# Heuristic Theorizing: Proactively Generating Design Theories

Robert Wayne Gregory
IESE Business School, University of Navarra, 08034, Barcelona, Spain, rwgregory@iese.edu

Jan Muntermann
University of Göttingen, 37073 Göttingen, Germany, muntermann@wiwi.uni-goettingen.de

Design theories provide explicit prescriptions, such as principles of form and function, for constructing an artifact that is designed to meet a set of defined requirements and solve a problem. Design theory generation is increasing in importance because of the increasing number and diversity of problems that require the participation and proactive involvement of academic researchers to build and test artifact-based solutions. However, we have little understanding of how design theories are generated. Drawing on key contributions by Herbert A. Simon, including the ideas of satisfice and bounded rationality and reviewing a large body of information systems and problem-solving literature, we develop a normative framework for proactive design theorizing based on the notion of heuristic theorizing. Heuristics are rules of thumb that provide a plausible aid in structuring the problem at hand or in searching for a satisficing artifact design. An example of a problem-structuring heuristic is problem decomposition and an example of an artifact design heuristic is analogical design. We define heuristic theorizing as the process of proactively generating design theory for prescriptive purposes from problem-solving experiences and prior theory by constantly iterating between the search for a satisficing problem solution, i.e., heuristic search, and the synthesis of new information that is generated during heuristic search, i.e., heuristic synthesis. Heuristic search involves alternating between structuring the problem at hand and generating new artifact design components, whereas heuristic synthesis involves different ways of thinking, including reflection and learning and forms of reasoning, that complement the use of heuristics for theorizing purposes. We illustrate the effectiveness of our heuristic theorizing framework through a detailed example of a multiyear design science research program in which we proactively generated a design theory for solving problems in the area of intelligent information management and so-called big data in the finance domain. We propose that heuristic theorizing is a useful alternative to established theorizing approaches, i.e., reasoning-based approaches. Heuristic theorizing is particularly relevant for proactive design theorizing, which emphasizes problem solving as being highly intertwined with theorizing, involves a greater variety of ways of thinking than other theorizing approaches, and assumes an engaged relationship between academics and practitioners.

*Keywords*: generating design theories; design science; proactive design theorizing; problem solving; heuristics; heuristic theorizing; heuristic search; heuristic synthesis; sciences of the artificial

*Design* [...] *is concerned with how things ought to be, with devising artifacts to attain goals.*

(Simon 1996, p. 114)

## 1. Introduction

Jan is an information systems (IS) scholar who performs proactive design science research (DSR) by drawing heavily on Simonion design logic to develop prescriptive problem solutions and construct design theory. A recent example of Jan's research in a large industry-academic consortium is in the area of intelligent information management in the finance domain. In this DSR program that involved multiple partners who work together for several years, Jan addressed the wicked problem of extracting and leveraging relevant information from massive heterogeneous data streams to support financial decision making. For this purpose, in cooperation with the consortium, Jan alternated in multiple iterations between structuring the problem and designing an instantiated artifact to find a satisficing solution to the previously described problem. Simultaneously, and intertwined with this iterative problem-solving process (i.e., heuristic search), Jan engaged in synthesizing the newly generated information from the recurrent use of heuristics (i.e., heuristic synthesis) to construct a design theory.

In the aftermath of this proactive DSR program, Jan was feeling satisfied about the research consortium's achievements. However, a key challenge remained about which he had reflected with Robert, a fellow IS scholar, already during the execution of this program. "How can I effectively explain that generating design theory and the heuristic search for a problem solution complemented one another in my research and actually went hand-in-hand?" Jan asked. Robert's initial reaction was as follows: "Your pragmatic approach seems atheoretical and messy to me. What does heuristic search and problem solving have to do with building theory?" Realizing that this type of reaction was not atypical among academics in fields such as IS and management, Robert and Jan recognized three important facts: (1) we lack an understanding of how design theories are generated and constructed; (2) this gap in our understanding prevents scholarly communities from validating and embracing the generation of design theories; and (3) improving our understanding of how design theories are generated is important because the engaged process and the prescriptive outcomes of generating design theories may have highly relevant impacts on practitioners, which should be a concern to scholars across disciplines.

This paper's purpose is to develop a normative framework for proactive design theorizing. We use "proactive design theorizing" to denote the intertwined act of problem solving and theorizing that we suggest as being an integral part of generating a design theory for prescriptive purposes. Furthermore, "proactive" assumes an engaged academic-practitioner research relationship. Our framework is inspired by the works of Herbert A. Simon, a highly influential scholar in the design science tradition and across a number of disciplines (e.g., management, economics, computer science, and artificial intelligence).

A design theory provides "explicit prescriptions (e.g., methods, techniques, principles of form and function) for constructing an artifact" (Gregor 2006, p. 620). Artifacts and more abstract design theories focus on problem solving and relate design or solution components to a set of defined problem requirements (Simon 1996, Walls et al. 1992). An example of constructing artifacts and generating a design theory is presented by the study from Markus et al. (2002) on systems that support emergent knowledge processes. This study closely resembles the example in our opening vignette. Both examples highlight the generative character of proactive design theorizing. That is, such theorizing involves recurrently generating and testing new solution components, which goes hand-in-hand with generating design theory through synthesizing the wealth of information that is produced through multiple iterations of heuristic search.

The value of the framework that we develop in this paper is that it (1) incorporates multiple heuristics (i.e., rules of thumb that provide a plausible aid in structuring the problem at hand or in searching for a satisficing artifact design) and (2) embraces the highly iterative nature of generating design theories that is intertwined with the heuristic search, thus suggesting that generating design theories involves alternating between problem structuring and artifact design. Both (1) and (2) have not been fully acknowledged in prior research on design theorizing.

Regarding (1), we adopt Simon's suggestion to expand on the idea of heuristics for explaining proactive design theorizing. Our heuristic theorizing framework includes two types of heuristics: heuristics for problem structuring and for artifact design. Prior studies have most often adopted a descriptive—as opposed to proactive—approach (Gregor and Jones 2007). However, this is only one way (not necessarily the best way), to develop design theory (Iivari 2007). Iivari (2014) discusses the similarities and differences between these two DSR approaches, referred to as "strategy 1" (i.e., descriptive) and "strategy 2" (i.e., proactive). As Iivari explains, the example and research method offered by Sein et al. (2011) marks an important step toward understanding proactive DSR, whereas design theorizing in this cumulative tradition is still not well understood.

Regarding (2), our heuristic theorizing framework suggests that proactive design theorists iterate back and forth between the use of problem-structuring heuristics and artifact design heuristics to generate design theory, which prior studies have not fully acknowledged. In this paper, this iterative problem solving and design process is referred to as *heuristic search*. Through heuristic search, new information is recurrently generated that must be integrated in and related to the different components of a design theory, which we refer to as *heuristic synthesis*.

Prior studies have sought to explain design theorizing through the idea of reasoning (e.g., Gregor and Jones 2007, Gregory and Muntermann 2011, Kuechler and Vaishnavi 2008). Reasoning has been defined as the "systematic process of drawing inferences (conclusions) from some initial information (premises)" (Holyoak and Morrison 2005, p. 2). Prominent examples include induction, deduction, and abduction. However, a long time ago, Simon (1996) stated, "We might question whether the forms of reasoning that are appropriate to natural science are suitable also for design" (p. 114). It can be argued that proactive design theorizing cannot be fully explained by relying exclusively on the idea of reasoning. In this paper, we contend that traditional forms of reasoning may play a role in heuristic synthesis, and other ways of thinking may also be involved.

In sum, our heuristic theorizing framework is based on a review of prior literature and integrates our cumulative understanding of design theorizing. The framework is particularly relevant for proactive design theorists who focus on the generation of design theory for prescriptive purposes in a pragmatic fashion, often addressing difficult or even wicked problems through collaboration with practitioners. There are many proactive design science researchers in the international IS community. However, we believe that our framework is useful across disciplines. For these theorists, our framework has important implications: It provides a useful toolkit for systematizing and communicating their theorizing approach for generating design theory. To illustrate our framework's effectiveness, we provide a detailed example of a multiyear DSR program in which a design theory was generated through heuristic search and heuristic synthesis.

We proceed by providing general background knowledge about theorizing (§2) and design science (§3). In the main part of the paper, we develop our normative framework (§4), which we subsequently illustrate with the help of a DSR program showcase (§5). Finally, we discuss and compare heuristic theorizing with other approaches (§6) and call for more proactive design theorizing that draws on our framework (§7).

