	[36]; [33]; [37]
Construct		[36]
Design artifact		[38]
Design guidelines		[39]
Design implications		[40]
Evaluation		[41]; [42]; [43]; [40]; [44]; [45]; [46]
Framework	Process Framework	[41]; [47]; [48]; [39]; [49]; [50]
Grammar	Modeling grammar	[51]; [37]
Graphical representation		[35]
Information System		[52]; [53]
Instantiation		[39]; [54]; [55]
IT-artifact		[33]; [56]; [57]
Meta-Design		[50]
Method	Method fragments, situational Method	[58]; [51]; [59]; [47]; [52]; [45]; [54]; [60]; [61]; [62]; [37]
Methodology	Modeling methodology	[30]; [47]; [63]
Metric		[42];
Model	Model prototype, design model	[64]; [65]; [66]; [31]; [49]; [46]; [55]; [53]
Principles	Design principles	[43]; [48]
Process		[30]
Protocol	Protocol extension	[32]; [67]
Prototype	Software Prototype	[51]; [66]; [43]; [40]; [68]; [69]
Recommendation		[44]
Rule		[70]
Simulation platform		[71]
System artifact		[72]
System design		[73]; [48]; [74]; [75]; [72]; [76]; [77]
Theory	Design Theory, Theory nexus	[78]; [79]; [65]; [41]; [44]; [39]; [80]; [81]; [54]; [57]
Tool		[33]
Typology		[36]
No term matching	*The paper delivers a designed artifact but it is not explicitly subsumed under any term*	[82]; [83]; [84]

3.4 Unified Typology

Looking at the number of types found in the previous step, it is obvious that the granularity varies. To derive an agreed-upon typology, we discussed every proposed type. For some types, we noticed that we used different names for the same thing (e.g. *technology method* vs. *algorithm*). In these cases, we decided to agree on one of the names used. In some cases we used a different granularity, for example, one typology contained one type where another differentiated several types. In these cases, we had to discuss which types would require similar description structures. Finally, some types were vague and covered a large number of different concepts, all of which were structurally similar. In those cases we introduced an auxiliary criterion *use* (as in *how is the artifact used with regard to a system?*), which helped to uncover possible differences between subtypes and allowed to map them to existing types or to introduce new narrower types.

Table 3. Author's personal typologies

Author A	**Author B**	**Author C**
Method	Construct	System Design
- Modeling Language	Model	Method
- Methodology	- Cost model	- Situational Method
- Evaluation	- Business model	- Guideline
Software	Requirements	- Design Method
- Prototype	Software architecture	- Business Method
- Algorithm	Instantiation	- Evaluation Method
- System	Method	- Learning Method
Model	- Technology Method	(Evaluation) Framework
- Algorithm	- Business Method	Instantiation
- Data Model	- Modeling approach	- Prototype
- Metric	IS curriculum	- Tool
- Generalized	Machine	(Modeling) Language
Description		Design Theory
- Principle		- For Method
- Architectural		- For System
Style		- Meta-Design
- Pattern		Algorithm/Protocol
- Guideline		Model
- Typology		- Cost Model
- Taxonomy		- Concept
- Ontology		- Ontological Design
		- Data Model
		- Typology
		- Architecture
		- Metric
		- Strategy
		Curriculum
		Codified Experience
		- Design Principles
		- Principles
		- Architectural Style
		- Pattern

Furthermore, we excluded two types that arguably are designs: for *implementation* we agreed that it is used to evaluate some more general design proposition. We were able to reclassify all papers stating to present an implementation as *system design*. For *IS curriculum* we agreed that it is not considered an IS research output as it is not related to an information system and to ignore the type for the further proceeding.

After analysis of term the individual classifications, we agreed upon eight types, listed in table 4.

