Enriching Conceptual Modelling Practices through Design Science

Ajantha Dahanayake[1] and Bernhard Thalheim[2]

[1] Georgia College and State University, J. Whitney Bunting School of Business Dept. of Information Technology and Marketing, Campus Box 12, Milledgeville 31061, GA, USA
ajantha.dahanayake@gcsu.edu
http://hercules.gcsu.edu/~adahanay

[2] Christian Albrechts University Kiel, Department of Computer Science, Olshausenstr. 40, D-24098 Kiel, Germany
thalheim@is.informatik.uni-kiel.de
http://www.is.informatik.uni-kiel.de/~thalheim

Abstract. Models, modelling languages, modelling frameworks and their background have dominated conceptual modelling research and information systems engineering for last four decades. Conceptual models are mediators between the application world and the implementation or system world. Design science distinguishes the relevance cycle as the iterative process that re-inspects the application and the model, the design cycle as the iterative model development process, and the rigor cycle that aims in grounding and adding concepts developed to the knowledge base. This separation of concern into requirements engineering, model development and conceptualisation is the starting point for this paper.

Research in design science and on conceptual modelling resulted in a large body of knowledge, practices, and techniques. The two research approaches have developed their approaches and solutions. This paper shows how the two approaches can be integrated without making a sacrifice for integration. Modelling is based on modelling activities. Integration therefore starts with an integrated view on modelling. As an example of this integration we shall use reasoning support for modelling. Each modelling step considers specific work products, orients towards specific aspects of the system or application, involves different partners, and uses a variety of resources.

Keywords: conceptual modelling, design science, intellectual support for modelling, modelling processes and workflows, description, prescription, documentation, conceptualisation, explanation.

1 Introduction

1.1 Conceptual Modelling as a Complex and Multi-facetted Intellectual Process

Conceptual modelling is one of the main activities during information system development. Models are used as mediating artifacts which describe the problem to be solved for the application and which are used as starting point for implementation. They are also used for documentation of the system, for migration and evolution processes, for

T. Halpin et al. (Eds.): BPMDS 2011 and EMMSAD 2011, LNBIP 81, pp. 497–510, 2011.
© Springer-Verlag Berlin Heidelberg 2011

optimisation of systems, for control of parts of systems, and for simulation of systems. Models must reflect the structure of a system, the functionality of a system, the support facilities of a system and the collaboration environment of a system. Therefore, models are rather complex or are using multiple sub-models. Furthermore, models must be scalable to different abstractions in dependence on stakeholders such as business architects, modellers, programmers, component developers or operators.

This inherent complexity of models makes conceptual modelling itself a complex task. Conceptual modelling is thus complex and multi-facetted due to the variety of aspects to be considered and due to scaling requirements. We may use context abstraction and scoping techniques during modelling activities and thus concentrate on some of the aspects. This separation of concern supports an evolutionary approach to model development.

We also may differentiate modelling activities by problem kinds that are of importance at a current modelling step. For instance, the abstraction layer model [35] distinguishes the application domain layer, the business process layer, the business operating layer, the conceptual layer and implementation layers. With such distinction we realise that tasks at each level are of different nature. For instance, at the application domain layer we describe the way how applications are operating, which application processes must be supported by which application data at which moment of time in which format, for which reason, by and for whom, and in which environments.

1.2 Design Science versus or Joined with Conceptual Modelling

Design science started with an analysis and assessment of research activities that are used for system development in computer engineering. The separation into the application world, the modelling and model world and the knowledge or science world [13,39] supported an assessment of the results of modelling and an evaluation of the results of research on modelling. The application world is characterised by three dimensions: people, organisations and the (technical) infrastructure. The science world is based on scientific theories with specific paradigms and justification underpinnings from one side and techniques for deployment of these theories on the other side. The modelling and model world is supported by the science world from one side and governed by the application world. This mediating role of models and modelling results clarifies the importance, value and application of models.

Conceptual modelling has been oriented in the past mainly to clarification on languages, on methods for deployment of such languages, on (mathematical) theories as foundations of syntactics, semantics and pragmatics of model, and on evaluation and quality guaranteeing methods [27,37,39]. The application world is used as a starting point for the development of systems that solve some problems of the application domain under consideration. The specific objectives of the application domain are mapped to requirements which are then used as the main governing background for the system development. A theory of conceptual models and of conceptual modelling is currently under development [38,39].