## 2. Background: Theorizing

"The process of theorizing consists of activities such as abstracting, generalizing, relating, selecting, explaining, synthesizing, and idealizing" (Weick 1995, p. 389). Theorizing is important for producing generalizable contributions to knowledge (Lee and Baskerville 2003). In the social and organization sciences, theorizing is associated with reasoning. Reasoning has been defined as the "process of drawing inferences (conclusions) from some initial information (premises)" and is viewed as a key type of the thinking that serves to accumulate scientific knowledge (Holyoak and Morrison 2005, p. 2). The two most well-established forms of reasoning are induction and deduction: "The unique contribution of science lies in its combination of deductive and inductive methods for the development of reliable knowledge" (Thompson 1956, p. 102). Induction has been defined as reasoning that "leads, by means of mechanically applicable rules, from observed facts to corresponding general principles" (Hempel 1966, p. 14). Samuels (2000) describes induction as "going from particulars to generals; deriving knowledge from empirical experience" (p. 214). In contrast, deductive reasoning focuses on constructing deductive arguments based on premises or hypotheses (Hempel 1966), which according to Samuels (2000) involves "going from generals to particulars; deriving conclusions based on premises through the use of a system of logic" (p. 214). Based on this foundation, theorizing in management, organization, and social science is often characterized as inductive, deductive, or both (Shepherd and Sutcliffe 2011).

Proactive design theorizing diverges from the well-known inductive and deductive approaches. It resembles to a greater extent the ways of thinking involved in abduction. Abduction is attributed to the pragmatist problem solving of Peirce (Aliseda 2006, Peirce 1974) and is typically described as a creative and intuitive form of reasoning (Cross 2006, Holmström et al. 2009, Martin 2010). The way of thinking involved in abduction has been described as assuming the form of guessing and experimenting with alternative tentative ideas. Thus, the way of thinking involved in abduction approaches the manner in which design theorists think about using the intertwined process of problem solving and design theorizing. However, similar to induction and deduction, abduction focuses too much on explanations, not design and prescriptions.

In sum, we argue in this paper that proactive design theorizing involves problem solving through artifact design and thus cannot be fully understood in terms of the classical forms of reasoning from the social and organizational sciences.

## 3. Background: Design Science

Simon established the foundation of design science by emphasizing the uniqueness of the sciences of the artificial (Simon 1996). As described in his seminal research, the sciences of the artificial focus on the design of artifacts that serve a human purpose. In IS, DSR focuses on constructing IT artifacts, such as constructs (e.g., specific data modeling formalisms), models (e.g., a set of interrelated data modeling formalisms), methods (e.g., a data modeling language), and instantiations (e.g., the realization of a data modeling model in an environment) (March and Smith 1995). In addition, more abstract artifacts, including design principles and design theory, are developed (Markus et al. 2002, Walls et al. 1992). In the following, we discuss the nature of artifacts and design theories. In addition, we clarify the key paradigmatic assumptions of our heuristic theorizing framework and introduce the idea of heuristic.

### 3.1. Artifact

"For the sciences of the artificial, the first and foremost requirement of knowledge is its efficiency and effectiveness for bringing into existence an artifact needed to solve a given problem, achieve a given goal, or otherwise fulfill a given need that is facing people in the real world" (Lee 2010, p. 346). Simon conceptualized an artifact as a human, man-made product or process that "can be thought of as a meeting point—an 'interface' in today's terms—between an 'inner' environment, the substance and organization of the artifact itself, and an 'outer' environment, the surroundings in which it operates" (Simon 1996, p. 6). The focus on artificially constructed products or processes led Simon to create a close link between the "sciences of the artificial" and design. He emphasizes that the designer "is concerned with how things *ought* to be" (Simon 1996, p. 4) and is focused on prescription, i.e., finding a "way in which that adaptation of means to environments is brought about" (Simon 1988, p. 68) until a satisficing, functioning problem solution is found. Explanatory and predictive knowledge, for example from the social and natural sciences, provides the rationale for the prescription and elucidates, for instance, why a certain prescription is made, what elements it consists of, and how it can be implemented (Walls et al. 1992). In sum, an artifact is a human product that prescribes something based on a certain rationale to attain goals and to function by relating an artifact design to artifact requirements (Simon 1996).

### 3.2. Design Theory

Important advances have been made over the years toward understanding the nature and anatomy of design theories. Dasgupta (1991) states that "the designer begins with a *problem*. The act of design, thus, starts with a description of the problem stated in terms of a set of *requirements*. The designer's brief is to develop a *form*—what we conventionally call "the design"—such that if an artifact is built according to this form (that is, if we were to *implement the design*) it will *meet* the requirements." (p. 13). Expanding this idea of design problems, Walls et al. (1992) argue that two key components of design theories are (a) meta-requirements and (b) meta-solution components.

Gregor and Jones (2007) characterize design theories as having six core components: a specification of (1) purpose and scope (i.e., meta-requirements, goals, outer environment), (2) constructs (i.e., underlying design ideas), (3) principles of form and function (i.e., the problem solution, the inner environment, the form), (4) artifact mutability (i.e., anticipated changes to the artifact), (5) testable propositions, and (6) justificatory knowledge. Additionally, they mention (7) principles of implementation and (8) the material instantiation as optional components. Thus, (1)–(3) of Gregor and Jones's (2007) components expands Dasgupta (1991) and Walls et al. (1992). Including further components, such as "justificatory knowledge," in the anatomy of a design theory agree with Simon's view that the construction of artifacts involves drawing on knowledge that then becomes an integral part of the product or process (Simon 1996).

### 3.3. Paradigmatic Assumptions

As previously explained, artifacts and design theories emphasize prescription. To achieve the previously noted primary purposes of prescription, an artifact must "attain goals" and it must "function" (Simon 1996, pp. 4–5), which has been rephrased in IS DSR as "solving identified organizational problems" through an IS research cycle that "creates and evaluates IT artifacts" intended for these purposes (Hevner et al. 2004, p. 77). Addressing and solving problems in practical reality is a defining element of the applied sciences, which emphasizes a pragmatic viewpoint. For example, Briggs et al. (2011) view applied science research in IS as "the last research mile." "The last research mile begins when a research team finds real people with a real problem in a real organization. They explore the problem, learn about stakeholder goals, and seek to discover drivers and constraints in the problem environment. They propose possible solutions to stakeholders and listen carefully to their responses" (p. 14).

The proactive and pragmatic design scientist is primarily concerned with such criteria as purposefulness, functionality, utility, and value (Niiniluoto 1993). Simon described this concern as searching for a problem solution that is "good enough" in terms of meeting the problem requirements, and he termed this solution *satisficing* (Simon 1996, p. 27). Simon also argued that human problem solvers have limited cognitive resources and that *bounded rationality* implies a focus on finding satisficing solutions because the "best" solutions cannot be found anyway. This view is particularly true in many complex DSR programs, in which the problems at hand are typically considered to be *wicked*, meaning that they lack a definitive formulation, conflicting perspectives may be involved, and it is uncertain whether a solution will be found (Buchanan 1992, Churchman 1967, Rittel 1972). In sum, the goal-oriented focus of design science lends itself to an instrumentalist problem-solving perspective with its truth-independent focus on usefulness (Kilduff et al. 2011).

### 3.4. Heuristics

Heuristics have been identified as a central idea in such diverse areas as mathematics (Pólya 1945), human problem solving (Newell and Simon 1972, Simon 1978, Simon and Newell 1958), decision making (Simon 1967), architectural design (Alexander 1964, Rowe 1987), computer science (Pearl 1983), engineering design (Pahl et al. 2007), design thinking (Cross 2008, Martin 2010), artificial intelligence (Feigenbaum and Feldman 1963), knowledge management systems (Trice and Davis 1993), and management (Smith 1988, 1992). Heuristics are defined in different ways in the literature: methods of "generating ideas and finding solutions" (Pahl et al. 2007, p. 53), "any principle, procedure, or other device that contributes to reduction in the search for a satisfactory solution" (Rowe 1987, p. 75), "a rule of thumb that often helps in solving a certain class of problems, but makes no guarantees" (Perkins 1981, p. 192), methods "serving to discover" (Pólya 1945, p. 113), "methods and criteria for judging the relative merits of alternative courses of planning or action" (Pearl 1983, p. 23), and "methods of finding out or knowing something" (Kinney 1979, p. 352). Many of these definitions originate in disciplines that focus on problem solving. Therefore, unsurprisingly, Simon once proposed adopting a heuristic perspective for design science (Simon 1996). In this paper, we define *heuristic* as a rule of thumb that provides a plausible aid in structuring the problem at hand or in searching for a satisficing artifact design.