Table 4. Artifact typology by number of occurrences (multiple types/paper permitted)

Artifact Type	Use	Structure	Paper
System design	Description	Structure or behavior-related description of a system, commonly using some formalism (e.g. UML) and possibly text	[73]; [48]; [74]; [75]; [72]; [82]; [76]; [77]; [56]; [79]; [33]; [78]; [31]; [34]; [66]; [40]; [52]; [64]; [43]; [68]; [53]; [29]; [71]; [81]; [69]; [51]
Method	Support	Definition of activities to create or interact with a system	[58]; [51]; [59]; [47]; [52]; [45]; [54]; [61]; [62]; [37]; [83]; [44]; [30]; [39]; [41]; [37]; [57]; [63]; [83]
Language/ Notation	Support	A (generally formalized) system to formulate statements that represents parts of reality	[65]; [37]; [63]; [55]; [35]; [51]
Algorithm	Description	Executable description of system behavior	[27]; [28]; [29]; [67]; [61]; [32]
Guideline	Support	Suggestion regarding behavior in a particular situation (if in situation X do Y)	[36]; [39]; [43]; [70]; [84]
Requirements	Description	Statement about System (A system of type X should have some property Y [because of Z])	[80]; [49]; [81]
Pattern	Support	Definition of reusable elements of design with its benefits and application context	[63]; [32]
Metric	Support	A mathematical model that is able to measure aspects of systems or methods	[42]
Re-classified as 'no design'		*After reading the whole paper the contribution was re-classified as not delivering/addressing a design artifact.*	[60]; [38]; [46]; [50];

In our typology, we define *system design* to be a description of an IT-related system. The description can be on any granularity level and can focus on any aspect like structure, process, and interactions. Examples are “software architecture”, “enterprise architecture”, “database schema” and “business process diagram”.

For us *requirements* are statements about a system in the form: “A system of type X should have some property Y.” Optionally a reason for the requirement can be given. Generally, requirements restrict the design space for a system design. Examples are: “A life-supporting system should never fail.” “A multi-tenancy system should never expose private data to other tenants.”

In our definition a *method* consists of activities, possibly in some order, that are performed by people in order to support the system development. Methods often define results/deliverables of activities and roles. Examples are “software engineering method”, “enterprise architecture method”, “requirements analysis method”, and “organizational change method”.

An *algorithm* is in some kind similar to a method in that it describes a sequence of activities. However, an algorithm is executed by a computer. In our definition, it is an executable description of a system behavior. Examples are “sorting algorithm”, “data mining algorithm”, and “protocol”.

A *pattern* provides generalized system design elements that can be used for many different kinds of system designs. In that sense, it is used not to describe how a specific system should look like. Rather, a pattern provides support to create such a system design. Patterns exist for programming, software architecture, enterprise architecture, organizational design etc. Usually, the pattern is described with its benefits and the context of application. Examples are “singleton”, “asynchronous message queue”, “service-oriented architecture”, and “matrix organization”.

A *guideline* provides a generalized suggestion about system development. In that sense it is similar to a pattern, but does not contain system design elements or statements about these elements. It does not have a fixed structure, but will usually make statements like: “In situation X one could/should do Y.” Examples are “If high code quality is required, pair programming should be used.” “If a project is late, don’t add more people.”

A *language / notation* provides concepts and their interrelation, used to support system development. A graphical notation might include modeling elements and rules how these elements can be related. Often, a language / notation is referenced as a result type in a method. Examples are “entity-relationship model”, “business process modeling notation”, and “object-oriented programming language”.

Finally, in our typology, we define *metric* as some kind of model that is used to evaluate aspects of a system design or a support for a system design. A metric will provide a conclusion about the evaluated construct. Usually, a metric is a mathematical model, but qualitative metrics are also possible. Examples for metrics are “business case”, “architecture evaluation model” and “cyclomatic complexity”.

3.5 Classification of Papers by the Typology

To verify how uniform our understanding of the eight types is and to ensure that we did not miss a type, each researcher reclassified each of the 62 papers using the unified typology. For this, only the extracted information from the first iteration and the papers' abstracts could be used. For 31 papers, all researchers used the same types. Two of these papers contained an IS curriculum, which we did not include in our final typology as discussed above. For 12 papers, a consensus could be reached after a short discussion. One further paper was classified as IS curriculum. We had to check the remaining 19 papers for two possibilities: whether the classification was ambiguous or impossible because the design was not clearly described in the excerpts reviewed by us. Or if they were an instance of a type that might still be missing from our typology. We re-read all papers in more detail and then discussed our individual findings. For five papers, we concluded that the paper actually did not contain any design. For the other 14 papers, a type could be identified after continued reading. We did not come across any papers that required adding a completely new type. Therefore, we believe that the typology is complete with respect to the data set used.

4 Discussion

The two major aspects of this article are to get a better understanding about what a design artifact in design science is and what types of artifacts can be separated. At this point we want to reflect on the results presented in the previous section, addressing the approach, the types identified and the relation to the existing artifact categorization systems.