If we analyse these two directions we come to a conclusion similar to [45]. In reality design science research and research on conceptual modelling are only two sides of the same coin. The two communities behind the two sides of the coin are currently engaged in a discussion of the added value of each side. We take another direction in this paper. We will show how these two sides can be neatly integrated. We realise that it is natural to integrate conceptual modelling theories and design science. We target at a revision and extension of classical information systems modelling methodologies and at overcoming limitations of such modelling processes. This paper is based classical notions[1] which are used for IS design and construction and extends the theory of conceptual models [39] by design science research. It generalises co-design for information systems [34,35,36], design science research [13,23], business information systems engineering [11,17], and different frameworks, e.g. FRISCO [12]. Based on this approach we enrich conceptual modelling practices by design science research.

1.3 Structure of the Paper

We first revisit design science and discover how design science and classical approaches to information systems development can be integrated. Section 4 is based on the theory of conceptual models [6,39] and of modelling activities [38]. We introduce five of the main workflows of modelling and restrict consideration to IS development processes, i.e., we base development on description of IS followed by prescription for construction of IS. Section 3 develops a path to integration of design science and conceptual modelling. Section 4 uses this approach for homogenisation of activities that are both used in design science and conceptual modelling. We show that design science and conceptual modelling theory can be neatly integrated.

2 Design Science Revisited

Information systems research (ISR) lies at the intersection of people, organizations, and technology [31]. It relies on and contributes to cognitive science, organizational theory, management sciences, and computer science. It is both an organizational and a technical discipline that is concerned with the analysis, construction, deployment, use, evaluation, evolution, and management of information system artifacts in organizational settings [21,28,40]. Within this setting, the design-science research paradigm is proactive with respect to technology. It focuses on creating and evaluating innovative IT artifacts that enable organizations to address important information-related tasks.

2.1 Design Research in Information Systems

The genesis of design science lies in Herbert Simon's The Sciences of the Artificial (first published in 1969) in which he articulated the difference between natural science, concerned with how things are, and design science, concerned with how things ought to be, based on his understanding of design as problem solving [32]. Following Simon's tradition, design science was introduced in IS (information systems) research by March and Smith [22], who presented it as prescriptive research aimed at improving

[1] See, for instance, chapters in the collection [6, chapters 1 (Modelling as programming), 2 (MMD), 6 (Extended ER models), 7 (Codesign), 8 (Business processes), 10 (Interaction models), 11 (Web IS development), 12 (Evolution and migration), 13 (Data integration)].

ICT (information, communication, and technology) performance, as opposed to natural science, corresponding to descriptive research aimed at understanding the nature of ICT. An important point was that IS research should actually integrate both perspectives, an argument that came back on Hevner, et. al., [13], establishing design science research in information systems (DSRIS). A DSRIS contribution requires identifying a relevant organizational ICT problem, demonstrating that no solution exists, developing an ICT artifact that addresses this problem, rigorously evaluating the artifact, articulating the contribution to the ICT knowledge-base and to practice, and explaining the implications for ICT management and practice [23].

2.2 The Three Design Science Research Cycles

A recent development of the initial design science framework presented in [13], decouples the previous goals into three distinct but interrelated research cycles, as shown in Figure 1. This three-cycle view of design science suggests that relevance is attained through identification of requirements (or business needs) and field testing of an artifact within an environment, while rigor is achieved by appropriately grounding the research in existing foundations, methodologies and design theories and subsequently making contributions that add to the existing knowledge base. The design aspect is achieved through a *design cycle* in which the artifact must be built and evaluated thoroughly before "releasing it" to the *relevance cycle* and before the knowledge contribution is output into the *rigor cycle* [14].

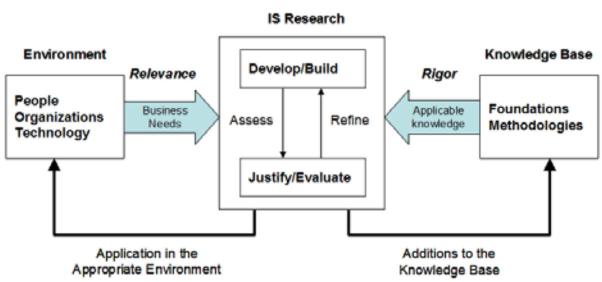


Fig. 1. Design science research cycles as introduced by A.R. Hevner et al. [13]

2.3 Strengths, Weaknesses, Opportunities and Threats of Design Science Research

IS design is successful when it has a contribution that applies a range of methods, when is guided by vision, and when it has a positive impact on society. Design science has relevance in all academic systems even though rarely cited in other fields [9]. Design science has achieved a hegemonic status and has become an orthodoxy instead of constantly evolving set of precepts taking its legitimate place amongst the panoply of concerns the IS community needs to deal with [19]. Those concerns characterized design science as a "wicked" problem that needs 'rethinking' in order to elevate DSRIS to the equal footing with other research paradigms in IS research [2,16,29,20].

