Models, To Model, and Modelling
-
Towards a Theory of Models, especially Conceptual Models and Modelling
-
Second Collection of Recent Papers (2015-2017)

Bernhard Thalheim

Christian-Albrechts-University Kiel, Department of Computer Science, 24098 Kiel, Germany
thalheim@is.informatik.uni-kiel.de

A **model** *is a well-formed, adequate, and dependable instrument that represents origins.*

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its *community of practice* within some *context* and correspond to the *functions* that a model fulfills in *utilisation scenarios*.

As an instrument or more specifically an artifact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form. The background is often implicit and hidden.

A well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*. Well-formedness enables an instrument to be *justified* by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability or validation, and by stability and plasticity within a collection of origins. The instrument is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions). A well-formed instrument is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.

Content of This Second Collection

20. B. Thalheim. Model-based engineering for database system development [Tha17c] 99

    See our approaches to development of methodologies for modelling, especially database modelling and the codesign framework in the 90ies and 00s.

21. B. Thalheim. General and specific model notions [Tha17b] 117

22. B. Thalheim. Normal models and their modelling matrix [Tha] 131

23. Y. Kropp and B. Thalheim. Data mining design and systematic modelling [KT17] 155

    See the previous work on data mining design [BT12,JP10,ST12].

24. A. Dahanayake and B. Thalheim. The rigor cycle of conceptual modelling [DT18] 163

    See also our contributions to design science foundations of conceptual modelling [DT10b,DT10a,DT11,DT13b].

25. V. Storey and B. Thalheim. Conceptual modeling: Enhancement through semiotics [ST17] 181

    See our previous work on the entity-relationship approach to modelling, e.g. [ET11,Tha00,MST09b,ST15], on semantics [ST13,Tha11b], on programming, on the theory of databases and information systems and on the technology of information systems at
    http://dblp.uni-trier.de/pers/hd/t/Thalheim:Bernhard
    or https://www.researchgate.net/profile/Bernhard_Thalheim/publications or
    https://scholar.google.com/citations?user=lkH3h9gAAAAJ or
    http://independent.academia.edu/BernhardThalheim or ...

The First Collection

available through research gate and academia

Table of Content of the First Collection

Remark: page numbers from the first collection in the sequel.

References

[ADT12a] S. Amarakoon, A. Dahanayake, and B. Thalheim. Domain requirements modeling framework for cross-disciplinary service systems development. In *ICCNDT*, pages 152–166. Gulf University, Bahrain, 2012.

[ADT12b] S. Amarakoon, A. Dahanayake, and B. Thalheim. A framework for modelling medical diagnosis and decision support services. *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 2(4):7–26, 2012.

[AFT09] Sabah S. Al-Fedaghi and Bernhard Thalheim. Personal information databases. *IJCSIS*, 5(1):11–20, 2009.

[BFA$^+$16] M. Bichler, U. Frank, D. Avison, J. Malaurent, P. Fettke, D. Hovorka, J. Krämer, D. Schnurr, B. Müller, L. Suhl, and B Thalheim. Theories in business and information systems engineering. *Business & Information Systems Engineering*, pages 1–29, 2016.

[BST09] E. Börger, O. Sörensen, and B. Thalheim. On defining the behavior of or-joins in business process models. *Journal of Universal Computer Science*, 15(1):3–32, 2009.

[BT12] P. Broman and B. Thalheim. Interactive data extraction from semi-structured text. In *Information Modelling and Knowledge Bases XXII*, volume 237 of *Frontiers in Artificial Intelligence and Applications*, pages 1–19. IOS Press, 2012.

[DT10a] A. Dahanayake and B. Thalheim. Co-evolution of (information) system models. In *EMMSAD 2010*, volume 50 of *LNBIP*, pages 314–326. Springer, 2010.

[DT10b] A. Dahanayake and B. Thalheim. Towards a framework for emergent modeling. In *ER Workshops*, volume 6413 of *Lecture Notes in Computer Science*, pages 128–137. Springer, 2010.