### 3.5. Problem Solving

In addition to expanding ideas of artifact, design theory, satisficing, and other elements reviewed above, the development of our framework was inspired by Simon's perspective on the logic of artifact design. Simon described artifact design as a problem-solving process. This process involves the search for purposeful solution alternatives or components that satisfy certain requirements with the goal of finding a satisficing solution. According to a theory of human problem solving of Simon and his coauthors (Newell et al. 1958, 1967; Newell and Simon 1972; Simon 1978), problem solving begins with an initial construction of an adequate internal representation (i.e., understanding, definition) of a problem. The pieces of information required for this construction include (1) the initial situation (i.e., the givens), (2) the goals, and (3) the operators or operations, which are the procedures or so-called moves used to transform elements of the problem representation (Wickelgren 1974). Together, these pieces of information form the problem state; i.e., "the state of the world of a problem, is the set of all the expressions [i.e., givens, goals, operations] that exist in the world of the problem at a particular time" (Wickelgren 1974, p. 15). This world of the problem is also frequently referred to as the problem space (Newell and Simon 1972, Rowe 1987). Problem solving is then defined as "a process in which the problem solver searches through the problem space to find a solution path" (Hayes 1978, p. 183). Newell and Simon (1972) depict the search process itself (known as the heuristic search) as a cycle of continuously applying operators to the current element of the problem, replacing or transforming this element to advance in the search, and then either abandoning the current solution path in case of success to address other elements of the problem or engaging in further iterations. With this Simonion view in mind, we continue with the development of a normative framework for proactive design theorizing.

## 4. Heuristic Theorizing

In this section, we introduce and explain our normative framework for proactive design theorizing, which is based on the notion of heuristic theorizing.

### 4.1. Heuristic Theorizing Framework

Our notion of heuristic theorizing involves the core idea that the heuristic search for a satisficing problem solution, which involves alternating between structuring the problem at hand and generating new artifact design components, is tightly intertwined with the heuristic synthesis of information about artifact requirements and solution components and their prescriptive relationships (Figure 1). Thus, problem solving is an integral part of proactive design theorizing.

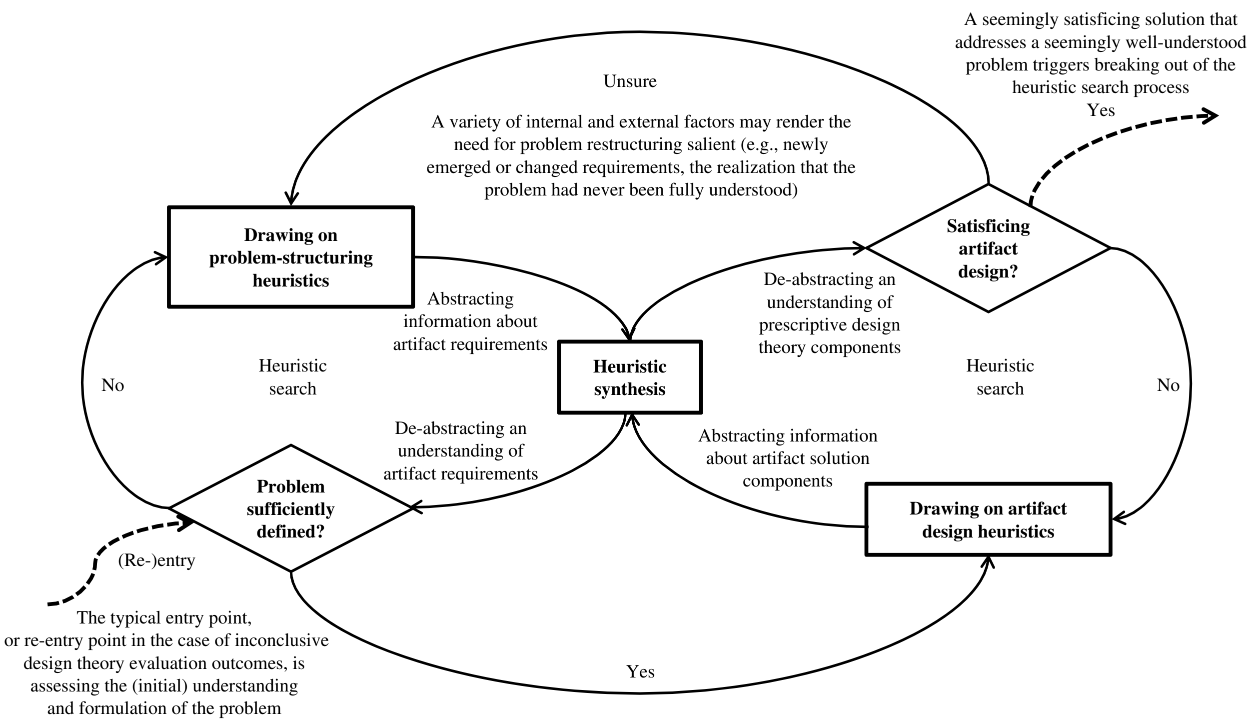
To explain our framework in detail, we discuss heuristics for problem structuring and artifact design (§§4.2.1 and 4.2.2, respectively). Subsequently, we explain heuristic search (§4.3) and heuristic synthesis (§4.4). Finally, we briefly discuss the nature of concurrent evaluation in proactive design theorizing (§4.5). Together, the concepts of heuristics, heuristic search, heuristic synthesis, and concurrent evaluation explain how design theories are proactively generated.

### 4.2. Two Types of Heuristics: The Foundation of Heuristic Search

**4.2.1. Problem-Structuring Heuristics.** One type of heuristic is focused on structuring the problem at hand. In proactive design science work, this structuring is an essential ingredient of theorizing because an artifact design addresses certain problem requirements that must be understood and artifact requirements are a core component of a design theory (see §3). Structuring a so-called wicked problem, which is a typical situation in proactive design theorizing, can be an arduous task and requires as much attention as the design of solution components and an artifact. Smith (1988) provides a useful introduction to problem structuring: structuring a problem is typically a core element of problem solving. However, it is also clear that the goal is not to find the "best" possible way of structuring the problem at hand. Frequently, it may be impossible to find an optimal problem structure and definition. Instead, the goal is to find a tentative structuring of the problem at hand with a working definition and formulation of the problem, which provides sufficient direction during a heuristic search for a problem solution or artifact design.

In sum, one element of heuristic search is iteratively drawing on problem structuring heuristics. Table 1 provides an overview of examples of heuristics for problem structuring.

**Figure 1** **Heuristic Theorizing Framework**




**4.2.2. Artifact Design Heuristics.** The other type of heuristic is concerned with finding working solution components and a satisficing artifact design. The primary goal is to iteratively generate new solution component candidates, to assess to what extent these candidates prescriptively relate to particular artifact requirements, and to gradually reduce the differences between the nascent artifact design and the tentatively formulated artifact requirements. Simon has referred in his research to generate-and-test procedures and means-ends analysis, which are the general ideas behind artifact design heuristics (Simon 1996). In IS research, the "generator-test cycle" has also been referred to for understanding DSR as a search process (Hevner et al. 2004).

In sum, another element of heuristic search is iteratively drawing on artifact design heuristics, which has a generative character. Table 2 provides examples of artifact design heuristics.

### 4.3. Heuristic Search

The foundation for heuristic search, i.e., the two different types of heuristics, was discussed previously.