The dual goal of DSRIS is producing both an artifact and a theoretical contribution. March and Smith [22] attach the activities of Discovery (generating or proposing scientific claims) and Justification (testing scientific claims for validity) to natural science and present them as separate from (but parallel to) the activities of Building (constructing an artifact for a specific purpose) and Evaluation (determining how well the artifact performs) attached to design science. In Hevner et al. [13], the activities were merged into Develop/Build and Justify/Evaluate. This helps state the case in favor of having both relevance and rigor in ISR, but leave behind lack of clarity with regards to how theory development should be seen in DSRIS.

On one end of the spectrum, March and Smith [13] explicitly exclude theory and theorizing from design science. On the other end, several authors contend that theory development should be an integral part of DSRIS [18,24,43]. Hevner et al. [13] do not seem to take a stance either way, since they present a dual build/develop and evaluate/justify design cycle, potentially allowing for both artifact building and evaluation as well as theory development and justification.

When DSRIS is used for theory development, the next question is what kind of theory can result. Walls et al. [43] speak of design theories, which are prescriptive theories about how to design information systems effectively and feasibly. Venable [42] claims that design theories should be reduced to utility theories, which are predictive (rather than prescriptive) about the utility of applying a meta-design to solve meta-requirements. Theory can also be related to the kinds of artifacts produced by DSRIS, which according to [13] may be constructs, models, methods and/or instantiations. For Winter [44] theories should be considered a fifth (intermediate) type of artifact. In contrast, Gregor and Jones [9] take a "broad view of theory" which encompasses constructs, models and methods, and where only instantiations correspond to the (material) artifact as such. Theory in this last sense is equivalent to what is termed elsewhere conjectures, models, frameworks, or bodies of knowledge, so the three outputs of design science, besides the instantiated artifact, are regarded as "components of theory" [44]. The options for a theory being (a) prescriptive design theory, (b) predictive utility theory, (c) an additional intermediate artifact or (d) all abstract artifacts.

According to Iivari [15], the different types of knowledge produced are what determine the epistemology underlying DSRIS. Gregor [8], however, has strongly argued that the type of theory produced by ISR should not depend on the underlying paradigm; for her, theory is independent from specific ontological or epistemological choices. Nonetheless, both recognize the importance of making explicit choices about the philosophy underlying the research. Although DSRIS is not specific in terms of an underlying epistemology, it may be seen as rooted in pragmatism in the sense that it emphasizes utility (the measure of truth in pragmatism) [13]. This assumption, although widely held, does not hold for all kinds of DSRIS, making it open to alternative epistemology, such as interpretivism [15,7,26].

A check list for understanding design science is Baskerville's [1] what design science is not list: Design science is not design, is not design theory, is not an IT artifact, is not a methodology, is not action research, is not computer science, is not a separate academic discipline, and it is not new. Also, design science research is in an up-hill struggle in defining the term design science [1,33].

First, in order to show the path for integrated understanding we harmonise Hevner et al. [13] and Winter [44] interpretations of design science research in information systems as the combination of two different types of information systems research contributions: (IS) design science as contributions that reflect at a generic level the construction and evaluation of artifacts and (IS) design research as construction and evaluation of specific artifacts. (IS) design science research agrees on the differentiation of constructs, models, methods and instantiations as types of design science research artifacts [3,22,40,44].

Second, we correlate notions used in design science research and information systems design and construction in Figure 2 between design science research and (professional) information systems design and construction.

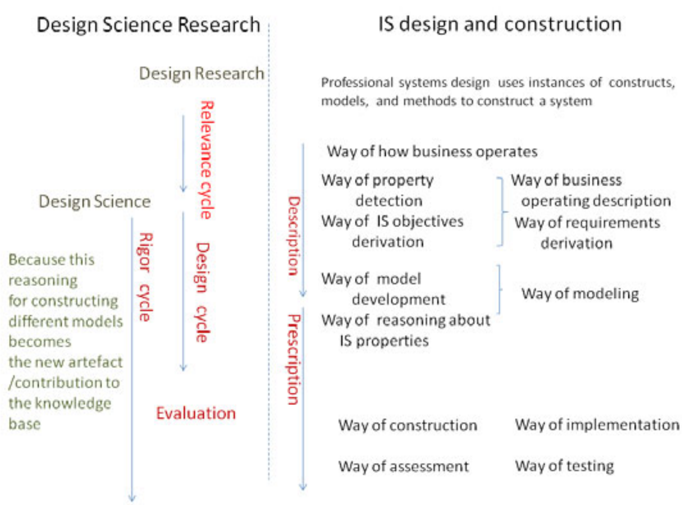


Fig. 2. Framework for understanding Design Science Research and Information Systems Design & Construction

The main difference between (professional) information systems design and construction and design science research is that; design science research (DSR) contribution requires identifying a relevant organizational ICT problem, demonstrating that no solution exists, developing an ICT artifact that addresses this problem, rigorously evaluating the artifact, articulating the contribution to the ICT knowledge-base and to practice, and explaining the implications for ICT management and practice [23]. The professional system design and construction use the available artifacts.