4.1 Review of Our Approach

In contrast to other design artifact categorization systems (cf. section 2), we chose to derive the artifact types from existing literature. The reflection of our approach must therefore focus on the quality of the data and our choice regarding its interpretation.

We noted that within the DESRIST publication the heterogeneity of notions about "design" and "design science" converged, while the quality with which contributions and mapping to a particular artifact type increased. This is not surprising for a young discipline and a young conference, but highlights the fact that the notion of what design is and what design artifacts are is not yet stable. The possibility that our typology is affected by such a shift in notions is possible but not a problem, as our aim is to facilitate discussion and the narrowing of understanding about artifacts and not to settle it.

As described above, we analyzed the articles on two levels: the first level included the title, the abstract and the keywords; the second level was to look at the whole article and read as much as necessary. In principle it might be that the understanding of what an article is about shifts, depending on how much one reads. This could affect both the decision about whether a design is presented as well as the categorization of the artifact. This risk is mitigated by several factors: we do not depend on quantitative data, so the possibility that the type of an artifact changes does not change our results. Also, we are focused on what the authors state their artifact is, as we want to find out what researchers consider artifacts to be. If an author states clearly in the abstract that, for example, a method is constructed, we considered this to be the relevant piece of information. We wanted to analyze what authors claim their design is; we did not want to determine what the authors have actually designed. Only if the first level of analysis did not contain such a clear categorization by the author we did consider the full text, which further reduced the number of possible occurrences.

We attempted to separate different types by their internal structure; that is, by the constituting elements and their relations. This separation took place based on the terms in table 3, which describes the researchers' individual interpretation of possible type candidates. The structures were not derived directly from the structures inherent in the reviewed articles but were introduced as needed, when a decision had to be made whether to keep two terms as separate types or to merge them into one. One might argue that this approach facilitates ad-hoc decisions. Yet, looking at our results, we believe that the types we identified are justifiably different from each other and are defined in a way that is comprehensible by other researchers and practitioners in the field.

4.2 Review of Our Results

The eight final types are introduced in detail in section 3.4. Here, we want to discuss some of the backgrounds and intermediate candidates.

Several of the reviewed articles used the term *model* to describe their contributed artifact (cf. table 2), as did all researchers in this paper (cf. table 3) in their intermediate grouping of terms. Consequently, *model* was a candidate artifact type for a while, but was excluded in the end. We found it to be too generic and in any particular instance shared too many characteristics with other types. Design models are covered by system design; mathematical models (at least in our sample) could be all subsumed under metrics. Other models were more aptly described as languages or notations. The possibility remains that in a larger data set we would find another type of model. In such a case, it would most likely still be possible to introduce specialized types of artifacts instead of returning to a general type *model*.

In this and other discussions the dimension of *use* was helpful. We identified three different uses that the IT design research outputs had:

- Description: How should a system look like?
- Support: How do I create a system design and the IT system?
- Evaluation: What properties does a system have?

While we succeeded to refer to structure as our only criterion to differentiate the types, *use* was very helpful in getting a first and general understanding why two types should or should not be differentiated. With a better understanding how two types are different, it was then possible to also identify their structural differences.

Another term highly debated within the community is *design theory* (e.g. [1, 19, 85]). Several authors of reviewed articles explicitly referenced *design theory* or *theory* with focus on design (cf. table 2). We identified two views on what such a theory would be. One view considered a design theory to be a design augmented by theoretical background and proposition about how instances of the design would behave in real life. We extracted such a design and categorized it under the appropriate type. A second view considered design theories to be very high level and generalized statements about design, which does not necessarily contain any design fragments itself. This view does not contain any design and was excluded from the review.

The eight identified types are somewhat related, as to be expected. *Guideline* and *pattern* have similarities – a pattern is a specialized form of guideline, as well as *guideline* and *requirement*, as they both contain descriptions of goals. Also, the level of abstraction is different: *system design* subsumes low level system designs as well as architecture descriptions and business processes, whereas *algorithm* and *metric* are more precisely and narrowly defined. Such "unevenness" might be unsatisfying or aesthetically unpleasing for a typology. This is, however, no concern to us. For one, the limited data set in a developing field, we discussed earlier is partly responsible for this. Also, we would evaluate the quality of the typology pragmatically, that is, by looking at the ease with which articles can be classified into one of our types. For us, based on the available data, this typology worked sufficiently well.