**Table 1** **Examples of Problem-Structuring Heuristics**

| Heuristic | Definition/meaning | Example from the DSR literature |
|---|---|---|
| Decomposing the problem | Decomposing a complex problem into less complex subproblems involves subdividing the problem into sets of simpler problems and attacking these individually (Simon 1973, Simon et al. 1981). | Ow and Smith (1987) develop two design principles for the design of knowledge-based systems. The initial job-shop scheduling problem is decomposed into two subproblems (see Figure 2) that are addressed individually by different approaches. |
| Problem class identification | Problem class identification involves relating the problem at hand to a higher-order class of problems of which the given problem is just one instance (Vaishnavi and Kuechler 2008). | Lee et al. (2011) discuss an example from the IS literature (Pries-Heje and Baskerville 2008, Pries-Heje and Vinter 2006) that involve the design of concrete organizational change initiatives in two companies and an instantiation of a decision support system for the selection of an organizational change and user involvement approach based on spreadsheet software. The design theorists in this example identified the problem class as multicriteria decision making. |
| Reformulating the problem | Reformulation of a problem involves altering one's definition of the problem and assessing alternative definitions (Smith 1988). | Lindgren et al. (2004) begin their research program by defining their problem as the inaccuracy and incompleteness of competence data used to manage competencies in organizations. Eventually, the research team reformulated the problem as the adoption of a skills-based paradigm (as opposed to a job-based paradigm) for competency management systems. |

**Table 2 Examples of Artifact Design Heuristics**

| Heuristic | Definition/meaning | Example from the DSR literature |
|---|---|---|
| Analogical design | Analogical design is defined by Goel (1997) as the "reminding and transfer of knowledge about one design situation to another, where the transfer can occur in the service of any design task in the new situation" (p. 63). | Kärkkäinen and Holmström (2002) generated the idea of transferring knowledge about wireless product identification technology, or radio-frequency identification (RFID), from past design situations (e.g., manufacturing and warehousing) to the problem of item-level supply chain management. |
| Ideation and prototyping | Generating a multitude of ideas and embedding them in prototypes is common in design. Prototypes "should command only as much time, effort, and investment as are needed to generate useful feedback and evolve an idea" (Brown 2008, p. 3). | Majchrzak and Gasser (2000) engaged in over 70 prototyping iterations over a time period of approximately four years in developing the TOP-MODELER solution, which consists of a modular knowledge base and an analytical method to address the problem class of emergent knowledge processes (Markus et al. 2002). |
| Playing with kernel theories | Kernel theories embody design principles and knowledge from past solutions or contain knowledge from the natural or social sciences (Walls et al. 1992). Design theorists "play" with such theories to find solutions (Kilduff et al. 2011). | Research on managerial information scanning and emerging issue tracking as well as theories of open loop control are synthesized to generate ideas how to address the problem class of vigilant information system design (Walls et al. 1992). |
| Modeling | Modeling entails creating and experimenting with different types of representation of the problem solution to graphically, conceptually, or technically capture artifact solution components (Rowe 1987, Smith 1992). | Müller-Wienbergen et al. (2011) develop a design theory for IT systems that supports convergent and divergent thinking and is grounded in theory on human cognition and the creativity support literature. For artifact design, the authors explain guidance using an abstract blueprint, and they graphically depict a conceptual solution schema (see Figure 1). |

In the following, we explain the remaining elements of heuristic search according to Figure 1.

*Entering the heuristic cycle*. Typically, a heuristic search starts with an initial description and understanding of the problem at hand. Proactive design theorizing involves asking "Is the problem sufficiently defined?" If the answer is "no," which is not atypical because of different stakeholders with diverging interests and the complexity of problems typically addressed in proactive design theorizing, it is recommended to draw on problem structuring heuristics until the question can be answered "yes." Multiple problem-structuring iterations may be required, which involve various problem-structuring heuristics.

*Switching from problem structuring to artifact design*. When the problem at hand is perceived to be sufficiently defined based on the available cognitive capacities and bounded rationality of the design theorists, artifact design heuristics are used to generate tentative solution components that match the artifact requirements identified during problem structuring. Similar to the way in which problem-structuring heuristics recurrently produce new information about artifact requirements across heuristic search iterations, artifact design heuristics produce new information about artifact solution components that address particular artifact requirements. Across the iterations that draw on artifact design heuristics, proactive design theorists ask "Have we generated a satisficing artifact design?" If the answer is "no" and the problem's current formulation remains stable, theorists iterate while alternating between the selection of any useful artifact design heuristic, its application to generate tentative artifact solution components together with new information, and reassessing to what extent artifact solution components have been found to address the given artifact requirements.

*Returning to problem restructuring*. A variety of internal and external factors may render the need for problem restructuring salient, which may cause the temporary abandonment of artifact design and a return to structuring the problem at hand. For example, requirements change over time, or entirely new requirements can suddenly emerge as a result of external factors that are not under the control of researchers, such as a change in the industry-academic project consortium (if there is one), the research team, or important changes in the technology and problem environment. The need to return from artifact design to problem structuring may also become salient through internal factors, such as theorists realizing in the process of searching for artifact solution components that the problem at hand was never fully understood. In proactive design theorizing, it is common that the generation of artifact solution components and the search for a satisficing artifact design begins with a limited initial understanding of the problem at hand. That is, finding a tentative solution to a problem that is not yet fully understood may also play a role in defining the problem itself.

*Breaking out (and potentially re-entering)*. A seemingly satisficing solution in the form and function of an artifact design that addresses a seemingly well-formulated problem and the associated artifact requirements facilitates breaking out of the heuristic search cycle (Figure 1). This breaking out occurs when theorists answer the question "Have we identified a satisficing artifact design?" with a tentative "yes." The answer to this question is tentative at the stage of breaking out of the cycle because proactive design theorizing typically involves gathering feedback from those who experience the problem in practice to determine whether the proposed design solution functions and is useful.

### 4.4. Heuristic Synthesis

As depicted in Figure 1 and previously explained, new information about artifact requirements and solution components is recurrently generated across iterations of heuristic search. Depending on the complexity of the problem being solved and the significance of the research project or program, a substantial amount of information is continuously generated during heuristic search. It may be challenging to select relevant pieces of information, abstract and relate these pieces of information to individual design theory components, and integrate them into a whole. In this paper, we refer to this process as heuristic synthesis.

Heuristic synthesis is an ongoing process that is closely intertwined with heuristic search (Figure 1). Heuristic search involves the recurrent selection and use of heuristics and the generation of new information, whereas the intertwined process of heuristic synthesis involves different ways of thinking for extracting relevant and theoretical insights from this large amount of information.

Heuristic synthesis is required for two related matters. First, heuristic synthesis is the essential link between heuristic search/problem solving and abstracting to a design theory. Second, heuristic synthesis links an iteration of heuristic search to the next iteration. These two linkages distinguish heuristic theorizing from heuristic problem solving, routine design, and system building.

Heuristic synthesis may involve a variety of different ways of thinking that complement the use of heuristics to extract theoretical insights from heuristic search (the link from heuristic search to heuristic synthesis) and apply synthesized theoretical insights to problem solving (the link from heuristic synthesis to heuristic search).

In iteration that alternates between heuristic search and heuristic synthesis, a key mechanism is cycling back and forth between abstraction (from heuristic search to heuristic synthesis) and de-abstraction (from heuristic synthesis to heuristic search) (Lee et al. 2011). Abstraction involves "leaving out specific details about where and when it [the information] originated, as well as other ideas that may have initially accompanied it" (Laurence and Margolis 2012, p. 3), and de-abstraction involves "the capacity of thinking the particular as contained under the universal . . . by descending from the universal to the particular" (Teufel 2012, p. 302).

Heuristic synthesis involves different ways of thinking for which Kuechler and Vaishnavi (2012) provide several suggestions (Appendix C-1 in the paper). Accordingly, traditional forms of reasoning, i.e., induction, deduction, and abduction, may play a role in heuristic synthesis. However, a key way of thinking for heuristic synthesis is "reflection and learning" (Sein et al. 2011). Reflection and learning involves "thinking back on what we have done to discover how our knowing-in-action may have contributed to an unexpected outcome" (Schön 1987, p. 26) and is based on the assumption that "our knowing is in our action" (Schön 1983, p. 49).

### 4.5. Concurrent Evaluation

Heuristic theorizing emphasizes the generation, as opposed to the final evaluation, of (nascent) design theory. Under the assumption that the selection and use of heuristics is performed systematically according to our framework (Figure 1), the depth and breadth of information that is generated through this heuristic search should provide a reasonably good foundation for generating (nascent) design theory. A formal, final evaluation of the emerged design theory, as suggested in previous DSR studies (Gregor and Hevner 2013), is unnecessary when our normative framework for proactive design theorizing is applied. Similar to other previous DSR studies (Sein et al. 2011), we suggest that the evaluation of the emerging artifact and (nascent) design theory should occur concurrently with the heuristic search and by breaking out of and re-entering the heuristic search cycle multiple times during the design theorizing process. It is important to ensure that the artifact design "works" in the eyes of those who experience the problem at hand in practice.