The main drawback in conducting design science research is the lack of a theory for construction of artifacts that supports the design cycle. Therefore, design science research needs to explore the types of models and reasoning behind model construction (actually this is conceptual modelling theory). The rigor cycle in design science targets at creation of design/modelling knowledge. It considers the the model or artifact constructed. The relevance and the design cycles of design research target at construction of artifacts. These two facets of DSR require a deep integration and sophisticated reasoning support.

2.4 Alternatives for a Notion of a Theory

The notion of a theory itself is be a matter of intensive research [5,25]. We base our understanding of the notion of a theory on the three dimensions and the main goal of conceptual modelling.The classical treatment of the notion of a theory is based on mathematical and philosophical logics and is far too strict. We may however inherit certain elements of such logics. Already the notion of semantics provides a larger number of choices [30] beyond those that are taken for granted in Computer Science.

Theories of conceptual modelling must step beyond axiomatic and analytical theories. They must also be operational and 'genetic'. Theories of conceptual modelling can be developed in the frameworks of logical empiricism, of context theories ('context of use', 'language game'), and of constructivism. The first framework supports to define purposes of conceptual modelling, to emphasise threats that should be handled with the help of models, to select appropriate modelling languages and methods, to reason on the quality of the model depending on the purpose of the model, to select measures for the quality of models, and to guide the process of modelling. The second framework relates models to the application domain, to the stakeholders participating in the development process, to the aspects reflected within a model, and to the resources provided either by the system and by the knowledge from the application domain. The last framework provides a basis for a general structure by a *language of constructs* that can be applied for the development of a model, a *set of constructors* that can be applied to combine models into a new model, and a *number of quality properties for characterisation of usage* of certain constructs.

3 Integration of Design Science and Conceptual Modelling

3.1 Associating Design Science Approaches with Conceptual Modelling

Conceptual modelling within the business information community and design science approaches are two sides of the same 'Janus' head of a coin. IS design science aims at the development of a general theory for models, modelling activities and modelling. We shall use the approach for a deeper insight into modelling. We claim that each enhance the other. We must however integrate the vocabulary used in the two approaches. For instance, models are called 'design' in [13] and 'theories' in [9].

The information systems community characterises the modelling process by seven guidelines [13]:

(1) model are purposeful IT artifacts created to address a problem;
(2) models are solutions to relevant and important problems;
(3) the utility, quality, and efficacy of models must be evaluated by quality assessment;
(4) modelling research must contribute to the state of the art;
(5) modelling research relies upon the application of rigorous methods;
(6) modelling is a search process and use termination conditions;
(7) models must be communicated both to technology-oriented as well as to management audiences.

We observe that guidelines (1), (2), and (7) are characterising the model. Guidelines (3), (6) characterise modelling activities. Guidelines (3) and (5) are related to modelling as a technology. Guideline (4) is a general statement that raise the issue whether modelling may become a science.

Main ingredients of modelling can be derived from these guidelines [9,23]. Core components are purpose and scope (causa finalis), artifacts (causa materialis), the oneness of form and function (causa formalis), artifacts mutability, testable propositions about the model, and theoretical underpinning. Additional requests are the potential implementation (causa efficiens) and utility for exposition and testing [9].

3.2 Associating Design Science Cycles with Main Activities of IS Modelling

Design science separates three cycles [41]: the relevance (or description) cycle, the design (or modelling) cycle, and the rigor (or conceptualisation and generalisation or knowledge development) cycle.

The rigor cycle also aims at the development of knowledge about the application domain and the model. This part of the rigor cycle is *conceptualisation*. The second target of the rigor cycle is the derivation of abstract knowledge and experience, of scientific theories that can be applied in similar application cases, of (pragmatical) experience for modelling, and of meta-artifact or reference models based on model-driven development (MDD) approaches. Design science aims at another kind of model refinement by adding more rigor after evaluation of a model. This refinement is essentially *model evolution* and *model evaluation*. Another refinement is the enhancement of models by concepts. This refinement is essentially a 'semantification' or *conceptualisation* of the model.

