

# Just Finished a Cycle of a Design Science Research Project: What's Next?

Ilia Bider<sup>\*</sup>, Erik Perjons, and Paul Johannesson

Department of Computer and Systems Sciences (DSV), Stockholm University,  
164 55 Kista, Sweden

[ilia@dsv.su.se](mailto:ilia@dsv.su.se), [perjons@dsv.su.se](mailto:perjons@dsv.su.se), [pajo@dsv.su.se](mailto:pajo@dsv.su.se)

**Abstract.** The article is devoted to developing a methodological support for planning Design Science (DS) research projects. More exactly, it is aimed at developing a classification of a particular kind of cycles, inherent to DS research projects, and guidelines on how to choose the next cycle in a specific project. The classification and guidelines are based on results from an analysis of DS literature and a reflective analysis of our own research practice. As far as own research practice is concerned, two DS projects have been analyzed in detail. The analysis revealed that though both projects included cycles, the nature of these cycles was different. In the first project, cycles concerned finding a better problem to solve, while in the second project, cycles concerned finding a better solution for more or less the same problem. Both projects concern developing methodologies in the area of socio-technical systems, and the guidelines on how to choose the next cycle have special provisions related to such systems. In conclusion, the article presents examples of other projects that followed the suggested guidelines and discusses the difference of the suggested approach from other approaches to conduct DS research projects.

**Keywords:** Design Science, Cycle, Project Management, Research Project, Socio-technical System.

## 1 Introduction

We start this article with referring to the work of [1]. Written in 2010, it gives a retrospective view on the IS discipline in the previous 25 years and tries to draft its path for the next 25 years. The main conclusion of this article is that there is a discrepancy between what IS is trying to achieve in the real world and in the university world. In the real world, IS is trying to establish itself as a profession based on the science of artificial, in the sense of [2]. In the academic world, however, in many cases, IS is trying to establish itself as a discipline based on the natural science. The difference reveals itself in what kind of theories (models, methods, etc.) are in the

---

<sup>\*</sup> Corresponding author

© 2020 Ilia Bider et al. This is an open access article licensed under the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reference: I. Bider, E. Perjons, and P. Johannesson, "Just Finished a Cycle of a Design Science Research Project: What's Next?" *Complex Systems Informatics and Modeling Quarterly*, CSIMQ, no. 22, pp. 60–86, 2020. Available: <https://doi.org/10.7250/csimq.2020-22.05>

Additional information. Author ORCID iD: I. Bider – <https://orcid.org/0000-0002-3490-6092>, E. Perjons – <https://orcid.org/0000-0001-9044-5836>, and P. Johannesson – <https://orcid.org/0000-0002-7416-8725>. PII S225599222000130X. Received: 23 October 2019. Accepted: 26 March 2020. Available online: 30 April 2020.

focus for these two contradictory directions. While the academic world is focused on theories that are aimed to *explain and predict*, in the categorization of [3], the real world needs the theories of *design and action*, also in the categorization of [3]. The article also suggests that the IS discipline needs to take the same path as medicine, engineering and architecture to become useful and relevant to the real world.

The paragraph above is included in this article to clarify the nature and aim of this work. This article is not about the **theory** of Design Science (DS), but about **DS practice**, more precisely, it is about how to conduct a DS research project in practice, and it is based, mainly, on the experience of the authors of running such projects. The article concerns only one, but important, issue of DS practice, namely, the nature of cycles of DS research projects. As a standard, a DS research project is considered as a sequence of phases, like *Identify problem* → *Define objectives of a solution* → *Design and development* → *Demonstration* → *Evaluation* → *Communication* [4]. Though cycles in the development are accepted, as, for instance, in [4], they are not encouraged or required to be in the plan. This constitutes a contrast to other practically oriented types of research, for instance, Action Research (AR) [5]. An AR project is considered to be inherently cyclic where each cycle in a reduced form can be represented as *Act* → *Review* → *Act* → *Review* ... [6].

There is a significant difference between the nature of cycles in AR and DS research. Cycles in AR are of the same type and they are aimed at getting better and better solutions to the initial problem. In contrast, DS cycles can be of varying nature, e.g. the next cycle can aim at adjusting the solution/artifact for solving a completely different problem than the one that was set for the previous cycle. Therefore, management of a DS research project would benefit from having rules on how to plan the next cycle dependent on the outcome of the previous one. These rules are not trivial; the best strategy depends on several factors, such as success or failure to solve the previous problem or availability of the current test site to continue testing (demonstration/evaluation phase) in the next cycle.

To the best of our knowledge, there is no framework, rules or systematic guidelines on how to choose what to do in the next cycle of a DS research project. Our research is aimed to fill this gap by developing a classification of possible cycles in DS research and guidelines for making an informed choice between them. This article presents results achieved so far towards this aim.

We would like to stress that this article is related to research practice, not to theory, the practical aim being facilitating the knowledge transfer to the younger generation of researchers. Being proponents of M. Polanyi, we agree that in the end, all knowledge is personal and tacit [7]. Seasoned researchers may not need any guidelines for planning and re-planning their research projects, as they can do it based on their tacit knowledge obtained through experiences. Transferring this knowledge to the younger generation on a tacit level, e.g. via apprenticeship, is effective but not very efficient as it may take years to complete. Thus, any explicit guidelines that might help in acquiring knowledge on how to conduct research could be useful. This especially concerns DS research that, as a new paradigm, does not have sufficiently many research centers that practice it, which makes arranging apprenticeship of a scale quite difficult. Note that though our guidelines are aimed at being used by novices in DS research, they might also be useful for the seasoned researchers, for instance, to make their choices more explicit and understandable for the whole team, or for the wider audience when publishing their work.

As we have a practical goal, our research itself is a DS research project where classification and guidelines of how to use it in practical circumstances are a solution/artifact to the problem of how to effectively manage a DS research project<sup>†</sup>. A proper way to create practical guidelines is by investigating the existing DS research practice. This can be done by analyzing practice reported in the literature, reflecting on our own practice, or following DS research projects completed by others.

As far as contemporary literature is concerned, the way research is reported nowadays makes it an unlikely source for investigation. In this article, we are interested in longer cycles, each of

---

<sup>†</sup> The analysis of the project from the DS perspective is placed in Section 6 of this article.

which results in designing, and possibly testing a solution/artifact. We need a report of at least a couple of completed cycles, so that we can analyze how these cycles are connected. Naturally, published papers concentrate on the results achieved, rather than details on how they have been achieved, and description of intermediate successes and failures. So far, we have not found reports in the IS area that could be used as a source of information for our investigation. Nevertheless, we have scanned and analyzed the existing literature related to cycles in DS research; and used what found when creating our classification and guidelines.

In this first step of creating a classification and guidelines, we mostly rely on analysis of our own experience of DS research projects, as here we have full information on how the projects went in reality, independent of how the results were reported in the papers. Thus, this research can be considered as a kind of reflective theory building [8] in the field of DS research where the authors function as practitioners. We have analyzed two representative research projects from our practice. Both projects were started in similar circumstances as problem solving projects in local practices. No specific plan of research was chosen at the start, as the main research focus was on formulating the problem and looking for a solution. However, at later stages the paths of these two projects diverged. Still both projects showed the presence of the cycles, but the analysis of them showed that the nature of these cycles were different. Both projects are in the mature phase with some results presented at scientific conferences and published in scientific journals. In both projects, authors of this article have actively participated, which gives us access not only to the results achieved in the project, but also to the details of the activities and decisions flows in the projects.

The first project in our study belongs to the field of enterprise modeling. It was started with the aim to develop a high-level business process modeling technique for deciding what kind of BPM tools/suites would be suitable for a particular business process [9]. In the next step, this technique was adjusted and tested for the analysis of suitability of tools already employed in a process [10]. After that, the technique was adjusted and tested for the analysis of the socio-technical structure of a software development project [11]. As the results of this project were applied to different sides of socio-technical systems, the project served as a source for designing specific guidelines for the DS research projects defining techniques applicable for modeling such systems. These guidelines are presented in Section 5.3.

The second project in our study belongs to the field of technology enhanced learning. It was started in connection to a problem in teaching/learning enterprise modeling in our department's courses and programs. More exactly, the students were learning how to draw an enterprise model, such as a process model, in a given modeling language based on a textual description, but not how to get information from a real-life enterprise that would serve as a basis for creating the model. To solve the problem, we decided to simulate a situation of apprenticeship using multi-media for presenting a realistic business case to the students as the basis for creating enterprise models [12]. The idea was implemented as a project web site that included multi-media presentation of the business case and a list of modeling tasks [13]. The site was tested at several occasions of the same course and was evaluated by students and teachers afterwards. The project then moved to the next step, investigating the possibility of sharing the whole or parts of a business case presentation in various courses. This direction was aimed at an additional educational goal – “combat” the fragmentation of knowledge in educational programs by creating connection between different subjects on the tacit level [14]. This project resulted in the participants getting AIS award for innovation in IS teaching in 2018.

To analyze the literature and our own experience, we needed to present activities completed in DS research projects in an abstract way, i.e. independently of the application domain of each project. The main requirement here was not to have any preconception on the flow of activities in the project. Based on this requirement, we chose to present a DS project as a series of movements inside and between two worlds suggested in [15]: (a) the real world of specific problems and solutions in local practices, and (b) the abstract world of generic problems and solutions. This approach worked well for the analysis of our own projects, and we believe it

would be also suitable for analysis of projects of others. The latter is planned as the next step in our work.

The rest of the article is structured in the following way. We start with describing the background and method in Section 2. After that, we move to the analysis of our knowledge base, literature devoted to cycles, in Section 3 and our own experience in conducting DS research in Section 4. After that, in section 5, we present our classification of cycles in DS research and guidelines on how to choose which cycle to use in the next iteration. In Section 6, we give examples of some other projects from our research practice and analyze them from the point of view of the classification and guidelines from Section 5. In Section 7, we present examples of more detailed guidelines related to specific DS research projects. In Section 8, we analyze the work presented in this article from the DS perspective. The last section, Section 9, contains concluding remarks and presents plans for future research.

## 2 Background and Method

As has already been introduced in the previous section, our work is connected to Design Science (DS) in two ways. Firstly, it concerns planning and managing DS research projects. Secondly, the research presented in this article is a DS project itself. We consider DS in a classical way [4], [16], [17] as being related to finding new solutions for problems known or unknown [18]. To be counted as a DS solution, or an artifact in the terminology of [4], [16], the solution should be of a generic nature, i.e. applicable not only to one unique situation, but to a class of similar situations, cf. Principle 1 of [19].