## 5. Illustration of the Framework: A DSR Program Showcase

### 5.1. Introduction to the Showcase and Theorizing Process

This study's second author was a principal researcher and work-package leader in a research program funded by the European Commission within the Seventh Framework Programme, in particular, within the Information and Communication Technology (ICT) research program. The DSR program was funded between 2010 and 2013 with a grant of approximately three million euros for three years. The general focus was intelligent information management in the finance domain, in particular the design of an artifact for the problem class of extracting relevant information from massive heterogeneous data streams to support financial decision making. The European multinational research consortium consisted of four distinct groups: (1) various research institutions (including universities), (2) IT services providers, (3) a bank, and (4) a stock exchange. This study's second author was involved from the initial generation of the project plan and the establishment of the consortium to the full implementation of the associated work plan. The first author joined the research program as an external observer to jointly reflect on the heuristic search for a satisficing solution and theorizing in DSR programs based on the experiences extracted from this case. In the following, we illustrate the iterative nature of drawing on problem structuring (SH) and artifact design (DH) heuristics (i.e., heuristic search). In addition, we illustrate how heuristic search in our case recurrently resulted in the generation of new information, which we synthesized to construct a design theory (i.e., heuristic synthesis).

### 5.2. Initial Understanding and a Description of the Problem

A basic motivation for the discussed DSR program was the increased availability of large amounts of unstructured (e.g., public Web content) data streams that may be used in the finance domain but that are not yet fully leveraged to support financial decision making. One example is the domain of market surveillance, in which there is a great need to develop automated techniques to better detect and prevent information-based market manipulation that involves the dissemination of false information in diverse media (e.g., social media) to manipulate share prices (Allen and Gale 1992). Based on a number of such concrete use cases defined in close collaboration with industry partners, the general class of problems was initially described as the "intelligent management of big data streams for financial decision support" that involves a number of meta-requirements, i.e., effectively addressing (1) data heterogeneity, (2) data quality (e.g., unreliability, noise), (3) big data stream processing in real time, (4) relevant information extraction, and (5) decision-making support in the context of market surveillance. The previously explained purpose and scope of the developed design theory (DT1) and the initial meta-requirements were formulated by reflecting upon use case requirements. These use case requirements had initially been specified by the industry partners in collaboration with the researchers.[1] In the following subsection, we reflect on the heuristic theorizing process of developing the additional components of a related design theory (illustrated in Figure 2).

### 5.3. Heuristic Search and Synthesis

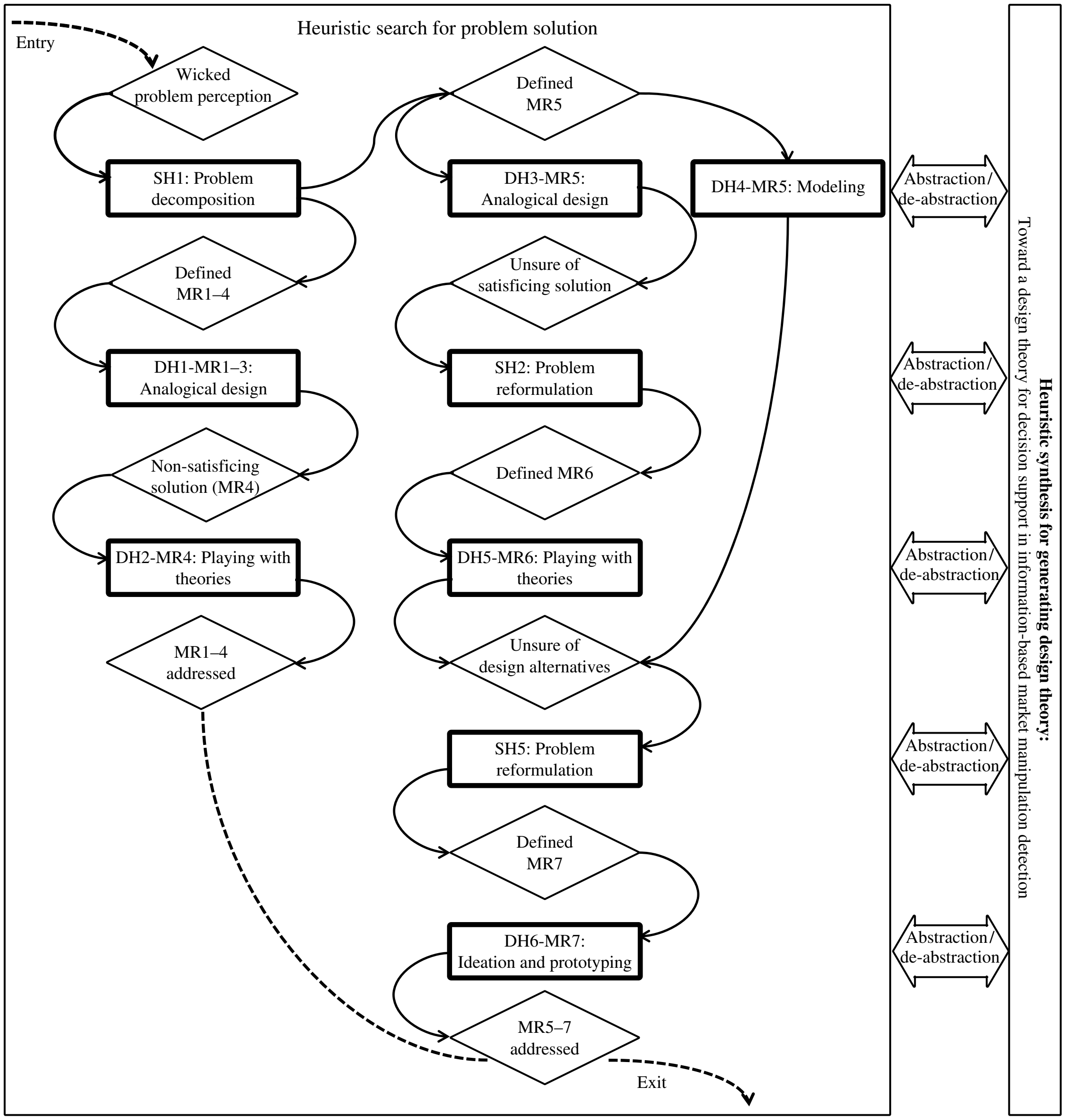
Heuristic search started with drawing on problem structuring heuristics. The addressed problem class appeared wicked in the sense that the overall complexity was perceived to be so high that starting to search for a problem solution was unreasonable. Instead, a problem-structuring heuristic, i.e., problem decomposition (SH1), helped decompose the problem class into two manageable smaller ones. On one hand, the subproblem of "data acquisition and information extraction from heterogeneous data streams" was identified, which is closely related to meta-requirements (1)–(4). On the other hand, the subproblem of "decision support components for market surveillance leveraging this extracted information" was formed, which addresses meta-requirement (5).

Within the first subproblem, heuristic search started by addressing the first meta-requirement (MR1), i.e., data heterogeneity, which results from the diversity of sources from which data were acquired in the context of this program (i.e., 160 websites, including financial portals, blogs, and forums). An artifact design heuristic, i.e., analogical design (DH1-MR1), yielded the idea to leverage the benefits using a Web standard (i.e., Rich Site Summary (RSS)), which proved to be useful for aggregating online data from different sources in a recurrent and timely manner via a standardized interface (i.e., an RSS feed).[2] In our program, 4,800 RSS feeds were subscribed to acquire data from 160 websites. As our first act of heuristic synthesis, reflection and learning from this approach resulted in our adding "Web feeds" as a relevant construct to our design theory (DT2). The second meta-requirement (MR2), i.e., data quality, resulted from the program's need to deal with user-generated Web content provided by nonprofessional content providers. Through analogical design (DH1-MR2), the idea was generated to apply technologies, such as boilerplate removal approaches (Kohlschütter et al. 2010) to remove undesired Web content (e.g., navigation bars, advertisements) and duplicate detection (Theobald et al. 2008) to remove redundant data.[2] Again, by reflecting on these insights from the use of a design heuristic, we realized during heuristic synthesis that "boilerplate" should be added as a construct to our design theory (DT2). The third meta-requirement (MR3), i.e., big

[1] FIRST Consortium D1.2 Use case requirements specification, http://project-first.eu/public_deliverables; D2.1 Technical requirements and state-of-the-art, http://project-first.eu/public_deliverables.

[2] FIRST Consortium D4.2 Semantic information extraction components, addressing noise and uncertainty, http://project-first.eu/public_deliverables.