4.3 Relation to Larger IT-Artifact Discussion

In section 2 we have discussed other definitions and typologies of artifacts that we could find within the information systems design science literature. When comparing those to our results, several differences become apparent. The literature identifies fewer types of artifacts and they are of a more abstract nature. Our typology shares with the published ones the separation between structural and behavioral artifacts (*model* / *method* in case of March and Smith [2] and *design product* / *design process* of Walls et al. [19]).

More insightful than comparing individual types might be to contrast the different groups of people addressed by the typologies. We argue that previous typologies of the IT artifact had the research community in mind, when deriving their typologies: March and Smith [2] with their the artifact type *construct* describe it as the vocabulary, with which to reason about design. Purao [15] explicitly considers the knowledge *about* an artifact as a relevant output. Similarly, Walls et al. [19] and Winter [17] focus on the design theory as a central output of design research. We agree with these authors in that all these outputs are relevant for design science and that building knowledge *about* designs is a crucial task for design science. Nevertheless, in our view the separation along the lines of who can make use of the artifacts (researcher or practitioner) is relevant and meaningful for the field of design science and that any discussion about types of artifacts should start with the "first order objects", namely the designs that are both object and result of research. The "second order objects", such as theories, representing knowledge about "first order objects" can follow. Any typology of artifacts though, should be grounded in the available literature, if only to have a chance to be established by its own outputs and not as possibly inaccurate analogies to other sciences.

5 Conclusion

The purpose of this paper was to identify types of design research output. By qualitatively analyzing all DESRIST publications and four MISQ papers, we have created a typology containing eight distinct types (see table 4). As we used 62 papers containing design research output to establish the typology, we are confident we covered most types currently relevant for the discipline. To our best knowledge, we are the first to ground such a typology in existing literature. Using the typology, researchers can clarify the object of their research, possibly also facilitating focus and usage of suitable research methodologies. Thereby, confusion of reviewers and readers is reduced and comparability of designs increased; the level of standardization as well as the quality of scientific communication within the design science community would increase.

Based on our grounded typology, a discussion can take place as to whether further types should be the object of design science research. It is possible that there are types that have not yet been presented. Reasons might be that such types are difficult to design, difficult to evaluate or even that a type has not yet been relevant. A periodical evaluation of "artifacts in use" can supply valuable insight for the ongoing discourse about the elements and boundaries of the discipline. A follow-up discussion could also sharpen the understanding of theory in design science research and its relevant artifacts.

References

1. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS QUART 28, 75–105 (2004)
2. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15, 251–266 (1995)
3. Orlikowski, W.J.: Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. Organization Science 11, 404–428 (2000)
4. Orlikowski, W.J., Iacono, C.S.: Research Commentary: Desperately Seeking the "IT" in IT Research—A Call to Theorizing the IT Artifact. Information Systems Research 12, 121–134 (2001)
5. Benbasat, I., Zmud, R.W.: The Identity Crisis Within the IS Discipline: Defining and Communicating the Discipline's Core Proporties. MIS QUART 27, 183–194 (2003)
6. Agarwal, R., Lucas Jr., H.C.: The Information Systems Identity Crisis: Focusing on High-Visibility and High-Impact Research. MIS QUART 29, 381–398 (2005)
7. Alter, S.: Sidestepping the it artifact, scrapping the is silo, and laying claim to "systems in organizations". Communications of AIS 12, 494–526 (2002)
8. Alter, S.: 18 reasons why it-reliant work systems should replace "the it artifact" as the core subject matter of the is field. Communications of AIS 12, 365–394 (2003)
9. Alter, S.: The is core – xi sorting out issues about the core, scope, and identity of the is field. Communications of AIS 12, 607–628 (2003)
10. Davis, G.B.: Information Systems Conceptual Foundations: Looking Backward and Forward. In: IFIP TC9 WG9.3 International Conference on Home Oriented Informatics and Telematics, pp. 61–82. Kluwer, B.V., Deventer (2000)
11. Benbunan-Fich, R., Mohan, K.: Information technology and systems - iii research publications in systems development during 2000-2004. Communications of AIS 17, 2–33 (2006)
12. Alter, S.: Work systems and it artifacts – does the definition matter? Communications of AIS 17 (2006)
13. Snodgrass, A., Coyne, R.: Is Designing Hermeneutical? Architectural Theory Review. Journal of the Department of Architecture 1, 65–97 (1997)
14. Vaishnavi, V.K., Kuechler Jr., W.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach (2007)