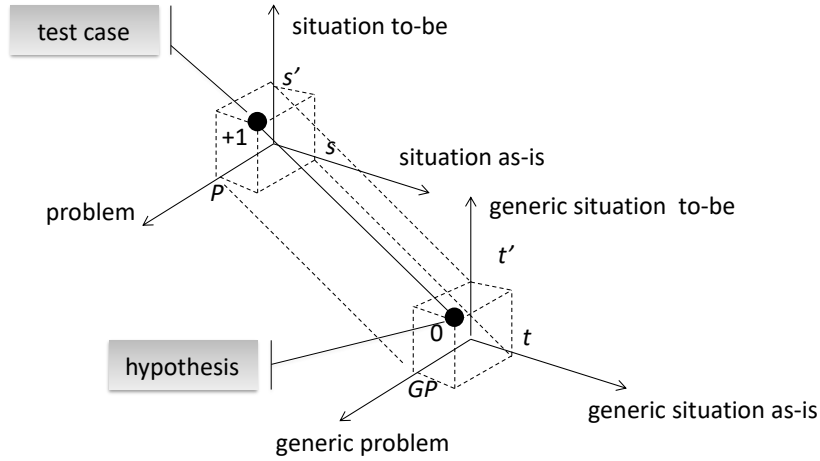
As our research concerns planning and management of DS research projects, we need a way of representing the actual, planned or possible traces of activities in a project. This way should be universal enough to be used for analyzing suggestions in the literature as well as examples from practice. For this end, we have chosen a way suggested in [15] that describes a DS research project as movement inside and between two worlds: (a) the real world of specific problems and solutions in local practices, and (b) the abstract world of generic problems and solutions. Here, we use terms “real” and “abstract” in a pragmatic, not philosophical way. As real, we consider a world where each individual system, e.g. enterprise or public organization, is treated separately, alongside with its specific problems and specific solutions. As abstract, we consider a world where each entity represents a certain class of systems; the problems and solutions in this abstract world become classes of problems and solutions as well, i.e. generic, as they represent patterns of system behavior.

As [15] might not be known to the reader, below, we give a short overview of its main ideas. The movements inside and between the two worlds mentioned above can be visualized, as in Figure 1. The upper part of Figure 1 represents the real world as the specific situation-problem-solution space<sup>‡</sup> with three axes: (1) situation as-is, (2) problem, and (3) situation to-be (i.e. a possible solution)<sup>§</sup>. A problem is defined as a set of situations considered undesirable from some point of view. For instance, the problem “a lack of communication between the members of our project team” can be represented as a set of all possible ways of organizing our team so that its members do not exchange important information between them in the frame of the project. The situation as-is, in this case, is “our team as of today”. A possible solution, situation to-be, could be “our team using collaborative software for communication”.

---

<sup>‡</sup> In [15], this space is called *individual situation-problem-solution space*. We use *specific* instead of *individual* in this article.

<sup>§</sup> This space includes all past, present and possible future situations, problems and solutions. Using orthogonal spaces in Figure 1 is a simplification that is made in order to use relatively simple pictures to illustrate the basic concepts. Note also that using the concept of possible solutions in no way implies that all solution already exists, they need to be discovered.



**Figure 1.** DS project's two worlds: the real world and the abstract world

A point  $\langle s, P, s' \rangle$  in the upper space in Figure 1 is called a test case. A test case represents a question of whether problem  $P$  in situation  $s$  (i.e.  $s \in P$ ) can be solved by transforming it into situation  $s'$  (i.e.  $s' \notin P$ ). The answer to this question can be either negative – transforming  $s$  into  $s'$  has not solved the problem  $P$  (i.e.  $s' \in P$ ), or positive – transforming  $s$  into  $s'$  has solved the problem  $P$  (i.e.  $s' \notin P$ ). In the latter case,  $s'$  is considered to be a solution for problem  $P$  in situation  $s$ . The results of tests can be visualized by assigning to point  $\langle s, P, s' \rangle$ :

- *weight* +1, if  $s'$  is a solution for  $P$  in  $s$  (see Figure 1), and
- *weight* -1, otherwise.

The lower part of Figure 1 represents the abstract world as a generic situation-problem-solution space with three axes: (1) generic situation as-is, (2) generic problem, and (3) generic situation to-be, (i.e. a possible generic solution). We consider that a generic situation (as-is or to-be) is represented by a template that defines a set of similar situations; the template serving as an extension of this set. The generic problem is defined as a set of situations that are undesirable from some point of view. The generic problem normally encompasses a larger number of situations than a specific problem from the real world. For instance, if a specific problem  $P$  can be defined as *a lack of communication between the members of a specific project team*, the corresponding generalized problem,  $GP$ , will be defined as *a lack of communication in project teams of certain kind*. The correspondence between  $P$  and  $GP$  can be defined through set inclusion:  $P \subset GP$ .

A point  $\langle t, GP, t' \rangle$  in the generic space is called a hypothesis and represents a question of whether a problem  $P$  ( $P \subset GP$ ) in a real world situation as-is  $s$  described by template  $t$  can be solved by transforming  $s$  into situation to-be  $s'$  according to template  $t'$ . Note that templates  $t$  and  $t'$  are not independent but use common variables, so that a pair  $\langle t, t' \rangle$  defines a kind of pattern based transformation. This transformation corresponds to the notion of artifact of type *method* used in DS literature [16]. Note that in practice, expressing an artifact in a declarative way as  $\langle t, t' \rangle$  is quite rare; normally the transformation is defined in a procedural way as some kind of algorithm. In the declarative definition as a pair  $\langle t, t' \rangle$ , the algorithm is separated from the definition. For practical purposes, it is important that the procedure/algorithm, besides being correct, is efficient. In this formalization, we do not explicitly discuss this requirement.

The relationships between the two spaces in Figure 1 are of the type *instantiation/generalization*<sup>\*\*</sup>. If real situation  $s$  satisfies template  $t$ , then  $s$  is referred to as an instantiation of  $t$ . Alternatively,  $t$  is considered to be a generalization of  $s$ . Extending the notion of instantiation/generalization to the points in the two spaces, test case  $\langle s, P, s' \rangle$  in the specific

<sup>\*\*</sup> Note that term *generalization* in this article is not used in the sense of UML where generalization is a relation between different classes. In this article, we use generalization as a relation between the points in specific vs generic state space.

space is considered to be an instantiation of hypothesis  $\langle t, GP, t' \rangle$  in the generic space if  $s$  is an instantiation of  $t$ ,  $s'$  is an instantiation of  $t'$ , and  $P \subset GP$ . Alternatively, hypothesis  $\langle t, GP, t' \rangle$  is called a generalization of test case  $\langle s, P, s' \rangle$ . Note that as  $t$  and  $t'$  use common variables, the specialization of  $t$  and  $t'$  cannot be done independently when instantiating a point in the generic space. The common variables should have the same values for both templates during instantiation. A similar requirement concerns generalization.

The knowledge about hypotheses in the generic space can be visualized by assigning weights to the points of this space according to the following rule:

- +1, when there is a statistically valid proof (many test cases) that  $t'$  is a generic solution for generic problem  $GP$  in generic situation  $t$ ;
- -1, when there is at least one test case instantiating the hypothesis with weights -1 assigned to it;
- 0, when there is at least one test case instantiating the hypothesis with weights +1 assigned to it, but the statistically valid proof has not been obtained so far.

In terms of the two spaces and weights introduced above, the primary goal of DS, as a process of generating and testing new solutions for adoption, can be defined as finding “blank” points (without weights) in the generic space and assigning them weights 0 or -1. Assigning weight +1 cannot be considered as a task of DS, as research itself cannot generate a statistically sufficient number of test cases (as discussed above). This assignment can only be made if practice adopts the solution and produces enough cases for empirical research to investigate. The efforts of DS are primarily directed at finding the generic solutions that work (weight 0). However, discovering the potential solutions that do not work during this process also constitutes valuable knowledge that should be published and marked on the map of Figure 1 with weight of -1.

DS does not impose a specific order of movements inside and between the two spaces. For instance, a researcher can start with a specific space and find an innovative solution for a problem in a specific situation, and then generalize the situation, problem and solution when moving to the generic space. In this case, the initial solution will automatically serve as a test case of a generic hypothesis. Another possibility is to start with a specific situation and problem, then generalize them and try to invent a generic solution that can be tested to solve the initial specific problem in the initial situation. From the opposite side, the researcher can start with designing a generic solution for an “unknown” problem, implement it in some specific situation, and then analyze whether it solves some specific problem, and whether this specific problem and situation as-is could be generalized.

Using a picture like Figure 1, we can depict a DS project trace via drawing, naming, and numbering arrows that show how the project proceeds inside each of the two worlds, and between them. Such traces will be used when analyzing literature and examples from our own practice in the next two sections.

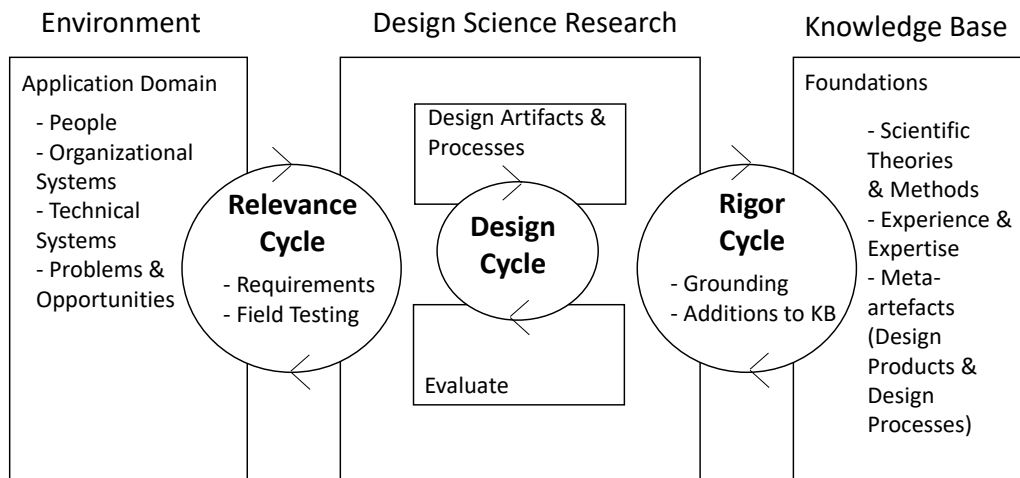
### 3 Knowledge Base – Literature on DS Cycles

This section offers an overview of the various types of cycles discussed in the DS literature. As theoretical literature on DS research is huge, we have chosen only representative works that explicitly discuss cycles in DS research. The section consists of three subsections, the first one surveys the literature, the second one discusses and analyzes the type of cycles introduced in this literature, and the third one clarifies which types of cycles we will address in this article.

#### 3.1 Multitude of Cycle Types in DS Literature

According to the seminal DS literature, e.g. [16], [19], [20], DS research typically is carried out in an iterative manner to cater for changing environments, shifting stakeholder interests, and unclear problem situations. In order to capture the iterative nature of design, researchers have

suggested various kinds of cycles that can be used to structure the DS research activities. Three main cycles are discussed in the literature (see, [21]): the *relevance* cycle, the *design* cycle, and the *rigor* cycle, see Figure 2.



**Figure 2.** Three main research cycles that are discussed in the design science literature [21]

The *relevance cycle* investigates the problems and opportunities in organizational practice that motivate the design science research, [16]. Furthermore, it provides the requirements on the artifact to be designed. Finally, it evaluates the designed artifact. This evaluation takes place in the application domain (environment, in Figure 2), for instance, in the form of field testing.