**Figure 2 Heuristic Theorizing in Our Showcase**




data stream processing in real time, emerged because of increasing data volumes that are continuously generated by diverse Web media, including social media. Expanding on analogical design (DH1-MR3) from the field of data processing, technologies such as pipelining and parallelization (Rong et al. 2007) were used to generate design solution components.[2] Again, we reflected on these insights and concluded that "pipelining" and "parallelization" represent constructs of our design theory (DT2).

The fourth meta-requirement (MR4), i.e., relevant information extraction (to be used within the second subproblem: "decision support components leveraging this extracted information") was a foundational motivation of the program that is closely associated with the European Commission's objective to promote research in the area of big data and related intelligent information management. Using abduction, the idea was generated to use theoretical advances in finance, i.e., behavioral finance, which acknowledges that many investors trade on noise rather than information. As Shleifer (2000) states, "Investors follow the advice of financial gurus" (p. 10). Based on this idea, an artifact design heuristic, i.e., playing with kernel theories (DH2-MR4), was used to apply sentiment analysis to the context of market manipulation.[3] During subsequent heuristic synthesis, "sentiment" was deductively derived from theory and added as an emerging construct to the design theory (DT2).

[3] FIRST Consortium D6.1 Models and visualisation for financial decision making, http://project-first.eu/public_deliverables.

The newly added construct helped to extract relevant information about artificially created price bubbles that may provide an indication for information-based market manipulation. In addition, we concluded that behavioral finance theory that explains how investors form their beliefs represented justificatory knowledge that we should add to our design theory (DT6).

The interim result that addressed the first identified subproblem by generating solution components for the meta-requirements (MR1–MR4) formed a basic infrastructure for intelligent information management in the context of big data streams. Expanding on this foundation, the program entered the stage of addressing the second subproblem, i.e., the fifth meta-requirement (MR5) of decision-making support in the context of market surveillance. Decision support should be provided to identify potential cases of information-based market manipulation by developing classifiers and models that build on the information extracted through the generated infrastructure.

To address the second subproblem, two alternative suggestions for design solutions were generated in parallel to address the fifth meta-requirement (the first attempt to present a satisficing problem solution). First, expanding on the fraud-detection literature (Ngai et al. 2011), analogical design (DH3-MR5) yielded the idea to train classifiers to identify suspicious (potentially manipulative) documents using support vector machines (SVM). Second, based on the extensive experience of a participating senior researcher, the idea was generated to apply qualitative multiattribute modeling (DH4-MR5) to address the same requirement. In contrast to the first solution proposal based on machine-learning techniques, this method integrated expert domain knowledge and resulted in easily comprehensible models.

In the course of generating the first tentative solution (explained above), it became apparent that a bag-of-words model (trained with SVM) may become vulnerable to the countermeasures of scammers. In particular, scammers could gain an understanding of the model's inner logic and use countermeasures, such as replacing particular words by suitable synonyms not recognized by the model. Thus, the researchers in the program recognized the need to draw again on problem-structuring heuristics. Through problem reformulation (SH2), it became apparent that a missing meta-requirement needed to be addressed. Based on this new insight, the additional meta-requirement (MR6) that the developed models must be robust against the potential countermeasures of scammers was identified and added to our design theory (DT1).

After gaining a better understanding of the meta-requirements, the researchers started again to search for a problem solution. Using abduction, the idea was generated to enhance the bag-of-words model using marketing theory that explains particularly effective marketing communication (Clark et al. 1990). This idea resulted in the use of artifact design heuristics (DH5-MR6), in particular the extension of the previously described bag-of-words model based on deducing additional features from the marketing literature for use in model construction. These additional features increased the model's robustness, which was again trained with SVM.[4] From the insight we gained during the problem reformulation and artifact design, we learned that our theory must anticipate artifact change that involves adjusted manipulation schemes (DT4).

While the two alternative market manipulation detection models were recurrently being refined and tested through iterations with industry domain experts (a second attempt to transition to the "solved" stage), feedback was received that regulatory authorities demand glass-box models be easily interpretable by users. As stated by Van Eyden (1996), machine-learning techniques, such as SVMs and neural networks, "cannot justify their answers" (p. 73). That is, the resulting classifiers represent black-box models. Because the previously described ideas for model development were primarily based on such machine-learning techniques (i.e., black-box models), the need to restructure the problem was recognized again. As before, by drawing on problem reformulation (SH5), a new meta-requirement (MR7) was defined and added to our design theory (DT1) that emphasized the need to ensure that end users understand the internal workings of the decision-support components of an instantiated decision-support system.

Therefore, the subsequent search for problem solutions and related artifact design was based more strongly on the previously described idea of qualitative multiattribute modeling. The idea was generated to use the transformed outputs of the developed SVM-trained black-box model as an additional input for the qualitative multiattribute model. Through ideation and prototyping (DH6-MR7), both models were embedded in a prototype with the black-box model generating input for the glass-box model.[5] From this insight into the combination of different methods for model development, a design principle

[4] FIRST Consortium D6.3 Highly scalable machine learning and qualitative models v2, http://project-first.eu/public_deliverables.

[5] FIRST Consortium D6.3 Highly scalable machine learning and qualitative models v2, http://project-first.eu/public_deliverables.

**Table 3 Design Theory for Decision Support in Information-Based Market Manipulation Detection**

| Component type | Component example |
|---|---|
| (DT1) Purpose and scope | The defined purpose of the design theory is to prescribe the detection and prevention of an information-based market manipulation that involves the dissemination of false positive information in diverse media to manipulate share prices. The seven previously stated meta-requirements define the scope of the design theory (the five initial meta-requirements plus the two new meta-requirements that emerged during the theorizing search process). |
| (DT2) Constructs | Constructs were defined related to underlying design concepts ("Web feeds" acquired via Web syndication, "parallelization" and "pipelining" during data processing, and "boilerplate" to be removed from Web pages) and investor sentiment ("sentiment objects" to be identified, "document sentiment" to be measured, and "financial instruments" to be affected by price manipulation). |
| (DT3) Principles of form and function | Principles were developed to address the seven meta-requirements and as an abstraction of the instantiated problem solution. For example, decision-support systems that address information-based market manipulation should be based on glass-box models extended by black-box models (the "principle of multiple classifier systems" that addresses primarily meta-requirements five and seven). |
| (DT4) Artifact mutability | Aggregation rules of the qualitative model must be adapted depending on the monitored market segment or the addressed manipulation scheme (e.g., "pump-and-dump" versus "market sounding" manipulation schemes). Furthermore, model robustness against scammer countermeasures must be recurrently assessed to purposefully evolve the artifact, for example, by adding additional features to the bag-of-words model or new models to the multiple classifier system. |
| (DT5) Testable propositions | Propositions are developed that correspond to and provide the rationale for the proposed design principles of form and function. For example (corresponding to the previously noted principle), the combination of glass-box and black-box models into multiple classifier systems increases the system's performance to identify suspicious market behavior. |
| (DT6) Justificatory knowledge | During heuristic theorizing, ideas from various kernel theories and the literature from different disciplines, including marketing, finance, and computer science, subsequently became a theoretical foundation and justification for the design theory (e.g., behavioral finance theory). |

was derived during heuristic synthesis through analogical reasoning that prescribes the use of multiple classifier models for market manipulation detection, which was added to our design theory (DT3). However, we questioned whether this model combination is superior for this problem class in general and added a corresponding testable proposition to our design theory (DT5). In developing our design theory, theoretical grounding resulted in the identification of studies on multiple classifier systems (e.g., Ranawana and Palade 2006) that could be used as a knowledge foundation to enhance our design principle by providing a rationale and explanation (DT6) of our design theory. After re-evaluating the progress of the design theorizing process, the DSR team realized that a satisficing problem solution had been developed (breaking out of the theorizing circle).

### 5.4. Design Theory

Workshops during program meetings yielded positive feedback about the developed solution. A program industry partner was determined to integrate the developed decision support component into an existing state-of-the-art market surveillance system. Meanwhile, the researchers started to reflect on their experiences during the course of multiple described iterations. Subsequently, theoretical grounding was performed for the other design principles to integrate them into the theory. Thus, our design theory started to emerge, which is briefly summarized in Table 3.

## 6. Discussion

The heuristic theorizing framework developed in this paper contributes to the literature by (1) showing how the ideas of heuristics and heuristic search in our understanding of problem solving can be used to better understand proactive design theorizing and (2) recommending heuristic synthesis to achieve a better understanding of the generative design theory development process based on new information that is recurrently and proactively generated across the iterative stages of problem structuring and artifact design.