We observe that the rigor cycle or stage is orthogonal to the modelling and relevance cycles. The modelling cycle may be broken into a description stage that relates the application domain to the model and a prescription stage that uses the model for system construction. The rigor cycle or stage is somehow orthogonal. It has at least two facets: one facet that is important for the model and one facet that is important for generalisation of the model, e.g., for derivation of pattern or reference models and for extraction of model and modelling knowledge beyond the actual modelling activity. In the sequel we concentrate on the first facet of the rigor cycle.

3.3 Stages Observed in Conceptual Modelling

We can now summarise our revision of design science. Design science uses the three cycles: relevance, modelling, and rigor cycle. The main purpose of conceptual modelling of information systems is the construction of an information system. This construction is based on description of the application domain and the of the model. We reflect some relevant properties of the application domain by obligations to the model and by the model. This model is typically used as a prescription for the construction. Therefore we distinguish the stages displayed in Figure 3.

The relevance cycle used in design science contains the modelling stage and a small fraction of the realisation stage. Conceptual modelling is not explicitly considering the relevance stage. It entirely uses the modelling and the realisation stage. Conceptualisation is an orthogonal activity that aims at a theoretical underpinning of models. It is used

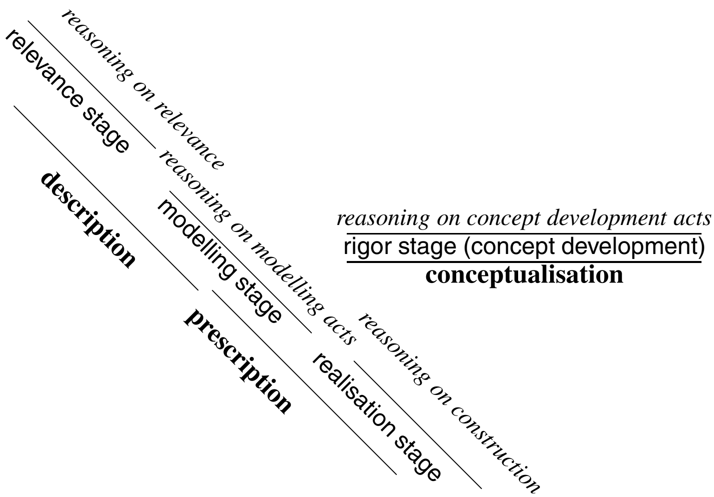


Fig. 3. The stages of conceptual modelling activities for the description-prescription workflow enhanced by conceptualisation

for *semantification* of models and for improvement of comprehensibility of models and elements used in models.

4 An Integration of Modelling Activities

Design science modelling and classical conceptual modelling of information systems mainly aim at construction of systems. Therefore, we may also restrict this paper to activities that support construction. In the sequel we introduce some of the main macro-activities and derive corresponding workflows: description, prescription, conceptualisation, explanation, and documentation. Other macro-activities are: exploration; optimisation and variation; verification.

Modelling is typically supported by languages that are well-founded and easy to apply for the description of the application domain, the requirements and the system solution. Modelling activities are ruled by the *purpose* of the model. The purpose in the center of this paper is construction of an artifact or of an information system. Beside this purpose there are many other purposes such as perception support for understanding the application domain, explanation and demonstration for understanding an origin, preparation to management and handling of the origin, optimisation of the origin, hypothesis verification through the model, control of parts of the application, simulation of behaviour in certain situations, and substitution for a part of the application.

The construction purpose is supported by a consideration of (1) the part of the application domain (called origin) that is of interest, (2) the model that is used for construction and that reflect the origin, (3) the enhancement of the model by concepts for conceptualisation, documentation and explanation, and (4) the realisation facilities used for the construction of the information system which is a technical artifact.

4.1 Description of Origin and Reflection by the Model

The application domain consists of people, organisational systems, and technical systems that interact to work towards a goal. This description of the application domain clarifies problems to be solved depending on models purpose, properties reflected or neglected, the scope to specific elements, the restrictions of the world associated with the model.

A *model* is typically a schematic description of a system, theory, or phenomenon of an origin that accounts for known or inferred properties of the origin and may be used for further study of characteristics of the origin.

The description of the origin depends on the language $\mathcal{L}_{AD}$ used within the application domain and on the theories $Theor(O)$ behind the origin. It results in a number of properties $\Phi(O)$ of the origin. These properties must be understood and mapped to objectives $\Psi(A)$ for the model A. This mapping depends on the language $\mathcal{L}_{artifacts}$ for artifacts used for the model. Figure 4 represents the association between an origin and a model.