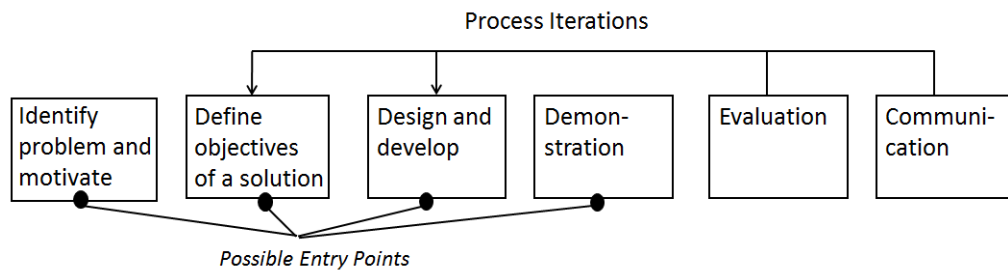
The *design cycle* is the “heart of any design science research project”, as stated by [21]. It consists of two major activities, build (design artifacts & processes in Figure 2) and evaluate. The cycle generates design alternatives and evaluates them against requirements until a satisfactory design is achieved.

The *rigor cycle* ensures that the designed artifact is a research contribution and not only routine design by relating the artifact to other artifacts existing in the knowledge base (KB) (see Figure 2) and argues for its innovativeness, [21]. Furthermore, the design of the artifact should be grounded in a knowledge base consisting of relevant kernel theories from other areas as well as other existing artifacts. Finally, the rigor cycle can add new findings from the design science research to the knowledge base. In order to elaborate on the rigor cycle, [22] adds the activity *theorize* to those of *build* and *evaluate* as well as three external practices: research community practice, general practice and local use practice. The article shows how these practices are related through several activity cycles, for instance, a *theorize – evaluate* cycle, *theorize – research community* cycle, and *build – use* cycle.

Beside the three major cycles in Figure 2, a fourth cycle has been suggested in [23], called the *change and impact cycle*, which relates the design science project to a wider environmental context, managing, for instance, long-term effects on and long-term changes in the environment. This kind of cycles is out of scope for this research, as, normally, it stretches beyond the timeline of an individual DS research project.

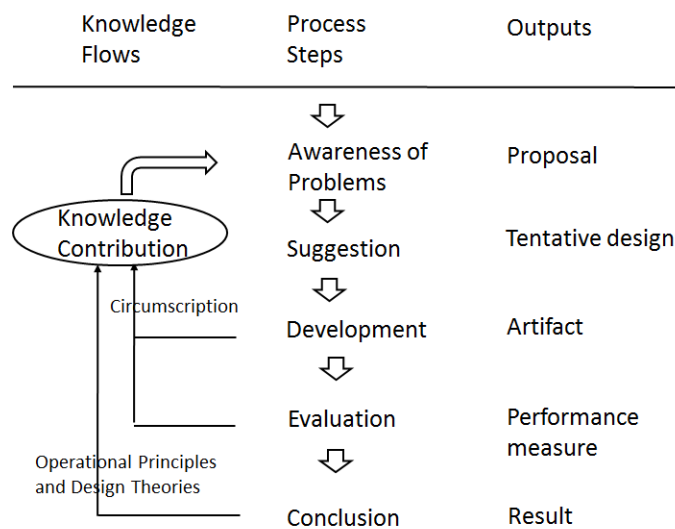
Various researchers have made an effort to integrate the three cycles of *design*, *relevance* and *rigor* into method frameworks for DS research. One example of such a framework is the DSRM process model [4], which consists of a set of phases (called “activities” in the original paper), presented in a sequential order as in Figure 3. The model allows cycles, i.e. the *Evaluation* and *Communication* phases can be followed by iteration back to earlier phases in the process model. For instance, the result of an evaluation can lead to changes in the design and development of the artifact aimed at improving the artifact (see arrow 2 in Figure 3). Important to note, however, is that the process iteration does not go all the way back to the Identify problem phase. Also, the order of phases in the model of Figure 3 is not mandatory for a DS project. DSRM does allow to

start with an artifact at hand and then go backwards to find a problem which the artifact is supposed to solve, see entry points in Figure 3. However, searching for the problem later on in a cyclic manner is not explicitly included in DSRM.



**Figure 3.** A design science process model adapted in a simplified form from [4]

Another version of a model of DS research process that includes cycles is presented in [20]. The model is built based on the cognitive perspective and is presented in Figure 4. A cycle can be initiated at any of the following phases (called “process steps” in the original work): *Development*, *Evaluation*, *Conclusion*, but always goes back to *Awareness of Problem*. The latter indicates each cycle results in a better understanding of the problem, which in turn can help in designing a better solution.



**Figure 4.** A design science process model adapted from [20]

A more agile version of the DS research process that includes cycles is proposed in [19] as an alternative to the DSRM. In this version, called Action Design Research (ADR), building an artifact, organizational intervention, and evaluation are intertwined in a cyclical process, inspired by action research. Similarly, [24] have suggested to introduce an evaluation of each activity in the DS process, thereby evaluating also problem analysis and requirements elicitation.

Besides the three classical cycles of *relevance*, *design*, and *rigor*, another cycle, which could be called the *abstraction* cycle, is often discussed in the literature. This cycle is related to distinction between the real world and the abstract world, as discussed in Section 2. For instance, the framework presented by [25] states that design is carried out in two distinct domains: instance and abstract, similarly to the specific and generic spaces presented in Section 2. The instance domain consists of instance solutions that aim to solve instance problems while the abstract domain consists of abstract solutions that aim to solve abstract problems. The activity going from instance problem to abstract problem is called abstraction, while going from an



abstract solution to an instance solution is called de-abstraction. The same kind of cycles is discussed in [26], but using a different terminology, i.e. *abstraction vs. application*.

Cycles in the frame of DS are also discussed in [27] and [28]. [27] proposes an evidence-based design cycle, which starts in scientific explanations, then moves on to generic technologies and further to customized solutions that are finally implemented in operations within one organization. [28] proposes a distinction between the research cycle and the client cycle, where the first one aims at developing an artifact addressing a class of problems while the second one aims at solving a problem at a specific client. Though distinct, the two cycles often become intertwined in design science projects when researchers move back and forth between specific problems and classes of problems, similarly to the ideas presented in Section 2.

### 3.2 Analysis and Discussion

The cycles discussed in the literature could be analyzed and given an interpretation based on considering DS research as a movement in the two worlds as described in Section 2. For instance, the cycles presented in Figure 2 can be interpreted and analyzed in the following way:

- The ending point of a *relevance* cycle, which we consider as the basic cycle of DS research process, is characterized by having a situation represented in Figure 1, with a test case given the value of +1 or -1, and the hypothesis given the value of 0 or -1.
- The starting point can differ from project to project, for instance, the cycle can start with a problem in a local practice which needs a solution, or with an artifact that needs a problem to be solved.
- The end of the relevance cycle in Figure 2 requires that the point in the abstract space is a new one, which implicitly requires having a *rigor* cycle inside the relevance cycle to ensure originality of the solution.
- The presence and importance of the *design* cycle inside the *relevance* cycle depends on the starting point of the relevance cycle. If we start with the problem, the design cycle is an important part of the next relevance cycle. However, if we start the *relevance* cycle with an existing artifact, there might be less needs for design.

The analysis shows that the cycles in Figure 2 are not independent but can be included in each other.

Let us now apply the metaphor of the two worlds for an analysis of ADR [19]. The main idea of ADR can be interpreted in the following manner. Instead of completing “long” movement in one world and then going to another one, ADR suggests moving more or less in parallel in these two worlds, i.e. constantly cycling between them to test whether a generic solution can be realized in a specific situation in a real practice.

Let us also consider the abstraction cycle [25] from the same perspective. The *abstraction* cycle is always part of the relevance cycle in terms of Figure 2, as the end of the relevance cycle includes points in both the abstract and the real world (called the abstract and instance domains in [25]). If we start with the real world, at least, one abstraction is required in the relevance cycle, and if we start with the abstract world, at least, one de-abstraction<sup>††</sup> is required. Normally, both abstraction and de-abstraction are included in one relevance cycle, e.g. local problem → generic problem (abstraction) → generic solution → local solution for local problem (de-abstraction).

Summing up the literature that deals directly or indirectly with cycles, we can conclude the following:

- Different works understand cycles in different ways, e.g. the scope of the cycle differs in different works from smaller cycles during the design as in ADR [19], to larger cycles that

---

<sup>††</sup> *De-abstraction* in terminology of [25] corresponds to *instantiation* used in other works

stretch over long periods of time in a DR research project (e.g. [16]) or even beyond such a project [23].

- While cycles are included in some practical methods, such as DSRM [4], there is a lack of guidelines how to decide whether to initiate a new cycle or not and how to organize and execute that new cycle. This article intends to fill this gap, at least partially, by providing methodological support for managing certain kind of cycles in DS research projects.

### 3.3 The Scope of This Work

In this work, we deal with cycles that are related to one DS research project. More concretely, we focus on the situation when one or more cycles in the project have been completed and a decision on how to conduct the next cycle needs to be taken. Thus, we are not concerned with how the previous cycle has been conducted, but only with the results achieved. Also, we are not concerned with how the next cycle will be planned in detail, but rather what goals to set for it.

The end cycle point, in which the decision on how to conduct the next cycle needs to be taken, is defined as:

- The project has reached a point of creating a hypothesis in the generic state-space, see Figure 1, which means that the generic situation, generic problem, and generic solution are specified. In the artifact terms of [4], it means that the project has developed an artifact that is ready for demonstration (see Figure 3), i.e. ready to be deployed in a specific type of situations for a specific purpose. The readiness is the *minimum* requirement, which means that at this point there is no intention to continue design before the decision on how to conduct the next cycle is taken.
- In addition, the project has completed one or more test cases and evaluated the results. In terms of Figure 1, it means that there are one or more points of the type test cases, i.e. points with assigned weights, in the specific situation-problem-solution space that are instantiations of the hypothesis. In DSRM terms of [4], see Figure 3, it means that one or several cases of demonstration have been completed.

The goal of the next cycle can be related to a change in the hypothesis along any of the axes in the generic situation-problem-solution-space. More concretely, it can be associated with:

1. Creating a better solution for the same class of specific situations, which means moving along the axis *generic situations to-be* (i.e. a possible generic solution) in Figure 1.
2. Trying the solution for another class of specific situation, which means moving along the axis *generic situations as-is* in Figure 1.
3. Trying the solution for a different class of specific problems, which means moving along the axis *generic problems* in Figure 1.

Note that case 1 above roughly corresponds to arrow 2 in Figure 3, while case 2 roughly corresponds to arrow 1 in Figure 3, and case 3 has no corresponding arrow in Figure 3. Note also that aiming at moving along one axis does not include not moving along others if needed.

## 4 Knowledge Base – Two Projects

In this section, we analyze and compare two DS projects from our practice focusing on the kinds of cycles that can be identified in them.

### 4.1 Project One – Developing a New Process Modeling Technique

The specific problem that initiated the first research project was encountered in the practice of a small Swedish consulting company, *IbisSoft*, to which one of the authors was affiliated. The company developed a tool for building Business Process Support (BPS) systems delivered as

web-based services. The tool, called *iPB* [29], is based on the state-oriented view of business processes [30], and it employs the shared spaces technique for organizing communication/collaboration between process participants [31].