In previous studies, scholars have attempted to explain the process of design theorizing (constructing design theory) by relying nearly exclusively on the idea of reasoning (Gregory and Muntermann 2011, Kuechler and Vaishnavi 2008, Van de Ven 2007). However, reasoning employs a narrow perspective and has been described as a special type of thinking among other thinking types, such as problem solving (Holyoak and Morrison 2005). Our concept of heuristic synthesis accounts for the diversity of the different ways of thinking involved in proactive design science work.

Among the established ways of thinking, abduction is particularly relevant for proactive design theorizing because of its focus on problem solving (see §2). Reflection and learning, discussed in Schön (1983) and in the IS field by Levina (2005) and Sein et al. (2011), is also particularly relevant because of its usefulness

**Table 4 Heuristic Theorizing Compared with Other Proactive Research Approaches**

| Research approach | Definition/meaning | Commonality with heuristic theorizing | How heuristic theorizing differs |
|---|---|---|---|
| Grounded theorizing | Grounded theorizing has been described as "the process of iteratively and inductively constructing theory from observations using a process of theoretical sampling in which emergent insights direct selection and inclusion of the "next" informant or slice of data. Grounded theorizing furthermore involves constant comparative analysis whereby groups are compared on the basis of theoretical similarities and differences" (Gephart 2004, p. 459). | An iterative process that involves alternating between fieldwork (data collection and problem solving) and analysis (data analysis and heuristic synthesis). | The focus is not on inductive construction of explanatory theory but proactive design theory generation. |
| Applied science research | Briggs et al. (2011) view applied science research as "the last research mile." "The last research mile begins when a research team finds real people with a real problem in a real organization. They explore the problem, learn about stakeholder goals, and seek to discover drivers and constraints in the problem environment. They propose possible solutions to stakeholders and listen carefully to their responses" (p. 14). | Proposing possible solutions to stakeholders in a given problem environment. | The emphasis transcends applying knowledge to a problem environment and involves extracting design science knowledge from proactively generated problem-solving experiences. |
| Action design research | Action design research is a method for design research. "The method conceptualizes the research process as containing the inseparable and inherently interwoven activities of building the IT artifact, intervening in the organization, and evaluating it concurrently" (Sein et al. 2011, p. 37). | Interwoven activities of engaging with the problem at hand, designing an artifact, and concurrent evaluation. | The emphasis transcends design research and building a material artifact to include the generation of design theory. |
| Engaged scholarship | Engaged scholarship has been proposed as a research approach based on a combination of abduction, induction, and deduction and involves "a participative form of research for obtaining the advice and perspectives of key stakeholders (researchers, users, clients, sponsors, and practitioners) to understand a complex social problem" (Van de Ven 2007, p. IX). | Proactive production of relevant knowledge through collaboration between practitioners working on so-called wicked problems. | The emphasis transcends seeking a better understanding of a problem to include providing prescriptions, which involves a larger variety of ways of thinking. |

for extracting insights and knowledge from problem-solving experiences.

Heuristic theorizing displays similarities and differences with other proactive research approaches that focus on contributions that are "rooted" in practice (cf. Iivari 2014) (Table 4).

## 7. Conclusion

In sum, heuristic theorizing is the process of proactively generating design theory for prescriptive purposes from problem-solving experiences and prior theory by constantly iterating between the search for a satisficing problem solution, i.e., heuristic search, and the synthesis of new information that is generated during heuristic search, i.e., heuristic synthesis. Heuristic search involves alternating back between structuring the problem at hand and generating new artifact design components, whereas heuristic synthesis involves different ways of thinking that complement the use of heuristics.

We would like to conclude by reminding ourselves of an important statement by Paul Gray: "If information systems is to be perceived as relevant as a profession then we must look forward not backward. We should help create the future, not just study the past" (Kock et al. 2002, p. 339). Addressing this call, we suggest a new approach in this paper termed heuristic theorizing and provide a normative framework for proactive design theorizing.

### Acknowledgments

The authors thank the senior editor, the associate editor, and the anonymous reviewers for their constructive comments during the review process. The senior editor in particular has been extraordinarily engaged in the process of helping craft the final version of this paper. The presented DSR program received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) within the context of the Project FIRST, large scale information extraction and integration infrastructure for supporting financial decision making, under grant agreement no. 257928. The authors thank all of the members of the FIRST project consortium for their contributions to the heuristic search process described in this paper. Finally, the authors thank Thomas Widjaja and Evgeny Káganer for the useful feedback that they provided on prior versions of the paper. The authors have contributed equally to this paper.

### References

Alexander C (1964) *Notes on the Synthesis of Form* (Harvard University Press, Cambridge, MA).

Aliseda A (2006) *Abductive Reasoning: Logical Investigations into Discovery and Explanation* (Springer, Dordrecht, The Netherlands).

Allen F, Gale D (1992) Stock-price manipulation. *Rev. Financial Stud.* 5(3):503–529.

Briggs RO, Nunamaker JF Jr, Sprague R (2011) Applied science research in information systems: The last research mile. *J. Management Inform. Systems* 28(1):13–16.

Brown T (2008) Design thinking. *Harvard Bus. Rev.* 86(6):84–92.

Buchanan R (1992) Wicked problems in design thinking. *Design Issues* 8(2):5–21.

Churchman CW (1967) Guest editorial: Wicked problems. *Management Sci.* 14(4):B141–B142.

Clark GL, Kaminski PF, Brown G (1990) The readability of advertisements and articles in trade journals. *Indust. Marketing Management* 19(3):251–260.

Cross N (2006) *Designerly Ways of Knowing* (Springer, Berlin).

Cross N (2008) *Engineering Design Methods: Strategies for Product Design*, 4 ed. (Wiley, Chichester, UK).

Dasgupta S (1991) *Design Theory and Computer Science* (Cambridge University Press, Cambridge, UK).

Feigenbaum EA, Feldman J, eds. (1963) *Computers and Thought* (McGraw-Hill Inc., New York).

Gephart RP (2004) Qualitative research and the academy of management journal. *Acad. Management J.* 47(4):454–462.

Goel AK (1997) Design, analogy, and creativity. *IEEE Expert* 12(3):62–70.

Gregor S (2006) The nature of theory in information systems. *MIS Quart.* 30(3):611–642.

Gregor S, Hevner A (2013) Positioning and presenting design science research for maximum impact. *MIS Quart.* 37(2):337–355.

Gregor S, Jones D (2007) The anatomy of a design theory. *J. Assoc. Inform. Systems* 8(5):312–335.

Gregory RW, Muntermann J (2011) Theorizing in design science research: Inductive versus deductive approaches. Galletta DF, Liang T-P, eds. *Proc. Internat. Conf. Inform. Systems* (AIS, Shanghai, China).

Hayes JR (1978) *Cognitive Psychology: Thinking and Creating* (Dorsey, Homewood, IL).

Hempel CG (1966) *Philosophy of Natural Science* (Prentice-Hall Inc., Englewood Cliffs, NJ).

Hevner AR, March ST, Jinsoo P, Ram S (2004) Design science in information systems research. *MIS Quart.* 28(1):75–105.

Holmström J, Ketokivi M, Hameri AP (2009) Bridging practice and theory: A design science approach. *Decision Sci.* 40(1):65–87.

Holyoak KJ, Morrison RG, eds. (2005) *The Cambridge Handbook of Thinking and Reasoning* (Cambridge University Press, Cambridge, UK).

Iivari J (2007) A paradigmatic analysis of information systems as a design science. *Scandinavian J. Inform. Systems* 19(2):39–64.

Iivari J (2014) Distinguishing and contrasting two strategies for design science research. *Eur. J. Inform. Systems*. Forthcoming.

Kärkkäinen M, Holmström J (2002) Wireless product identification: Enabler for handling efficiency, customization and information sharing. *Supply Chain Management: Internat. J.* 7(4):242–252.

Kilduff M, Mehra A, Dunn MB (2011) From blue sky research to problem solving: A philosophy of science theory of new knowledge production. *Acad. Management Rev.* 36(2):297–317.

Kinney J (1979) Classifying heuristics. *College Composition Comm.* 30(4):351–356.

Kock N, Gray P, Hoving R, Klein H, Myers M, Rockart J (2002) Is research relevance revisited: Subtle accomplishment, unfulfilled promise, or serial hypocrisy? *Comm. Assoc. Inform. Systems* 8(1):330–346.