15. Purao, S.: Design Research in the Technology of Information Systems: Truth or Dare. Information Systems, Atlanta (2002)
16. Gregg, D.G., Kulkarni, U.R., Vinzé, A.S.: Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. Information Systems Frontiers 3, 169–183 (2001)
17. Winter, R.: Design science research in Europe. European Journal of Information Systems 17, 470–475 (2008)
18. Bucher, T., Winter, R.: Dissemination and Importance of the "Method" Artifact in the Context of Design Research for Information Systems. In: DESRIST 2008, Atlanta, Georgia (2008)
19. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. Information Systems Research 3, 36–59 (1992)
20. Vahidov, R.: Design Researcher's IS Artifact: a Representational Framework. In: DESRIST 2006, CGU, Claremont, CA (2006)
21. Carlsson, S.A.: Towards an Information Systems Design Research Framework: A Critical Realist Perspective. In: DESRIST 2006. CGU, Claremont (2006)
22. van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. Journal of Management Studies 41, 219–246 (2004)
23. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS QUART 26, xiii–xxiii (2002)
24. Creswell, J.W.: Research Design - Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publications, Thousend Oaks (2009)
25. Stelzer, D.: Enterprise Architecture Principles: Literature Review and Research Directions. In: TEAR 2009, pp. 21–36 (2009)
26. Gregor, S.: The Nature of Theory in Information Systems. MIS QUART. 30, 611–642 (2006)
27. Bryant, W.: Mixer Selection in Peer-to-Peer Ad hoc Collaboration: A Design Science Approach. In: DESRIST 2007, CGU, Pasadena, CA (2007)
28. Li, S., Lee, R., Lang, S.-D.: Mining Distance-based Outliers from Categorical Data. In: DESRIST 2007, CGU, Pasadena, CA (2007)
29. Huang, W., Stohr, E.: Design and Implementation of a Business Process Rules Engine. In: DESRIST 2007, CGU, Pasadena, CA (2007)
30. Carlsson, S.A., Henningsson, S., Hrastinski, S., Keller, C.: An Approach for Designing Management Support Systems: The Design Science Research Process and its Outcomes. In: DESRIST 2009. ACM, Malvern (2009)
31. Li, L., Vashnavi, V.K., Vandenberg, A., Metikurke, S.: Generating Dynamic and Adaptive Knowledge Models for Web-Based Resources. In: DESRIST 2006, CGU, Claremont, CA (2006)
32. Erenkrantz, J., Gorlick, M., Taylor, R.N.: Rethinking Web Services from First Principles. In: DESRIST 2007, CGU, Pasadena, CA (2007)
33. Abhichandani, T., Kosaka, K., Chatterjee, S.: Designing a P2P Ad-hoc Secured Collaboration Tool. In: DESRIST 2006, CGU, Claremont, CA (2006)
34. Nguyen, V., Lee, R., Dutta, K.: An Aspect Architecture for Modeling Organizational Controls in Workflow Systems. In: DESRIST 2006, CGU, Claremont, CA (2006)
35. Woodard, J.: Modeling Architectural Strategy Using Design Structure Networks. In: DESRIST 2007, CGU, Pasadena, CA (2007)
36. Berente, N., Lyytinen, K.: The Iterating Artifact as a Fundamental Construct in Information System Design. In: DESRIST 2006, CGU, Claremont, CA (2006)