The *iPB* tool was developed during a pilot project of building a system to support one of the processes in the social security office of the Swedish municipality of Jönköping. *iPB* proved to be useful, not only for supporting the original process of the pilot project, but also for other processes in the municipality. It proved to be suitable for supporting so-called loosely structured business processes, i.e. processes that were controlled by events and information gathered in the course of the process, rather than by a predefined sequence of operations.

Based on the results from the pilot project, *IbisSoft* decided to market *iPB* as a generic Business Process Management (BPM) tool/suite outside the municipality of Jönköping. In this business activity, an issue arose of how to convince a customer that the *iPB* tool was right for the process in question. An additional issue related to the main one was how to differentiate *iPB* from competing products, especially those that were built using the workflow technique. In the terms of Figure 1, the situation as-is and the problem in the upper state space (real world) when dealing with a specific customer can be defined as follows:

- (*Situation*) A given specific BPM tool, i.e. *iPB*, and a specific customer process, e.g. processing reclamations in a specific organization;
- (*Problem*) The customer has doubts that *iPB* is a suitable tool for their process
- (*Solution to find*) A convincing demonstration that *iPB* is a suitable tool for this particular process.

The first solution to be tested is to solve the problem while remaining on the specific level by developing a functioning prototype for the customer process using *iPB*. This solution, however, has a major drawback: if *iPB* shows to be unsuitable for the process in question, the time invested has been lost.

To eliminate this drawback, *IbisSoft* decided to change the point of view from a vendor-centered – how to sell the tool – to a customer-centered one, how the customer can find a suitable tool for their process(es). The latter can be formulated as:

- (*Situation*) A specific customer process, e.g. processing reclamations in a specific organization, and a specific set of BPM tools, like *iPB* and Apian BPM suite;
- (*Problem*) The customer does not know which tool is suitable for their process;
- (*Solution to find*) A tool to choose with convincing arguments on why it is the most suitable tool for this process.

In business terms, *IbisSoft* decided to develop an independent consulting service to match the needs of the process with the capabilities provided by various BPM tools available on the market. This, obviously, required generalizing the situation and problem, and finding a generic solution, i.e. to go over to the lower state space in Figure 1 (the abstract world). The generic situation and problem for such generalization can be formulated as:

- (*Generic situation*) A template  $t$  that defines the set of all possible pairs  $\langle x, Y \rangle$ , where  $x$  is a business process and  $Y = \{y_1, \dots, y_n\}$  is a subset of possible BPM tools.
- (*Generic problem*) For pairs  $\langle x, Y \rangle$ , it is not clear which tool in the set  $Y$  is suitable for process  $x$ .
- (*Generic solution to find*) A template  $t'$  that for each pair  $\langle x, Y \rangle$  in the set defined by  $t$ , identifies a tool  $y_i \in Y$  alongside with convincing argumentation on why tool  $y_i$  is the most suitable for process  $x$ .

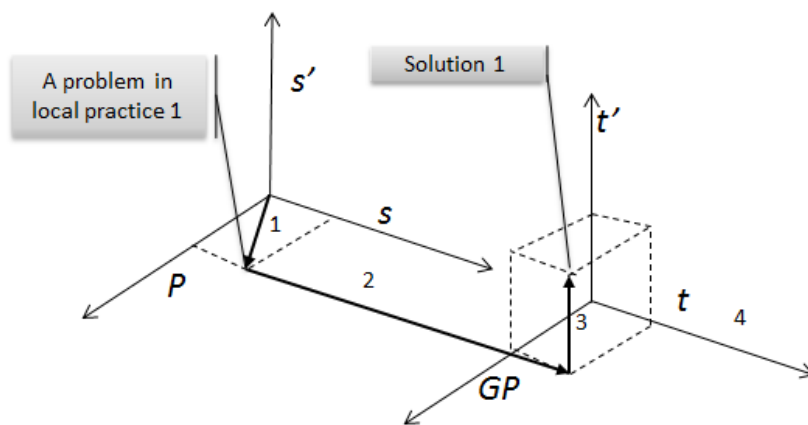
We did not set a requirement to mandatorily define the transformation from situation to solution in a declarative form  $\langle t, t' \rangle$ . Instead, we looked for a procedural method of analyzing a given pair  $\langle x, Y \rangle$  in order to identify the most suitable tool. It was also important for a procedure of determining  $y_i$  and producing argumentation should be cost-effective. Building a BPS system

using each of the tools and then making a comparison would not be such a good solution having this requirement in mind.

A solution for the generic problem above has been developed and published, see, for instance, [9]. The core component of this solution is a new business process modeling technique. The technique is aimed at building high-level business process models that are suitable for determining which capabilities of a BPM tool are needed for a particular business process. In this technique, a process model, called step-relationship model, is defined with the help of two kinds of elements: process steps, and relationships between the steps. The model can be presented in two ways (1) as several diagrams where the steps are represented as rectangles and relationships as arrows connecting these rectangles, and (2) as a set of orthogonal matrices representing relationships between the steps.

The trace of our project up to the point of developing the generic solution is represented in Figure 5, where:

- Arrow 1 corresponds to determining specific situation and problem,
- Arrow 2 corresponds to generalizing specific situation and problem,
- Arrow 3 corresponds to developing a generic solution.



**Figure 5.** Project 1 – from specific problem to generic solution

By the time the solution had been developed, the situation at *IbisSoft* changed so that all activities of marketing *iPB* as a general BPM tool were frozen. Thus, testing the new solution in a situation that originated the research became impossible, which initiated a search for a similar situation for testing. While we failed to find a situation of exactly the same kind, we found a slightly different situation that we considered appropriate for testing.

The new situation was at a large ICT provider who went through reengineering of their software development process. The reengineering concerned transforming this process from a traditional phase-based development with local software development teams, to a process of working in an iterative manner using the Scrum project management methodology and employing geographically distributed teams that have cultural differences and often work in different time zones.

The new software development process employed several tools for managing requirements and test cases, tracking bugs and problem reports among others. Still, the project management felt that there were some problems in the new settings that warranted analysis of the suitability of the tools employed. The decision was made to start a small-scale project to analyze the situation and suggest improvements in both the tools employed and the process organization.

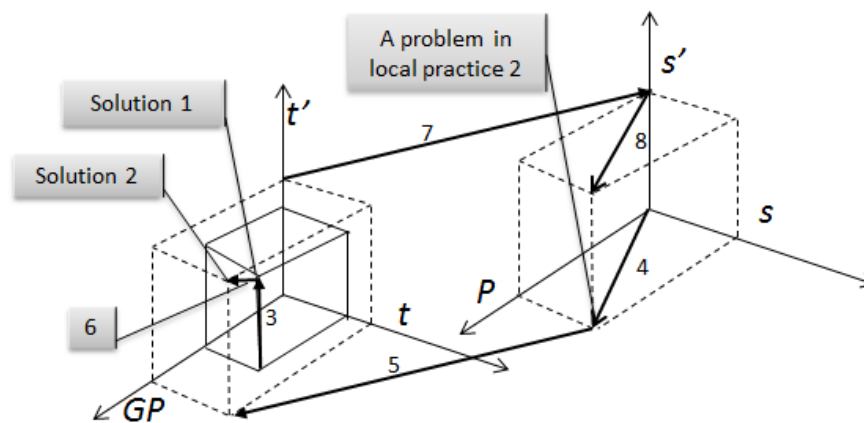
For the new specific situation and problem, a generalization has been made as follows:

- (*Generic situation*) A template  $t$  that defines the set of all possible pairs  $\langle x, Y \rangle$ , where  $x$  is a business process and  $Y = \{y_1, \dots, y_n\}$  is a set of IT tools used to support the process.

- (*Generic problem*) For any such pair  $\langle x, Y \rangle$ , it is not clear which tools in the set  $Y$  are suitable for process  $x$ , and which are not.
- (*Generic solution to find*) A template  $t'$  that for each pair  $\langle x, Y \rangle$  in the set defined by  $t$ , tags each tool  $y_i \in Y$  as to be suitable/not suitable for supporting the process  $x$  alongside with convincing argumentation on why.

As before, while looking for a solution, we did not try to express it in a declarative way, but rather as a method of investigating the correspondence between a process and the tools used in it. The project was completed [10] based on our step-relationship process modeling technique that had been adjusted to the new generic situation and problem by adding a new matrix related to the tools employed. The trace of the second part of the project is represented in Figure 6, where:

- Arrow 4 corresponds to determining a new pair of specific situation and problem,
- Arrow 5 corresponds to generalizing this pair,
- Arrow 6 corresponds to adjusting the existing generic solution,
- Arrow 7 corresponds to implementing the solution in the specific situation,
- Arrow 8 corresponds to validating the solution with project management to see whether they accept the results and recommendations (which they did).



**Figure 6.** Project 1 – adjusting and testing solution for a new situation-problem

Upon completion of the evaluation phase, it was discovered that the step-relationship model built for IT tools analysis could be used for analysis of social problems that were known to be present in the project. The problems were connected to professional and cultural differences between the project units, as well as geographical distance between them, which are typical for a global software development project. It was decided to start a new research initiative with another software development project in the same organization with the focus on detecting risks related to distances and mitigating them.

To implement this decision, a new cycle in our first project has been started and finished following the same pattern as in Figure 6, i.e. a new specific situation and problem were formulated and generalized to developing a modeling technique suitable for identifying potential areas of risks in a global software development project. While pursuing this goal, we modified, and extended our step-relationship process modeling technique and tested it in the suggested project. At this stage, we also changed the terminology from process-oriented to more system-oriented. For instance, the term of *step* was substituted by *functional component* (of a socio-technical system), and term *weak dependency* was substituted by term *feedback relationship*. As a result, we got a modeling technique for decomposing and presenting complex socio-technical systems, capable of representing such concepts as heterogeneity/homogeneity of teams and various kinds of distances between the systems components, e.g. geographical, cultural, professional, organizational [11].

Changing the problem to solve in each cycle was not intentional. It was not planned from the beginning of the project. The need for change was due not being able to test the artifact in the local practice where the initial problem had been discovered. Thus, a search for another test site has been initiated. When conducting the search, it was not possible to find an organization that was interested in the same problem, so we accepted the one that seemed suitable for testing provided that the problem and solution were slightly modified. In the end, the series of changes in the problem to solve and adjust the solution to the changes proved to be very productive. It helped to extend the modeling technique so that it covered a more interesting and urgent problem of global software development. This, hopefully, can lead to better chances of our solution being adopted by the industry.

Note that the description above represents a simplification in respect of the switching between the real and abstract worlds in the course of each outer cycle. Actually, in the first cycle, the generic solution/artifact development has been done mainly while in the abstract world, using a simplified example as a test-bed, and relying on the past experience. In the second, and especially the third, cycles adjustment of the generic solution has been done in parallel with testing it in the case organization. So, inside these outer cycles there were inner cycles of moving from the abstract to the real world and correcting the generic solution based on the results of the tests. These inner cycles correspond to the ones suggested in ADR, [19].