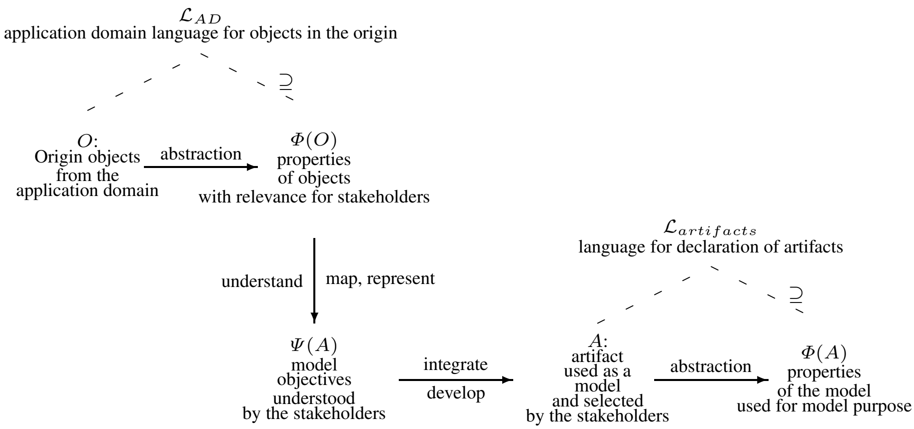


Fig. 4. Modelling for description of the origin and as the basis of realisation by a technical artifact

4.2 Prescription for Construction of Systems

Properties $\Phi(A)$ of the model A rule the construction of a system in the realisation stage. Properties also incorporate static and dynamic integrity constraints [35]. Typically we assume that the model and the properties are combined within a specification. Modern methodologies are based on interactive automatic mappings to the technical artifact TA. These properties serve as objectives for the technical artifact development, i.e. *prescribing* the information system.

4.3 Conceptualisation of Models

Conceptualisation extends the model by a number of concepts that are the basis for an understanding of the model and for the explanation of the model to the user. [39] introduces a general theory of concepts that can be used for conceptualisation. Concepts are used in everyday life as a communication vehicle and as a reasoning chunk. Concept definition can be given in a narrative informal form, in a formal way, by reference to some other definitions etc. We may use a large variety of semantics [30], e.g., lexical or ontological, logical, or reflective.

Our revision of the design science and the conceptual modelling frameworks in Figure 3 shows that concept enhancement is an orthogonal activity. It can be partially completed without having a negative impact on the realisation phase if stakeholders involved have a common understanding of the model and its properties and a commonsense about the objectives for realisation and a good mapping facility. Therefore, conceptualisation may also be implicit and use some kind of lexical semantics, e.g. word semantics within a commonly agreed name space.

4.4 Explanation

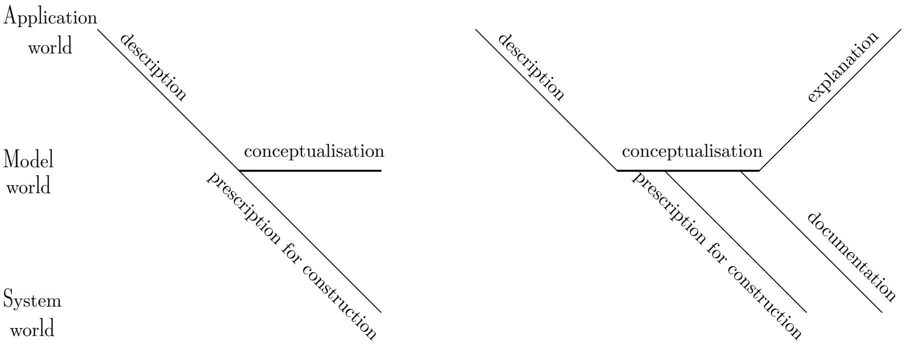
According to [10], MIS distinguishes four main tasks: description of application domains, construction of systems, prognosis of the impact of the IS, and finally explanation to business users. Explanation is often understood as a self-description of the model and thus mistreated in discussions [29,16]. Design science research and behaviouristic approaches use explanation for reasoning and/or observing on system or model properties in terms of the application world. Reasoning is thus based on deductive-nomological methods and methods of empirical research.

4.5 Documentation Based on Modelling Results

Classical information systems development follows sometimes the late documentation approach, i.e. the development documentation is based on the final result modelling and/or realisation and partial consideration of the conceptualisation, i.e. consists of the model and its translation to relational or object-relational system languages. Documentation should however reflect all thoughts and decisions applied in modelling and realisation, i.e., record all reasoning, construction and decision processes for model and system development.