37. D'Aubeterre, F., Singh, R., Iyer, L.: Secure Activity Resource Coordination: A Method to Design Secure Business Processes. In: DESRIST 2008, Atlanta, Georgia (2008)
38. Umapathy, K., Purao, S., Barton, R.R.: Designing Enterprise Integration Solutions - Effectively. In: DESRIST 2008, Atlanta, Georgia (2008)
39. Gurzick, D., Lutters, W.G.: Towards a Design Theory for Online Communities. In: DESRIST 2009. ACM, Malvern (2009)
40. Roussinov, D., Chau, M.C.-L.: Web Fact Seeking For Business Intelligence: a Meta Engine Approach. In: DESRIST 2006, CGU, Claremont, CA (2006)
41. Keller, C., Gäre, K., Edenius, M., Lindblad, S.: Designing for Complex Innovations in Health Care: Design Theory and Realist Evaluation Combined. In: DESRIST 2009. ACM, Malvern (2009)
42. Tremblay, M.C., Berndt, D.J., Hevner, A.R.: Measuring Information Volatility in a Health Care Information Supply Chain. In: DESRIST 2009. ACM, Malvern (2009)
43. Vaishnavi, V.K., Vandenberg, A., Zhang, Y., Duraisamy, S.: Towards Design Principles for Effective Context- and Perspective-Based Web Mining. In: DESRIST 2009. ACM, Malvern (2009)
44. Pries-Heje, J., Baskerville, R.: Designing Organizational Change in IT: A Theory Nexus. In: DESRIST 2006, CGU, Claremont, CA (2006)
45. Gericke, A., Fill, H.-G., Karagiannis, D., Winter, R.: Situational Method Engineering for Governance, Risk and Compliance Information Systems. In: DESRIST 2009. ACM, Malvern (2009)
46. van Steenbergen, M., Bos, R., Brinkkemper, S.: The application of a prescriptive and a constructive perspective to design-science research. In: DESRIST 2008, Atlanta, Georgia (2008)
47. Kunene, K.N., Weistroffer, H.R.: Design of a Method to Integrate Knowledge Discovery Techniques with Prior Domain Knowledge for Better Decision Support. In: DESRIST 2006, CGU, Claremont, CA (2006)
48. Chung, W.: Designing Web-based Business Intelligence Systems: A Framework and Case Studies. In: DESRIST 2006, CGU, Claremont, CA (2006)
49. Even, A., Shankaranarayanan, G., Berger, P.D.: Economics-Driven Design of a Data Warehouse. In: DESRIST 2007, CGU, Pasadena, CA (2007)
50. Eden, H., Fischer, G., Giaccardi, E., Ye, Y.: Report on NSF Science of Design Team Project: "A Meta-Design Framework for Participative Software Systems. In: DESRIST 2007, CGU, Pasadena, CA (2007)
51. Lee, J., Wyner, G.M., Pentland, B.T.: Process grammar as a tool for business process design. MIS QUART. 32, 757–778 (2008)
52. An, Y., Dalrymple, P.W., Rogers, M., Gerrity, P., Horkoff, J., Yu, E.: Collaborative Social Modeling for Designing a Patient Wellness Tracking System in a Nurse-Managed Health Care Center. In: DESRIST 2009. ACM, Malvern (2009)
53. Garrett, N., Thoms, B., Soffer, M., Ryan, T.: Extending the Elgg Social Networking System to Enhance the Campus Conversation. In: DESRIST 2007, CGU, Pasadena, CA (2007)
54. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS QUART. 32, 731–755 (2008)
55. Judge, R.: A Design Language for Developing and Simulating Implementation Strategies for Knowledge Management Systems (KMS) in Small to Mid-size Enterprises (SME). In: DESRIST 2007, CGU, Pasadena, CA (2007)