#### **4.2 Project Two – Technology Enhanced Teaching/Learning**

The second project in our study concerns teaching/learning modeling skills in Information Systems (IS) courses. It has been initiated by the problems in our own practice at the department of Computer and System Sciences (DSV) of Stockholm University. DSV has a tradition of using a case-based method for teaching/learning modeling skills. Cases are artificial and they are created by teachers as the needs arise. Cases are often reused in different occasions of the same course, often with modifications. A case represents an imaginary organization, e.g. an enterprise, a hospital, etc., and it consists of two basic components: case presentation and suggested solution. Case presentation is in the form of a text that describes the behavior and structure of the imaginary organization, and can include requirements on an IT system that the organization would like to introduce, or an organizational change that it wants to carry out.

The problem with the practice as described above is that it does not explicitly focus on teaching and learning of how to obtain information needed for building a model. Students merely learn syntax and semantics of modeling languages, and how to build a formal model based on text descriptions prepared for them by their teachers or found on the Internet. Therefore, they remain unprepared to real practice, which does not have any full textual descriptions, or even for conducting their field work related to BS or MS thesis.

Teaching/learning how to obtain information needed for building a model from unstructured reality is difficult while remaining in the classroom. This is a kind of tacit knowledge [7] called Ways of Thinking and Practicing in pedagogical literature [33]. This kind of knowledge is best acquired through apprenticeship [34]. As arranging apprenticeship for all students has not been practically possible, we have decided to simulate a situation of apprenticeship. In the simulated situation, the students follow a modeling master and help him/her to do some part of the work of building models. More specifically, the master chooses the information sources to be used for building a model, and hands the work of building the model to the students. Such sources include (but are not limited to): (a) recorded interviews with stakeholders, e.g. CEO, CIO, (b) samples of relevant documents, e.g. meetings protocols, forms for managing orders, (c) web-based sources, e.g. a company web site, results of twitter search on company name.

To implement the idea, we have built a website structure that combines multi-media information sources with description of modeling tasks and links to the simulated sources to be used in these tasks. The site was tested in a first year course that approximately corresponds to “Introduction to IS”. The site was used in three occasions of this course (in 2013, 2014 and

2015), and it was evaluated with the help of surveys and interviews by both students and teachers [12], [13]. The evaluation showed that the students appreciated the site and even suggested some improvements. Teachers experienced enhanced activity of the students and some progress in the examination results in comparison to the previous occasions of the same course, in which the case was described in textual form.

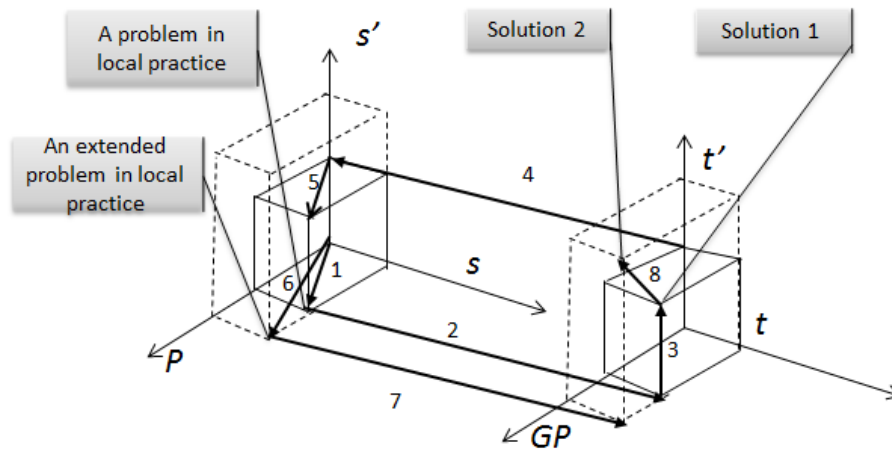
The next natural step would be using the same website structure for all courses. This, however, requires considerable resources to convert all current case descriptions into multi-media fragments and put them in the developed structure of the website. Though building a site for one case does not require much resources [13], the sum for all cases for all courses that include teaching/learning modeling skills may be quite high. To solve the resource problem, we are currently investigating possibility of reusing the multi-media case presentations between the courses. It is not clear whether a whole case can be reused in a different course, but it looks plausible to reuse some fragments, with or without modification [35].

Saving resources is only one problem that can be solved through case reuse. Another, and maybe more important, problem that can be solved in this manner is coordinating different courses taught in the frame of a program. Using a set of intersecting business cases may facilitate creation of connections between different islands of knowledge taught in various courses. This, in turn, will help the student to move from learning fragmented codified knowledge to acquiring holistic internalized knowledge in the discipline. Both problems are in the focus of our second cycle in this project that has been, at least partially completed [14].

The project discussed in this section can be presented as movement between specific and abstract worlds, as shown in Figure 7. The arrows in Figure 7 are as follows:

- Arrow 1: determining the problem in teaching/learning modeling skills in our specific courses;
- Arrow 2: generalizing the problem to the modeling courses in higher education of IT related disciplines, e.g. IS, computer science, via literature search;
- Arrow 3: developing a generic solution – the structure of the project web site and recommendations on how to fill it;
- Arrow 4: implementing the solution on one case in one specific course;
- Arrow 5: running the course occasion and evaluating the results;
- Arrow 6: extending the original problem to facilitating reuse of cases and/or fragments between the modeling courses in our department;
- Arrow 7: generalizing the extended problem to other departments/universities and their IT related courses;
- Arrow 8: finding a solution for the extended generic problem, if any. This is the current state in this project.

As with the first project, the trajectory of movement represented in Figure 7 is a simplification. Actually, the generalization of the specific situation, problem and solution was done in parallel with investigating the problem and creating a solution for a specific situation (specific course in our department). For instance, developing a technical structure for the web site used for simulating the apprenticeship was done in parallel with creating the content for the specific case and “feeding” it into the structure. Though, technically, the structure and the content were kept separated so that the structure could be used for other cases.



**Figure 7.** Project 2 – trajectory of movements inside and between the two worlds

### 4.3 Comparing the Two Projects

The comparison between the two projects described in the previous sections can be done based on comparing their traces represented in Figure 5–Figure 7. The result of this comparison is presented in Table 1 and is explained in more detail below.

**Table 1.** Results of projects comparison

| # | Parameter                     | Project 1  | Project 2  |
|---|-------------------------------|--|--|
| 1 | Initiation                    | In a local practice  | In a local practice  |
| 2 | Cycles                        | Yes  | Yes  |
| 3 | Testing                       | Different local practices in each cycle                              | The same local practice in each cycle  |
| 4 | Changes in problem definition | Different problem definitions for each cycle                         | Deeper understanding of the initial problem in each cycle                      |
| 5 | Changes in solution           | Aimed at adjusting the solution to a different problem in each cycle | Aimed at adjusting the solution to better solve the same problem in each cycle |

#### *Similarities*

- Both projects were initiated in local practices, *IbiSoft* in the first project, and *DSV* in the second project.
- Both projects have a cyclic nature, where design of generic solution/artifact alternates with its testing in a specific situation.
- In both projects, each cycle resulted in changing the problem definition to which the already built solution needed to be adjusted. In the first project, the problem was changed to reflect the needs of the local practice used for testing. In the second project, the problem was extended to employ the simulation in many courses while using reasonable amount of resources and attending a goal of tying together the knowledge the student got from different courses.
- In both projects, participation of the designers of the generic solutions in implementing them in the specific situations was of utter importance.

#### *Differences*

- While our first project changed the specific situation for testing in each cycle, the second project returned to the same specific situation in the next cycle (teaching in *DSV* courses).



- While the first project changed the generic situation and problem in each cycle rather arbitrary (actually, depending on the test case at hands), the second project was only extending the original problem (adding additional dimensions to the original problem of teaching/learning modeling).

### *Conclusions from the analysis*

As was explained in the previous sections, cycling through different problems in our first project was not part of a plan, but was forced by us being unable to continue with the initial problem and local practice in which it was detected. However, the cycling itself had a positive effect on the project, helping in finding a more interesting problem for our artifact/solution. Based on this observation, we can suggest that a DS project might benefit if a search for the “best problem” is initiated intentionally, thus becoming part of the project plan. In this case, at least some cycles in the project will be related to searching for a problem. The latter does not exclude a possibility of having cycles aimed at improving or extending a solution of the same or extended problem, which is the case with our second project.

If searching for a better problem becomes part of the plan, the question arises in which cases such search makes sense, i.e. having a chance to succeed. For instance, it is difficult to imagine that the solution being designed in our second project can be used for something else than education. Even if we were forced to change the local practice for testing the solution, we would still need a situation connected to improving teaching/learning. While it is difficult to answer the question generally, one hypothesis on the conditions that warrants such search can be derived from the analysis of our first project below.

The solution suggested in the first project consists of two components: (1) a new modeling technique, and (2) how to solve a specific problem using this technique. When we were adjusting our solution to a new problem, the first component was extended, while the second component was substituted to be in line with a practical problem we tried to solve. In the second cycle, it was substituted to a related problem, i.e. instead of the problem of developing tools/systems, we dealt with the problem of assessing the tools/systems already employed. In the third cycle, the distance between the initial problem and the one considered in this cycle became bigger. The third problem was identifying potential risks in a large distributed software development project considered as a complex socio-technical system.

Generalizing the analysis of the first project above, we can conclude that if a solution/artifact designed in a DS research project contains a new modeling technique as one of its components, a try should be given to use this technique for solving some other problem(s).

## **5 Artifact – a Classification of DS Cycles and Guidelines for Making a Choice**

### **5.1 A Classification**

In this section, we present a preliminary classification of cycles that can happen or be planned in a DS project. The classification is built based on the knowledge base presented in the previous sections. The classification is presented in Table 2, and it concerns the nature of the next iteration cycle in relation to the previous one. In addition, only relevance cycles are considered in this preliminary classification, i.e. cycles starting with a solution/artifact built in the previous cycle; thus, inner cycles described in [19], and mentioned in the ending paragraphs of the sections devoted to analysis of examples, are not included in the preliminary classification. The previous cycle may or may not include the demonstration phase, but it is considered as producing a triad <generic situation, generic problem, generic solution> that represents a point in the abstract world of Figure 1. The table has been designed by investigating all possible kinds of

changes that could be introduced in the next cycle in relation to the previous one and checking them against what has been found in the literature, and our own experience.

Table 2 includes five columns:

1. *Goal*. This column names the goal of the cycle and can be used as a name for the cycle itself.
2. *Result from the previous demonstration*. The column identifies whether the solution/artifact has been tested in practice in the previous cycle and whether the test has produced satisfactory results or not, i.e. the test case got +1 or -1 weight assigned to it.
3. *New generic problem*. The column identifies whether the new cycle will be related to the same problem, extended (more comprehensive) problem, or a different problem in relation to the previous cycle.
4. *New generic situation*. This column identifies the difference between the generic situation in the previous and the new cycle, i.e. whether it represents the same or extended (encompassing) application area, or a different application area.
5. *Comments*. This column explains the new cycle in more details and refers to the figures from the literature and/or examples discussed in this article where possible.