4.6 Modelling Workflows

So far we introduced different stages such as the relevance stage, the modelling stage, and the realisation stage. These stages can be combined with conceptualisation, explanation and documentation. Figure 5 shows how these stages can be combined. Often, system development is based on agile development, i.e., we use description followed by prescription. This development process may be enhanced by adding more rigor to the model, i.e., concept enhancement of the model. This rigor is also partially useful for the system construction. Therefore, we arrive at the picture at the right side of Figure 5. Additional conceptualisation might be used for system development documentation. The full conceptualisation is a good starting point for an explanation of the system to the real life business users and for empirical studies in the application domain.



(a) Direct realisation without considering concepts

(b) Realisation with consideration of concepts

Fig. 5. Two approaches to systems development

5 Conclusion

Based on the definition of two types of design science research contributions (design science and design research) we illustrated the parallels between design science research and professional information systems design and construction. Mapping the relevance and design cycle to professional information systems modelling and models, the overlaps and gaps within the descriptive and prescriptive modelling needs were identified and presented. This gap analysis led to the integration of reasoning in design science and conceptual modelling and is a source of inspiration for abstraction and theory formation. Our research demonstrates the potential of an integration of design science research from one side and of conceptual modelling from the other side.

Even though, presented and discussed separately, research in design science and on conceptual modelling resulted in a large body of knowledge, practices, and techniques, contributing to each other. We have presented a starting point that can generate many contributions leading to fruitful discussions on the inter relationships and contributions of design science research and conceptual modelling. This paper is the first series of articles that will devote to improve the understanding, parallels, similarities and contributions of to each other in design science research and conceptual modelling.

Our revision and integration of design science and conceptual modelling frameworks led to a better understanding of potential relationship of conceptual modelling and design science. We have been using an integrated model. Alternatively we could use a model suite [4] consisting of combined models, e.g., a description model, a prescription model, a conceptual model, an explanation model, a documentation, etc.

References

1. Baskerville, R.: What design science is not. EJIS 17(5), 441–443 (2008)
2. Baskerville, R., Lyytinen, K., Sambamurthy, V., Straub, D.: A response to the design-oriented information systems research memorandum. European Journal of Information Systems, 1–5 (2010)

3. Vom Brocke, J., Buddendick, C.: Reusable conceptual models - requirements based on the design science research paradigm. In: DESRIST, pp. 576–604 (2006)
4. Dahanayake, A., Thalheim, B.: Co-evolution of (Information) system models. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. Lecture Notes in Business Information Processing, vol. 50, pp. 314–326. Springer, Heidelberg (2010)
5. Deppert, W.: Theorie der Wissenschaft. Christian-Albrechts-University at Kiel, Lecture notes for academic year 2008/2009 (2009), `http://wolfgang.deppert.de`
6. Embley, D., Thalheim, B. (eds.): Handbook of conceptual modelling: Its Usage and Its Challenges. Springer, Heidelberg (2011)
7. Klabbers, J.H.G.: A framework for artifact assessment and theory testing. Simulation & Gaming 37(2), 155–173 (2006)
8. Gregor, S.: The nature of theory in information systems. MIS Quarterly 30(3), 611–642 (2006)
9. Gregor, S., Jones, D.: The anatomy of a design theory. Journal of Association for Information Systems 8(5), 312–335 (2007)
10. Heinrich, L.J., Heinzl, A., Riedl, R.: Wirtschaftsinformatik: Einführung und Grundlegung, 4th edn. Springer, Berlin (2011)
11. Hesse, W.: Dinosaur meets Archaeopteryx? or: Is there an alternative for Rational's Unified Process? Software and System Modeling 2(4), 240–247 (2003)
12. Hesse, W., Verrijn-Stuart, A.A.: Towards a theory of information systems: The FRISCO approach. In: EJC, pp. 81–91 (2000)
13. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. MIS Quaterly 28(1), 75–105 (2004)
14. Hevner, A.R.: The three cycle view of design science. Scandinavian J. Inf. Systems 19(2) (2007)
15. Iivari, J.: A paradigmatic analysis of information systems as a design science. Scandinavian J. Inf. Systems 19(2) (2007)
16. Junglas, I., Niehaves, B., Spiekermann, S., Stahl, B.C., Weitzel, T., Winter, R., Baskerville, R.: The inflation of academic intellectual capital: the case for design science research in europe. European Journal of Information Systems, 1–6 (2010)
17. Koch, S., Strecker, S., Frank, U.: Conceptual modelling as a new entry in the bazaar: The open model approach. In: OSS. IFIP, vol. 203, pp. 9–20. Springer, Heidelberg (2006)
18. Kuechler, B., Vaishnavi, V.: On theory development in design science research: Anatomy of a research project. European Journal of Information Systems 17(5), 489–504 (2008)
19. Land, F., Loebbecke, C., Angehrn, A.A., Clemons, E.K., Hevner, A.R., Mueller, G.: ICSI 2008 panel report: Design science in information systems: Hegemony, bandwagon, or new wave? Communication of the Association for Information Systems 24(29), 501–508 (2009)
20. Lyytinen, K., Baskerville, R., Iivari, J., Téeni, D.: Why the old world cannot publish? overcoming challenges in publishing high-impact is research. EJIS 16(4), 317–326 (2007)
21. Madnick, S.E.: The challenge: To be part of the solution instead of being part of the problem. In: Second Annual Workshop on Information Technology and Systems, pp. 1–9 (1992)
22. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision Support Systems 15(4), 251–266 (1995)
23. March, S.T., Storey, V.C.: Design science in the information systems discipline: An introduction to the special issue on design science research. MIS Quarterly 4, 725–730 (2008)
24. Markus, M.L., Majchrzak, A., Gasser, L.: A design theory for systems that support emergent knowledge processes. MIS Quarterly 26(3), 179–212 (2002)
25. Mittelstraß, J. (ed.): Enzyklopädie Philosophie und Wissenschaftstheorie. J.B. Metzler, Stuttgart (2004)