56. Miller, T., Leroy, G., Huang, J., Chuang, S., Charlop-Christy, M.H.: Using a Digital Library of Images for Communication: Comparison of a Card-Based System to PDA Software. In: DESRIST 2006, CGU, Claremont, CA (2006)
57. Zheng, G., Vaishnavi, V., Vandenberg, A.: A Multidimensional and Visual Exploration Approach to Project Prioritization. In: DESRIST 2008, Atlanta, Georgia (2008)
58. Ramaprasad, A., Papagari, S.S.: Ontological Design. In: DESRIST 2009. ACM, Malvern (2009)
59. Liu, X.M., Wyner, G.M.: Coordination Analysis: A Method for Deriving Use Cases from Process Dependencies. In: DESRIST 2009. ACM, Malvern (2009)
60. Johnstone, M.N., Venable, J.R.: Enacting Design Science with Action Research: An Illustrative Case Study in Methodology Development. In: DESRIST 2008, Atlanta, Georgia (2008)
61. Chou, C.-h., Sinha, A.P., Zhao, H.: Designing Web Page Classifiers as Alternatives to Commercial Internet Filters. In: DESRIST 2008, Atlanta, Georgia (2008)
62. Ghaisas, S., Venkatesh, R., Kanhere, A., Kumar, A., Kumar, M., Rose, P.: Identifying unobvious requirements — an exemplar of the Design Science approach. In: DESRIST 2008, Atlanta, Georgia (2008)
63. Athenikos, S.J., Song, I.-Y.: A Framework of Ontology-based Modeling Patterns. In: DESRIST 2008, Atlanta, Georgia (2008)
64. Baloh, P., Desouza, K.C.: Towards knowledge needs-technology fit model for knowledge management systems. In: DESRIST 2009. ACM, Malvern (2009)
65. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: DESRIST 2009. ACM, Malvern (2009)
66. Su, J., Parsons, J.: Analysis of Data Structures to Support the Instance-based Database Model. In: DESRIST 2006, CGU, Claremont, CA (2006)
67. Dutta, A.K., Molle, M., Phatak, T.: Protocol Alchemy: Designing a TCP-Based Approach to Support Large-Scale Deployment of VoIP on the Internet. In: DESRIST 2006, CGU, Claremont, CA (2006)
68. Murphy, J.D., Petter, S., Cheng, K., Briggs, R.O.: Hitting the Collaboration Target: Computer-Guided thinkLet Selection. In: DESRIST 2008, Atlanta, Georgia (2008)
69. Vandenberg, A., Vaishnavi, V.K., Li, L., Pandya, M.: LAIDBACK: A Workbench for User-Driven Dynamic Web Mining. In: DESRIST 2007, CGU, Pasadena, CA (2007)
70. Parsons, J., Wand, Y.: Using Cognitive Principles to Guide Classification in Information Systems Modeling. MIS QUART. 32, 839–868 (2008)
71. Oh, D., Shih-Fen, C., Dan, M., Bapna, R.: Designing an Experimental Gaming Platform for Trading Grid Resources. In: DESRIST 2007, CGU, Pasadena, CA (2007)
72. Vahidov, R., He, X.: Design for Situated Decision Support: Application to Personal Finance. In: DESRIST 2008, Atlanta, Georgia, pp. 1–6 (2008)
73. Reynoso, J.M.G., Olfman, L., Ryan, T., Horan, T.: An Information Systems Design Theory for an Expert System for Training. In: DESRIST 2006, CGU, Claremont, CA (2006)
74. Abbasi, A., Hsinchun, C.: CyberGate: A Design Framework and System for Text Analysis of Computer-Mediated Communication. MIS QUART. 32, 811–837 (2008)
75. Reiss, S.P.: Designing Internet-Based Software. In: DESRIST 2007, CGU, Pasadena, CA (2007)
76. Parsons, J., Ralph, P.: Generating Effective Recommendations by Exploiting Viewing Time and Item Attributes. In: DESRIST 2007, CGU, Pasadena, CA (2007)
77. Etezadi, J.: iMade (Internet Multi-Attribute Decision Expert): A system for analysis of multi-attribute decision problems. In: DESRIST 2008, Atlanta, Georgia (2008)

78. Burkhard, R.J., Horan, T.A.: A Design Theory Approach to Incorporating Emotional Intelligence into Group Support Systems. In: DESRIST 2006, CGU, Claremont, CA (2006)
79. Gregor, S., Jones, D.: The formulation of an Information Systems Design Theory for e-learning. In: DESRIST 2006, CGU, Claremont, CA (2006)
80. Park, Y., El Sawy, O.A.: Towards a Design Theory for Hedonic Systems: Delivering Superior User Experience in the Digital Home Entertainment Context. In: DESRIST 2008, Atlanta, Georgia (2008)
81. Singh, R., D'Aubeterre, F., Iyer, L., Salam, A.F., Ehrhardt, R.: Secure Semantic eBusiness Processes: A Design Theoretic Approach to Secure Semantic Systems. In: DESRIST 2007, CGU, Pasadena, CA (2007)
82. Green, K.E., Gugerty, L.J., Witte, J.C., Walker, I.D., Houayek, H., Rubinstein, J., Daniels, R., Turchi, J., Kwoka, M., Dunlop, I., Johnson, J.: The Animated Work Environment: A Design-Research Exemplar Supporting Social and Virtual Collaboration in an Increasingly Digital Society. In: DESRIST 2008, Atlanta, Georgia (2008)
83. Amoussou, G.-A., Cashman, E., Steinberg, S.: Building an Interdisciplinary Community to Learn and Teach Creativity and Design. In: DESRIST 2007, CGU, Pasadena, CA (2007)
84. Petter, S., Khazanchi, D., Murphy, J.D.: A design science based evaluation framework for patterns. In: DESRIST 2008, Atlanta, Georgia (2008)
85. Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the AIS 8, 312–335 (2007)