**Table 2.** A preliminary classification of DS cycles

| Goal (Cycle name)                          | Results from the previous demonstration        | New generic problem | New generic situation | Comments  |
|--|--|---------------------|-----------------------|---|
| <i>Improvement</i>                         | Not satisfactory                               | Same                | Same                  | Continue trying to solve the original problem. Corresponds to arrow 2 in Figure 3.  |
| <i>Enhancement</i>                         | Satisfactory                                   | Extended            | Same or Extended      | Solving a more comprehensive problem than the one identified in the previous cycle. Roughly corresponds to the cycle denoted in Figure 4 that ends with awareness of problems. Also, corresponds to the second and third cycles in our Project 2. |
| <i>Seeking an area of applicability</i>    | Not satisfactory                               | Same                | Different             | Continue trying to solve the original problem, but in another situation. For instance, switching from the situation of teaching children to the situation of teaching adults.   |
| <i>Extending the area of applicability</i> | Satisfactory                                   | Same                | Extended              | Looking for other areas whether the same generic problem can be found. For instance, testing a solution that works well when educating children in the situation of teaching adults.  |
| <i>Problem seeking</i>                     | Satisfactory/<br>Not satisfactory/<br>Not done | Different           | Different             | Corresponds to the cycles 2 and 3 in our Project 1  |

It is worthwhile to mention that only some of the cycles that can occur in DS research projects correspond to iterations known in practical design projects (i.e. projects that are not research projects), for instance, software development. In fact, only two types of cycling may be considered as normal for practical projects: *Improvement* and *Extending the area of applicability*. The most unusual for practice type of cycling is *Problem seeking*, as a practical project is always focused on solving a specific problem. However, this is not mandatory within research. Starting with one problem and discovering a solution for a different, and more interesting, problem is as good as finding a solution for the original problem. Therefore, using the *Problem seeking* cycles could be advantageous not only in cases where continuing with the original problem is not possible due to the external circumstances, but also for the sake of finding a more interesting problem to solve.

## 5.2 General Guidelines

As can be seen from the Table 2, there are several alternatives among which to choose for the next cycle, even for the same result of the test from the previous cycle. Based on our experience, we suggest guidelines for making a choice, which are presented in Table 3 that includes four columns:

1. *Result from the previous demonstration.* This column has the same meaning as the column with the same name in Table 2.
2. *Assumption about situation.* The column identifies the researchers' subjective opinion on the potential prospects of the generic solution developed in the previous cycle.
3. *External environment.* This column contains assessment of chances to arrange a test case for demonstrating the validity of a solution in real world.
4. *Cycle to choose.* This column contains the name of the cycle that corresponds to the situation at hand. This name can be used as an entrance to Table 2 via its column 1 (Goal).

**Table 3.** Pragmatic rules for choosing the next cycle

| # | Results from the previous Demonstration        | Assumption about the generic solution  | External environment (possibility to test the generic solution)  | Cycle to choose                            |
|---|--|--|--|--|
| 1 | Not satisfactory                               | Can be improved to fit the original generic situation and problem.   | The improved generic solution can be instantiated and tested in the same specific situation, or in a similar environment available for testing.  | <i>Improvement</i>                         |
| 2 | Not satisfactory                               | Can be adapted to fit a different generic situation to solve the original generic problem.                       | There is a possibility to find a specific situation that corresponds to a new generic situation for testing the adapted generic solution.  | <i>Seeking an area of applicability</i>    |
| 3 | Satisfactory                                   | Can be adapted to fit a more comprehensive (extended) generic problem in the same or extended generic situation. | A more comprehensive problem exists, and the adapted generic solution can be instantiated and tested in the same specific situation or a similar situation where this extended problem is available for testing. | <i>Enhancement</i>                         |
| 4 | Satisfactory                                   | Can be adapted to fit another generic situation solving the same generic problem.                                | There is a possibility to find a specific situation that corresponds to a new generic situation for testing an instantiation of an adapted generic solution.   | <i>Extending the area of applicability</i> |
| 5 | Satisfactory/<br>Not satisfactory/<br>Not done | Can be adapted to fit a different generic situation and different generic problem.                               | There is a possibility to find a specific situation that corresponds to a new generic situation and a new generic problem for testing an instantiation of an adapted generic solution.                           | <i>Problem seeking</i>                     |

Note that the guidelines in Table 3 do not give a single suggestion for a particular DS research project, as there can be several assumptions about the prospects of the solution under development, and more than one possibility to arrange a test case. Note also, that at the time being, the guidelines do not support making a correct assessment of the prospects of the solution developed in the previous cycle. Creating guidelines on how to make such assessments are outside the scope of this article.

## 6 Short Analysis of Other Examples of DS Research Projects

Though the results presented in Section 5 are derived from the analysis of only two of our DS research projects, similar types of cycles that we described in Section 4 can be observed in our other projects. In this section, we give two examples of such projects without carrying out a formal analysis based on the metaphor of two worlds.

## 6.1 Fractal Enterprise Model

Fractal Enterprise Model (FEM) is a technique for enterprise modeling [36]. FEM has a form of a directed graph with two types of nodes, *Processes* and *Assets*, where the arrows (edges) from assets to processes show which assets are utilized by which processes and arrows from processes to assets show which processes help to have specific assets in “healthy” and working order. The arrows are labeled with meta-tags that show in what way a given asset is utilized, e.g. as *workforce*, *reputation*, *infrastructure*, etc., or in what way a given process helps to have the given assets “in order”, i.e. *acquire*, *maintain* or *retire*.

Initially, FEM has been developed as a means for finding all or the majority of the processes that exist in an organization. The result of this research produced more than a solution to the original problem, as FEM includes not only relations between the processes, but produces a map of assets usage and management in the organization. Therefore, we continue our work on FEM looking for other problems/challenges that can be solved using FEM; and enhancing FEM when necessary [37]. One example of a specific application of FEM is using FEM for business model innovation, see, for instance, [32]. Another example is using FEM for arranging process documentation to make it possible to more easily find a model someone needs [38].

From the point of view of guidelines from Section 5, the first cycle of the project ended up with an artifact/solution that consisted of (1) a modeling technique and (2) how to use it for finding the majority of the business processes in an organization. The second part of this artifact has never been tested in practice. The first part showed to be quite interesting on its own, so we decide to start a new “problem seeking” cycle according to the fifth row in Table 3. The problem seeking went in different direction, resulting into several different, but connected, subprojects named above.

## 6.3 Business Process Canvas

Business Process Canvas (BPC) is a model of a process in a nutshell that presents the essential properties of the process and the context in which it is run, including the position of the process in the business process ecosystem [39]. The canvas was initially designed to solve an educational problem of the students having difficulties in business process modeling, especially, when the modeling should be done not in a workflow notation [40]. The original idea was that the students would use the canvas to gather basic information about the process before starting modeling. Thus, the original idea of BPC was creating a tool that would improve the quality of process models in the educational context. BPC has been tested in two course rounds of the course called Business Process and Case Management (BPCM). The results achieved were encouraging. The students considered that BPC helps in both building process models and getting better understanding of the concepts of the business process domain. We also undertook some efforts of dissemination of using BPC in the educational context by holding tutorials at international conferences.

While BPC was initially designed for a particular educational problem, its structure has no special connection to education. Therefore, we consider that the next cycle or cycles of the project could concern using BPC in other context or for another goal. An example of another context could be using the canvas in professional modeling practice. Going in this direction corresponds to the fourth row in Table 3 (extending the area of application). Another opportunity is using BPC in the decision making. This direction represents both changing the situation and the problem. The situation is decision making, and the problem can be formulated as insufficient data or not clearly presented data used in the decision making. We consider that this direction is in accordance to the fifth row of Table 3 – looking for another problem to solve. In such cases, the situation (context) often is different than in the previous cycle.

## 7 Developing More Detailed Guidelines

The guidelines presented in Section 5 in form of Table 3 have a general nature; they can be applied to any DS research project. Developing more detailed guidelines require narrowing down the type of situations, problems or solutions with which a DS research project deals. In this section, we present an example of detailed guidelines for the DS projects that deal with socio-technical systems. The reader who is not acquainted with the concept of socio-technical systems, or has no interest in such systems, can skip this section.

### 7.1 Specific Guidelines for the Case When a New Modeling Technique is Involved

As has been discussed in Section 5.2, a case where part of a solution/artifact is a new modeling technique is especially suitable for choosing *Problem seeking* for the next cycle, see the last row in Table 3. In this section, we will elaborate on which choices are open in this case in the IS domain. The elaborated guidelines are based on two characteristics of using models in an IS domain:

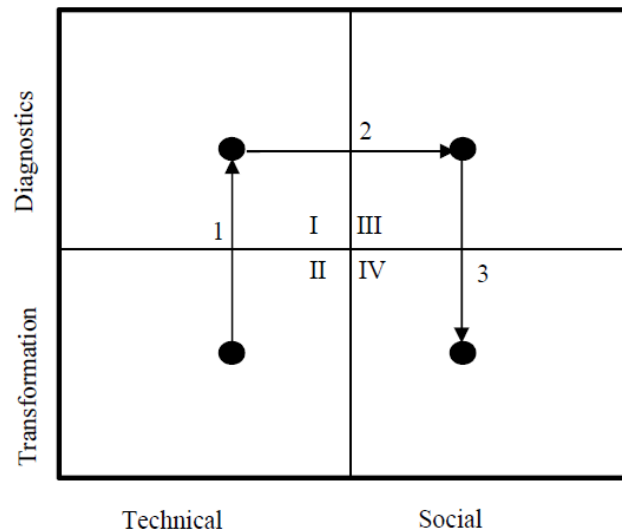
1. The use of modeling technique in practice of IS can be roughly classified in two categories: (a) *transformation*, (b) *diagnostics*. Under transformation, we understand using modeling for creating something new or different, for instance, a new IT system, or a different business model. Under diagnostics, we understand using modeling for uncovering problems in a given system, for instance, organization, for instance, analyzing risks. Actually, diagnostics is also a kind of transformation, but it happens in the mental plan of the stakeholders rather than in the physical world.
2. IS deals with socio-technical systems [41], [42], which consist of two parts (or two views) social and technical. A particular model can be related more to the technical part (view) or to the social one. These two parts could be split in their own terms: social can be split to people and structure, and technical to tasks and technology. The four parts can be represented as a 2x2 matrix as in [42] or in a more abstract way as in, for instance, [43] or [44], but we will not go to this level of details. The different parts of a socio-technical system are tightly integrated and should be aligned with each other [44]. However, changes in them are, often, initiated by changes in one of them, which requires realignment. Thus, it is possible to investigate problems, and suggest solution related to one part when the other is considered as fixed.

Based on the two categorizations above, we can define a two-dimensional space to classify the use of modeling in IS practice as shown in Figure 8. The figure depicts four distinct quadrants:

- I. *Technical Diagnostics (TD)* – diagnostics of a technical (IT) system, e.g. how well it corresponds to the social one, or/and the external environment.
- II. *Technical Transformation (TT)* – transformation of a technical system, e.g. to make it better suit the social one, or/and the external environment.
- III. *Social Diagnostics (SD)* diagnostics of a social system, e.g. how well it corresponds to the technical one, or/and the external environment.
- IV. *Social Transformation (TS)* transformation of a social system, e.g. to make it better suit the technical one, or/and the external environment.