26. Niehaves, B.: On epistemological pluralism in design science. Scandinavian J. Inf. Systems 19(2) (2007)
27. Olivé, A.: Conceptual modeling of information systems. Springer, Berlin (2007)
28. Orlikowski, W.J.: Using technology and constituting structures: A practice lens for studying technology in organizations. Organization Science 11(4), 404–428 (2000)
29. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on design-oriented information systems research. European Journal of Information Systems, 1–4 (2010)
30. Schewe, K.-D., Thalheim, B.: Semantics in data and knowledge bases. In: Schewe, K.-D., Thalheim, B. (eds.) SDKB 2008. LNCS, vol. 4925, pp. 1–25. Springer, Heidelberg (2008)
31. Silver, M.S., Markus, M.L., Beath, C.M.: The information technology interaction model: A foundation for the MBA core course. MIS Quarterly 19(3), 361–390 (1995)
32. Simon, H.: The Sciences of the Artificial. MIT Press, Cambridge (1981)
33. Sutton, R.I., Staw, B.M.: What theory is not. Administrative Science Quartly 40(3), 371–384 (1995)
34. Thalheim, B.: Codesign of structures, functions and interfaces in database applications. Technical Report Preprint I-05-1997, Brandenburg University of Technology at Cottbus, Institute of Computer Science (1997) (in German)
35. Thalheim, B.: Entity-relationship modeling – Foundations of database technology. Springer, Berlin (2000)
36. Thalheim, B.: Co-design of structuring, funtionality, distribution, and interactivity of large information systems. Computer Science Reports 15/03, Cottbus University of Technology, Computer Science Institute (2003)
37. Thalheim, B.: Database component ware. In: ADC 2003. Australian Computer Science Communications, vol. 25(2), pp. 13–26 (2003)
38. Thalheim, B.: Towards a theory of conceptual modelling. Journal of Universal Computer Science 16(20), 3102–3137 (2010), http://www.jucs.org/jucs_16_20/towards_a_theory_of
39. Thalheim, B.: The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling. In: The Handbook of Conceptual Modeling: Its Usage and Its Challenges, ch. 17, pp. 547–580. Springer, Berlin (2011)
40. Vahidov, R.: Design researcher's is artifact: a representational framework. In: DESRIST, pp. 9–33 (2006)
41. Venable, J.R.: Design science research post hevner et al.: Criteria, standards, guidelines, and expectations. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 109–123. Springer, Heidelberg (2010)
42. Venable, J.R.: The role of theory and theorising in design science research. In: DESRIST (2006)
43. Walls, J.G., Widmeyer, G.R., El Sawy, O.: Building an information system design theory for vigilant eis. Information Systems Research 3(1), 36–59 (1992)
44. Winter, R.: Design science research in Europe. European Journal of Information Systems 17(5), 470–475 (2008)
45. Zelewski, S.: Kann Wissenschaftstheorie behilflich für die Publikationspraxis sein? In: Lehner, F., Zelewski, S. (eds.) Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik, pp. 71–120. GTO (2007)