Kohlschütter C, Fankhauser P, Nejdl W (2010) Boilerplate detection using shallow text features. Davison BD, Suel T, Craswell N, Liu B, eds. *Proc. Third ACM Internat. Conf. Web Search and Data Mining* (ACM, New York), 441–450.

Kuechler B, Vaishnavi V (2008) On theory development in design science research: Anatomy of a research project. *Eur. J. Inform. Systems* 17(5):489–504.

Kuechler B, Vaishnavi V (2012) Characterizing design science theories by level of constraint on design decisions. Peffers K, Rothenberger M, Kuechler B, eds. *Design Science Research in Information Systems. Advances in Theory and Practice* (Springer, Berlin), 345–353.

Laurence S, Margolis E (2012) Abstraction and the origin of general ideas. *Philosophers' Imprint* 12(19):1–22.

Lee AS (2010) Retrospect and prospect: Information systems research in the last and next 25 years. *J. Inform. Tech.* 25(4): 336–348.

Lee AS, Baskerville RL (2003) Generalizing generalizability in information systems research. *Inform. Systems Res.* 14(3):221–243.

Lee J, Pries-Heje J, Baskerville R (2011) Theorizing in design science research. Jain H, Sinha A, Vitharana P, eds. *DESRIST 2011*, LNCS 6629 (Springer, Berlin Heidelberg), 1–16.

Levina N (2005) Collaborating on multiparty information systems development projects: A collective reflection-in-action view. *Inform. Systems Res.* 16(2):109–130.

Lindgren R, Henfridsson V, Schultze U (2004) Design principles for competence management systems: A synthesis of an action research study. *MIS Quart.* 28(3):435–472.

Majchrzak A, Gasser L (2000) TOP-MODELER©: Supporting complex strategic and operational decision making. *Inform. Knowledge Systems Management* 2(1):95–110.

March ST, Smith GF (1995) Design and natural science research on information technology. *Decision Support Systems* 15(4):251–266.

Markus ML, Majchrzak A, Gasser L (2002) A design theory for systems that support emergent knowledge processes. *MIS Quart.* 26(3):179–212.

Martin R (2010) Design thinking: Achieving insights via the "knowledge funnel." *Strategy Leadership* 38(2):37–41.

Müller-Wienbergen F, Müller O, Seidel S, Becker J (2011) Leaving the beaten tracks in creative work—A design theory for systems that support convergent and divergent thinking. *J. Assoc. Inform. Systems* 12(11):714–740.

Newell A, Simon HA (1972) *Human Problem Solving* (Prentice-Hall, Englewood Cliffs, NJ).

Newell A, Shaw JC, Simon HA (1958) Elements of a theory of human problem solving. *Psych. Rev.* 65(3):151–166.

Newell A, Shaw JC, Simon HA (1967) The process of creative thinking. Gruber H, Terrell G, Wertheimer M, eds. *Contemporary Approaches to Creative Thinking* (Atherton Press, New York), 63–119.

Ngai EWT, Hu Y, Wong YH, Chen Y, Sun X (2011) The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems* 50(3):559–569.

Niiniluoto I (1993) The aim and structure of applied research. *Erkenntnis* 38(1):1–21.

Ow PS, Smith SF (1987) Two design principles for knowledge-based systems. *Decision Sci.* 18(3):430–447.

Pahl G, Beitz W, Feldhusen J, Grote KH, Wallace K, Blessing L (2007) *Engineering Design: A Systematic Approach*, 3rd ed. (Springer, London).

Pearl J (1983) On the discovery and generation of certain heuristics. *AI Magazine* 4(1):23–33.

Peirce CS (1974) Collected papers of Charles Sanders Peirce. Hartshorne C, Weiss P, eds. *Pragmatism and Pragmaticism and Scientific Metaphysics*, Vol. V and VI (Belknap Press, Cambridge, MA).

Perkins DN (1981) *The Mind's Best Work* (Harvard University Press, Cambridge, MA).

Pólya G (1945) *How to Solve It: A New Aspect of Mathematical Method* (Princeton University Press, Princeton, NJ).

Pries-Heje J, Baskerville R (2008) The design theory nexus. *MIS Quart.* 32(4):731–755.

Pries-Heje J, Vinter O (2006) A framework for selecting change strategies in IT organizations. Münch J, Vierimaa M, eds. *Product-Focused Software Process Improvement* (Springer, Heidelberg, Germany), 408–414.

Ranawana R, Palade V (2006) Multi-classifier systems: Review and a roadmap for developers. *Internat. J. Hybrid Intelligent Systems* 3(1):35–61.

Rittel HWJ (1972) On the planning crisis: Systems analysis of the first and second generations. *Begrifsokonomen* 8:390–396.

Rong H, Tang Z, Govindarajan R, Douillet A, Gao GR (2007) Single-dimension software pipelining for multidimensional loops. *ACM Trans. Architecture and Code Optim.* 4(1):paper 7.

Rowe PG (1987) *Design Thinking* (MIT Press, Cambridge, MA).

Samuels WJ (2000) Signs, pragmatism, and abduction: The tragedy, irony, and promise of Charles Sanders Peirce. *J. Econom. Issues* (*Association for Evolutionary Econom.*) 34(1):207–217.

Schön DA (1983) *The Reflective Practitioner: How Professionals Think in Action* (Basic Books, New York).

Schön DA (1987) *Educating the Reflective Practitioner* (Jossey-Bass, San Fransisco).

Sein MK, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. *MIS Quart.* 35(1):37–56.

Shepherd DA, Sutcliffe KM (2011) Inductive top-down theorizing: A source of new theories or organization. *Acad. Management Rev.* 36(2):361–380.

Shleifer A (2000) *Inefficient Markets: An Introduction to Behavioral Finance* (Oxford University Press, Oxford, UK).

Simon HA (1967) The logic of heuristic decision making. Rescher N, ed. *The Logic of Decision and Action* (University of Pittsburgh Press, Pittsburgh), 1–35.

Simon HA (1973) The structure of ill structured problems. *Artificial Intelligence* 4(3–4):181–201.

Simon HA (1978) Information-processing theory of human problem solving. Estes WK, ed. *Handbook of Learning and Cognitive Processes* (Erlbaum, Hillsdale, NJ), 271–295.

Simon HA (1988) The science of design: Creating the artificial. *Design Issues* 4(1/2):67–82.

Simon HA (1996) *The Sciences of the Artificial*, 3rd ed. (MIT Press, Cambridge, MA).

Simon HA, Newell A (1958) Heuristic problem solving: The next advance in operations research. *Oper. Res.* 6(1):1–10.

Simon HA, Langley PW, Bradshaw GL (1981) Scientific discovery as problem solving. *Synthese* 47(1):1–27.

Smith GF (1988) Towards a heuristic theory of problem structuring. *Management Sci.* 34(12):1489–1506.

Smith GF (1992) Towards a theory of managerial problem solving. *Decision Support Systems* 8(1):29–40.

Teufel T (2012) What does Kant mean by "power of judgement" in his critique of the power of judgement? *Kantian Rev.* 17(2): 297–326.

Theobald M, Siddharth J, Paepcke A (2008) Spotsigs: Robust and efficient near duplicate detection in large Web collections *Proc. 31st Annual Internat. ACM SIGIR Conf. Res. Development in Inform. Retrieval* (ACM, Singapore), 563–570.

Thompson JD (1956) On building an administrative science. *Admin. Sci. Quart.* 1(1):102–111.

Trice A, Davis R (1993) Heuristics for reconciling independent knowledge bases. *Inform. Systems Res.* 4(3):262–288.

Vaishnavi V, Kuechler W (2008) *Design Science Research Methods and Patterns: Innovating Information and Communication Technology* (Auerbach Publications, Boca Raton, FL).

Van de Ven AH (2007) *Engaged Scholarship: A Guide for Organizational and Social Research* (Oxford University Press, New York).

Van Eyden RJ (1996) *The Application of Neural Networks in the Forecasting of Share Prices* (Finance and Technology Publishing, Haymarket, VA).

Walls JG, Widmeyer GR, El Sawy OA (1992) Building an information system design theory for vigilant EIS. *Inform. Systems Res.* 3(1):36–59.

Weick KE (1995) What theory is not, theorizing is. *Admin. Sci. Quart.* 40(3):385–390.

Wickelgren WA (1974) *How to Solve Problems: Elements of a Theory of Problems and Problem Solving* (W. H. Freeman and Company, San Francisco).