Changes in the problem definition for the next cycle can be defined as moving between the quadrants in Figure 8. As an example, consider our Project 1 from Section 4. It was started in Quadrant II with the goal of developing a method to define requirements on a BMP tool/suite for building a support system for a given process (i.e. transformation of a technical system). In the next cycle, we moved to quadrant I (see arrow 1 in Figure 8) to adjust our modeling technique for analyzing the suitability of software tools already employed for supporting a given process (i.e. diagnostics of a technical (IT) systems). In the third iteration, we moved to quadrant III (see

arrow 2 in Figure 8) adjusting our modeling technique for investigating risks in socio-technical structure related to the distances between the members of a globally distributed project team (i.e. diagnostics of a social system). In project 1, we have made one more transition that has not been mentioned in Section 4, namely, we adjusted the modeling technique for a method of gradual transition from the traditional software development to the agile software development [45], (see arrow 3 in Figure 8) (i.e. transformation of a social system). With this transition, we moved to quadrant IV, making the full traverse of the possible usage of a modeling technique.



**Figure 8.** A problem seeking space

Figure 8 can be used for determining the next usage of a modeling technique according to the simple procedure. First, determine in which quadrant the current usage lie, then deliberate whether the technique could be used in another quadrant. The easiest way, probably, would be to go to an adjacent quadrant, as was the case in our Project 1. It does not mean, however, that moving diagonally is forbidden.

Note that classification presented in Figure 8 is not the only one that could be useful for *Problem seeking*. Another dimension for deliberation could, for instance, be introduced based on the dichotomy *Using in practice vs. Using in Education*, see an example in Section 6.

## 7.2 Directions for Further Development of Detailed Guidelines

As has been stated in the introduction, the classification and guidelines presented in this article are mainly derived from the analysis of our own DS projects, more specifically, from the analysis of two projects described in Section 4. Though, in principle, additional guidelines could be suggested based on theoretical considerations, we believe that it would be better to continue the guidelines development based on the analysis of practice, both our own as well as the practice of others. In particular, the rules for *Seeking an area of applicability* and *Extending the area of applicability* deserve special consideration.

As in the case described in the previous section, the rules might be dependent on the nature of the solution/artifact in question. One potentially useful classifications of areas of applicability for an artifact in the form of IT system to support business processes is suggested in [44]. The context of usage of the system is defined in terms of flexibility of the business process to be supported by it, which produces four generic categories: *Loose* process (most flexible, less standardized), *Guided* process (some standardization has been achieved), *Restricted* process (much standardized, but with some flexibility left), *Stringent* process (fully standardized allowing no deviations). Each category of context sets different requirements on a Business

Process Support (BPS) system. Thus, a failure of a new kind of a BPS system in one context may be reverted when trying in a different and more suitable context.

Besides setting requirements on BPS system, which is a technical component of a socio-technical system, the framework from [44] also sets requirements on other components of the socio-technical system, as well as defines rules for the correspondence between external environment and process flexibility. These also can be used when assessing success or failure of introducing a new BPS in an organization. For instance, the system might be adjusted to the given level of flexibility, but other components of the given socio-technical system, e.g. people, are not adjusted to it. Then the failure of introducing the system in this context may be related not to the suitability of system as such, but to other components of the socio-technical system not being adjusted to the given level of flexibility. Based on this analysis, a more exact definition of a potential area of applicability of a new BPS can be defined.

Defining specific rules for choosing a new area of applicability is not in the focus of this article. The deliberation above is presented to show that defining guidelines for applicability is possible; but requires further investigation which is in our plans for the future.

## 8 Our Research as a DS Research Project

As has been pointed in the introduction, we consider the development of the classification and guidelines as a DS research project on its own. In terms of the real world, a specific situation, the problem and solution for this project can be described in the following way:

- (*Situation*) A researcher has developed a solution/artifact for some generic problem that can be represented as a point in the generic situation-problem-solution space  $\langle t_1, GP_1, t'_1 \rangle$ . In addition, this solution might have been tested in a specific situation to solve a specific problem and get the result of success or failure. In other words, there can be a point in the specific situation-problem-solution space  $\langle s_1, P_1, s'_1 \rangle$  to which a weight -1 or 1 has been assigned. In addition, the researcher has some assumptions about his/her solution/artifact and some potential testing sites.
- (*Problem*) The researcher is not satisfied with the current state of the project; but does not know how to proceed. The dissatisfaction can be because the test of the solution has been negative, or the researcher is interested in the further development of the solution.
- (*Solution to find*) A new pair of  $\langle \text{generic situation, generic problem} \rangle - \langle t_2, GP_2 \rangle$  and convincing arguments that  $\langle t_2, GP_2 \rangle$  is a good candidate for adjusting  $t'_1$  to become a solution for  $GP_2$  in situation  $t_2$ .

Note that points in the generic space become part of the specific space which corresponds to the meta-level of our current project. A generalization of such  $\langle \text{situation, problem, solution} \rangle$  can be defined in the following way:

- (*Generic situation*) A template  $t$  that defines the set of points in the generic situation-problem-solution space  $\langle t_1, GP_1, t'_1 \rangle$  each of which supplied, but optionally, with a test case (or a set of test cases) with a weight assigned to it.
- (*Generic problem*) For each of the points and the test case(s) attached, it is not clear how to proceed in the next iteration of the project.
- (*Generic solution to find*) A template  $t'$  that for each point  $\langle t_1, GP_1, t'_1 \rangle$  with attached test case(s) and weight(s) defined by  $t$  assigns a pair of  $\langle \text{generic situation, generic problem} \rangle = \langle t_2, GP_2 \rangle$  and convincing arguments that  $\langle t_2, GP_2 \rangle$  is a good candidate for adjusting  $t'_1$  to become a solution for  $GP_2$  in situation  $t_2$ .

The classification and guidelines presented in Section 5 represent the first version of a solution to this problem expressed in procedural form, more specifically, in the form of decision tables. As the classification and rules have been designed, mostly, based on the analysis of our own DS projects, the latter can be considered as test cases for the solution presented. However, the real

test for the usefulness of our solution comes when somebody decides to use them for answering the question posed in the title “Just finished a cycle of a design science research project. What’s next?”.

## 9 Conclusion

### 9.1 Contributions

The contribution of this article lies in the area of providing methodological support for DS research. Here, we understand term *methodology* in very general sense, as, for instance, defined in the Cambridge dictionary: “methodology – a system of ways of doing, teaching, or studying something”. The actual contributions of this article can be summarized as follows:

1. Our literature overview shows that there is no published research that systematically investigates all possible cycles in a DS research project. In particular, there is no classification of possible cycles accompanied with guidelines on how to choose the next cycle based on the results from the previous one. The absence of such a classification and guidelines might hinder the efficient transfer of knowledge on how to conduct a DS research project to the new generation of researchers. This constitutes a particular problem in the expansion phase of the new paradigm, when the experienced researchers are few, while the number of young proponents is growing. Formulating the problem as above and highlighting its importance constitutes the first contribution of this article.
2. Our approach for developing a classification as well as guidelines is based on the analysis of existing practice. For this end, we proposed to use presentation of a DS project as trajectory of movement inside and between two worlds from [15]. The way of how such approach can be applied has been demonstrated in Section 4 that analyzes two DS research projects from our own practice. We believe that this approach can be applied to other DS research projects in order to extend the classification and guidelines presented in this article. Such approach might also be useful for analysis of DS practice while having other objectives in mind.
3. Based on the analysis of our experience of DS projects and DS literature, we have drafted a classification of relevance DS cycles (Section 5.1) and guidelines on how to choose the next cycle (Section 5.2). The type of the next cycle is “calculated” based on the result achieved in the previous cycle, assumptions about the solution/artifact and possibility to find a site for testing. The classification and rules are of general nature and require further elaboration and specialization. They might seem trivial for an experienced DS researcher, but could be useful for the newcomers. As we stressed in the introduction, our aim is not theorizing on the topic of DS, but present rules that can orient novices and give them better understanding of the nature of DS projects. For this aim, we believe even general rules of heuristic type as presented in this article are better than nothing.
4. In Section 7, we have shown that the general rules of Section 5.2 could be elaborated and specialized considering the nature of specific DS research projects. This section shows how the more elaborated guidelines could be developed, and it presents a first step in this direction.

### 9.2 The Nature of Our Guidelines

Our work on methodological support for DS research somewhat differs from other approaches in this direction, such as the DSRM process model [4], or ADR in [19]. We do not propose a procedure to follow, but rather guidelines for decision making when a certain point has been reached in a DS research project. This is true for our original work [15], and for suggestions in this article. The guidelines neither cover the steps needed to reach the decision point, nor provide a stepwise plan what to do next (they rather set the goal to be achieved in the next cycle). Thus,



the guidelines require some understanding and, even better, experience (positive and/or negative) of DS research projects.

We are fully aware that non-procedural nature of the guidelines might make them challenging for teaching/learning in educational classes. However, using these guidelines does not prohibit or contradict using any procedural method, such as DSRM or ADR. In a teaching/learning situation, the guidelines from Section 5 could constitute an addition to more procedural methods. So far, we have not tested using our guidelines in the educational process, but we plan to do it in the future.

### 9.3 Plans for the Future

In our view, an appropriate way to proceed with developing the classification of DS cycles and guidelines for choosing the next cycle is to continue investigating existing practice, our own, as well as the practice of others. As papers describing practical cases at the needed level of details are rare, the best way to proceed would be the collective development of the classification and guidelines, where other researchers join the efforts and start investigating their own practice. Disseminating the results achieved so far could help to attract other DS researchers to the problem and inspire them to reflectively investigate their practice. Publication of this article would serve as a means for this purpose.

### References

- [1] A. S. Lee, "Retrospect and prospect: information systems research in the last and next 25 years," *Journal of Information Technology*, vol. 25, pp. 336–348, 2010. Available: <https://doi.org/10.1057/jit.2010.24>
- [2] H. Simon, *The Sciences of the Artificial*. Cambridge, MA: MIT Press, 1969.
- [3] S. Gregor, "The Nature of Theory in Information Systems," *MIS Quarterly*, vol. 30, no. 3, pp. 611–642, 2006. Available: <https://doi.org/10.2307/25148742>
- [4] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–78, 2007. Available: <https://doi.org/10.2753/MIS0742-1222240302>
- [5] R. Baskerville and M. D. Myers, "Special Issue on Action Research in Information Systems: Making IS Research Relevant to Practice – Foreword," *MIS Quarterly*, vol. 28, no. 3, pp. 329–335, 2004. Available: <https://doi.org/10.2307/25148642>
- [6] ALARA, 2002. Action Research. [Online] Available: <http://www.alarassociation.org/pages/al-and-ar/action-research> [Accessed April 2015].
- [7] M. S. Polanyi, *Knowing and Being*. Chicago: University of Chicago, 1969.
- [8] V. Mott, "Knowledge comes from practice: Reflective theory building in practice," *New Directions for Adult and Continuing Education*, vol. 1996, no. 72, 1996. Available: <https://doi.org/10.1002/ace.36719967209>
- [9] I. Bider and E. Perjons, "Design science in action: developing a modeling technique for eliciting requirements on business process management (BPM) tools," *Software & Systems Modeling*, vol. 14, pp. 1159–1188, 2015. Available: <https://doi.org/10.1007/s10270-014-0412-6>
- [10] I. Bider, A. Karapantelakis, and N. Khadka, "Building a High-Level Process Model for Soliciting Requirements on Software Tools to Support Software Development: Experience Report," *Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013)*. CEUR, vol. 1023, Riga, Latvia, pp. 70–82, 2013.
- [11] I. Bider, H. Otto, and S. Willysson, "Using a Socio-Technical Model of a Global Software Development Project for Facilitating Risk Management and Improving the Project Structure," *Complex Systems Informatics and Modeling Quarterly*, vol. 15, pp. 1–23, 2018. Available: <https://doi.org/10.7250/csimq.2018-15.01>

- [12] I. Bider, M. Henkel, S. Kowalski, and E. Perjons, "Simulating apprenticeship using multimedia in Higher Education: a case from the information systems field," *Interactive Technology and Smart Education*, vol. 12, no. 2, pp. 137–154, 2015. Available: <https://doi.org/10.1108/ITSE-04-2015-0004>
- [13] I. Bider, M. Henkel, S. Kowalski, and E. Perjons, "Teaching enterprise modeling based on multi-media simulation: a pragmatic approach," *Proceedings of 6th International MCETECH Conference, LNBIP*, vol. 209, Springer, pp. 239–254, 2015. Available: [https://doi.org/10.1007/978-3-319-17957-5\\_16](https://doi.org/10.1007/978-3-319-17957-5_16)
- [14] I. Bider and M. Henkel, "Using the structure of tacit knowing for acquiring a holistic view on is field," *Proceedings of the 11th IADIS International Conference Information Systems 2018, IS 2018*. IADIS Press, pp. 19–26, 2018.
- [15] I. Bider, P. Johannesson, and E. Perjons, "Design science research as movement between individual and generic situation-problem-solution spaces," *Organizational Systems. An Interdisciplinary Discourse*. Springer, pp. 35–61, 2013. Available: [https://doi.org/10.1007/978-3-642-33371-2\\_3](https://doi.org/10.1007/978-3-642-33371-2_3)
- [16] A. R. Hevner, S. T. March, J. Park and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. Available: <https://doi.org/10.2307/25148625>
- [17] R. L. Baskerville, J. Pries-Heje, and J. Venable, "Soft Design Science Methodology," *DERIST 2009*. ACM, pp. 1–11, 2009. Available: <https://doi.org/10.1145/1555619.1555631>
- [18] J. Anderson, B. Donnellan, and A. R. Hevner, "Exploring the Relationship between Design Science Research and Innovation: A Case Study of Innovation at Chevron," *EDSS 2011, CCIS*, vol. 286, Springer, pp. 116–131, 2012. Available: [https://doi.org/10.1007/978-3-642-33681-2\\_10](https://doi.org/10.1007/978-3-642-33681-2_10)
- [19] M. K. Sein et al., "Action Design Research," *MIS Quarterly*, vol. 35, no. 1, pp. 37–56, 2011. Available: <https://doi.org/10.2307/23043488>
- [20] V. Vaishnavi and W. Kuechler, *Design Science Research in Information Systems*. [Online] [desrist.org](http://www.desrist.org) (October 23, 2013) Available: <http://www.desrist.org/desrist/> [Accessed May 2015].
- [21] A. R. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems*, vol. 19, no. 2, article 4, 2007.
- [22] G. Goldkuhl, "Activity Cycles in Design Research: A Pragmatic Conceptualisation of Inter-related Practices," *European Design Science Symposium (EDSS 2012)*. Springer, CCIS, vol. 388, pp. 49–60, 2013. Available: [https://doi.org/10.1007/978-3-319-04090-5\\_6](https://doi.org/10.1007/978-3-319-04090-5_6)
- [23] A. Drechsler, and A. R. Hevner, "A Four-Cycle Model of IS Design Science Research: Capturing the Dynamic Nature of IS Artifact Design," *Breakthroughs and Emerging Insights from Ongoing Design Science Projects: Research-in-Progress Papers and Poster Presentations from the 11th International Conference on Design Science Research in Information Systems and Technology (DESRIST)*, 2016.
- [24] C. Sonnenberg, and J. vom Brocke, "Evaluation Patterns for Design Science Research Artefacts," *Practical Aspects of Design Science. European Design Science Symposium*. Springer, pp. 71–83, 2011. Available: [https://doi.org/10.1007/978-3-642-33681-2\\_7](https://doi.org/10.1007/978-3-642-33681-2_7)
- [25] J. S. Lee, J. Pries-Heje, and R. Baskerville, "Theorizing in Design Science Research," *DESRIST 2011*. Springer, LNCS, vol. 6629, pp. 1–16, 2011. Available: [https://doi.org/10.1007/978-3-642-20633-7\\_1](https://doi.org/10.1007/978-3-642-20633-7_1)
- [26] S. Gregor, L. Chandra Kruse, and S. Seidel, "The Anatomy of a Design Principle," *Journal of the Association for Information Systems*, Forthcoming, 2020.
- [27] R. Winter, "Towards a Framework for Evidence-Based and Inductive Design in Information Systems Research," *Design Science: Perspectives from Europe. European Design Science Symposium*. Springer, pp. 1–20, 2013. Available: [https://doi.org/10.1007/978-3-319-13936-4\\_1](https://doi.org/10.1007/978-3-319-13936-4_1)
- [28] R. Wieringa and A. Morali, "Technical Action Research as a Validation Method in Information Systems Design Science," *Design Science Research in Information Systems. Advances in Theory and Practice, LNCS*, vol. 7286, Springer, pp. 220–238, 2012. Available: [https://doi.org/10.1007/978-3-642-29863-9\\_17](https://doi.org/10.1007/978-3-642-29863-9_17)
- [29] IbisSoft, 2009. *iPB Reference Manual*. [Online] Available: <http://docs.ibissoft.se/node/3> [Accessed 10 Augustus 2013].
- [30] M. Khomyakov and I. Bider, "Achieving Workflow Flexibility through Taming the Chaos," *OOIS 2000 – 6th international conference on object oriented information systems*. Springer, pp. 85–92, 2000. Available: [https://doi.org/10.1007/978-1-4471-0299-1\\_7](https://doi.org/10.1007/978-1-4471-0299-1_7)

- [31] I. Bider, P. Johannesson, and R. Schmidt, "Experiences of Using Different Communication Styles in Business Process Support Systems with the Shared Spaces Architecture," *Proceedings of CAiSE 2011*. Springer, LNCS, vol. 6741, pp. 299–313, 2011. Available: [https://doi.org/10.1007/978-3-642-21640-4\\_23](https://doi.org/10.1007/978-3-642-21640-4_23)
- [32] Bider, I. and Perjons, E., 2018. Defining Transformational Patterns for Business Model Innovation. In *Perspectives in Business Informatics Research: 17th International Conference, BIR 2018, Stockholm, Sweden, LNBIP*. Springer, vol. 330, pp. 81–95. Available: [https://doi.org/10.1007/978-3-319-99951-7\\_6](https://doi.org/10.1007/978-3-319-99951-7_6)
- [33] V. McCune and D. Hounsell, "The development of students' ways of thinking and practising in three final-year biology courses," *Higher Education*, vol. 49, no. 3, pp. 255–289, 2005. Available: <https://doi.org/10.1007/s10734-004-6666-0>
- [34] G. Gowlland, "Apprenticeship as a Model for Learning in and Through Professional Practice," *International Handbook of Research in Professional and Practice-based Learning*. Springer, pp. 759–799, 2014. Available: [https://doi.org/10.1007/978-94-017-8902-8\\_28](https://doi.org/10.1007/978-94-017-8902-8_28)
- [35] I. Bider, M. Henkel, S. Kowalski, and E. Perjons, "Reuse of Simulated Cases in Teaching Enterprise Modelling," *Advances in Conceptual Modeling. Proceedings of ER 2015 workshops, LNCS*, vol. 9382, Springer, pp. 337–346, 2015. Available: [https://doi.org/10.1007/978-3-319-25747-1\\_33](https://doi.org/10.1007/978-3-319-25747-1_33)
- [36] I. Bider, E. Perjons, M. Elias, and P. Johannesson, "A fractal enterprise model and its application for business development," *Software & Systems Modeling*, vol. 16, no. 3, pp. 663–689, 2017. Available: <https://doi.org/10.1007/s10270-016-0554-9>
- [37] I. Bider and A. Chalak, "Evaluating Usefulness of a Fractal Enterprise Model Experience Report," *Enterprise, Business-Process and Information Systems Modeling. BPMDS 2019, EMMSAD 2019, LNBIP*, Springer, vol. 352, pp. 359–373, 2019. Available: [https://doi.org/10.1007/978-3-030-20618-5\\_24](https://doi.org/10.1007/978-3-030-20618-5_24)
- [38] M. Josefsson, K. Widman, and I. Bider, "Using the Process-Assets Framework for Creating a Holistic View over Process Documentation," *Enterprise, Business-Process and Information Systems Modeling, LNBIP*, vol. 214, Springer, pp. 169–183, 2015. Available: [https://doi.org/10.1007/978-3-319-19237-6\\_11](https://doi.org/10.1007/978-3-319-19237-6_11)
- [39] G. Koutsopoulos and I. Bider, "Business Process Canvas as a Process Model in a Nutshell," *Enterprise, Business-Process and Information Systems Modeling, LNBIP*, vol. 318, Springer, pp. 49–63, 2018. Available: [https://doi.org/10.1007/978-3-319-91704-7\\_4](https://doi.org/10.1007/978-3-319-91704-7_4)
- [40] G. Koutsopoulos and I. Bider, "Teaching and Learning State-Oriented Business Process Modeling. Experience Report," *Enterprise, Business-Process and Information Systems Modeling, LNBIP*, vol. 287, Springer, pp. 171–185, 2017. Available: [https://doi.org/10.1007/978-3-319-59466-8\\_11](https://doi.org/10.1007/978-3-319-59466-8_11)
- [41] M. Mumford, "The story of socio-technical design: reflections on its successes, failures and potential," *Information Systems Journal*, vol. 16, no. 4, pp. 317–342, 2006. Available: <https://doi.org/10.1111/j.1365-2575.2006.00221.x>
- [42] R. P. Bostrom and J. S. Heinen, "MIS problems and failures: A socio-technical perspective," *MIS Quarterly*, vol. 1, no. 3, pp. 17–32, 1977. Available: <https://doi.org/10.2307/248710>
- [43] S. Kowalski, *IT Insecurity: A Multi-disciplinary Inquiry*. Doctoral thesis. Stockholm, Sweden: Royal Institute of Technology, 1994.
- [44] I. Bider and S. Kowalski, "A framework for synchronizing human behavior, processes and support systems using a socio-technical approach," *BPMDS 2014 and EMMSAD 2014. LNBIP*, Springer, vol. 175, pp. 109–123, 2014. Available: [https://doi.org/10.1007/978-3-662-43745-2\\_8](https://doi.org/10.1007/978-3-662-43745-2_8)
- [45] I. Bider and O. Söderberg, "Becoming Agile in a Non-disruptive Way – Is It Possible?" *ICEIS 2016 – Proceedings of the 18th International Conference on Enterprise Information Systems*. SciTePress, pp. 294–305, 2016. Available: <https://doi.org/10.5220/0005785302940305>