[DT11] A. Dahanayake and B. Thalheim. Enriching conceptual modelling practices through design science. In *BM-MDS/EMMSAD*, volume 81 of *Lecture Notes in Business Information Processing*, pages 497–510. Springer, 2011.

[DT12] A. Dahanayake and B. Thalheim. A conceptual model for IT service systems. *Journal of Universal Computer Science*, 18(17):2452–2473, 2012.

[DT13a] A. Dahanayake and B. Thalheim. The conception of a service model. In *Proc. ICNVICT'2013*, pages 36–49, Amman, 2013. IEEE.

[DT13b] A. Dahanayake and B. Thalheim. Continuous database engineering. *Int. Journal Business Inf. Syst. (IJBIS)*, 11(4):26–58, 2013.

[DT15] A. Dahanayake and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter W^*H: The conceptual Model for services, pages 145–176. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.

[DT18] A. Dahanayake and B. Thalheim. The rigor cycle of conceptual modelling. Tributes, page forthcomng. College Publications, 2018.

[ET11] D. Embley and B. Thalheim, editors. *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*. Springer, 2011.

[JP10] K. Jannaschk and T. Polomski. A data mining design framework - A preview. In *Advances in Databases and Information Systems - 14th East European Conference, ADBIS 2010, Novi Sad, Serbia, September 20-24, 2010. Proceedings*, volume 6295 of *Lecture Notes in Computer Science*, pages 571–574. Springer, 2010.

[JT10] H. Jaakkola and B. Thalheim. Framework for high-quality software design and development: a systematic approach. *IET Software*, 4(2):105–118, 2010.

[JT11] H. Jaakkola and B. Thalheim. Architecture-driven modelling methodologies. In *Information Modelling and Knowledge Bases*, volume XXII, pages 97–116. IOS Press, 2011.

[KST09] M. Kirchberg, O. Sörensen, and B. Thalheim. A BPMN case study: Paper review and submission system. In *GI Jahrestagung*, volume 154 of *LNI*, pages 4067–4081. GI, 2009.

[KT17] Y. Kropp and B. Thalheim. Data mining design and systematic modelling. In *Proc. DAMDID/RCDL'17*, pages 349–356, Moscov, 2017. FRC CSC RAS.

[MST09a] Hui Ma, K.-D. Schewe, and B. Thalheim. Modelling and maintenance of very large database schemata using meta-structures. In *UNISCON*, volume 20 of *Lecture Notes in Business Information Processing*, pages 17–28. Springer, 2009.

[MST09b] Hui Ma, Klaus-Dieter Schewe, and Bernhard Thalheim. Geometrically enhanced conceptual modelling. In *ER*, volume 5829 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2009.

[MSTQ09] Hui Ma, Klaus-Dieter Schewe, Bernhard Thalheim, and Q.Wang. A service-oriented approach to web warehousing. In *iiWAS*. ACM, 2009.

[MSTW09a] H. Ma, K.-D. Schewe, B. Thalheim, and Q. Wang. A theory of data-intensive software services. *Service Oriented Computing and Applications*, 3(4):263–283, 2009.

[MSTW09b] H. Ma, K.-D. Schewe, B. Thalheim, and Q. Wang. A theory of data-intensive software services. *Service Oriented Computing and Applications*, 3(4):263–283, 2009.

[MSTW11] H. Ma, K.-D. Schewe, B. Thalheim, and Q. Wang. Cloud warehousing. *Journal of Universal Computer Science*, 17(8):1183–1201, 2011.

[MSTW12] Hui Ma, K.-D. Schewe, B. Thalheim, and Q. Wang. Conceptual modelling of services. *Journal of Universal Computer Science*, 18(17):2361–2363, 2012.

[NT14] R. Noack and B. Thalheim. Multi-dimensional schema composition for conceptual modelling in the large. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 25–44. IOS Press, 2014.

[NT15] I. Nissen and B. Thalheim. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter The Notion of a Model, pages 615–618. De Gryuter, Boston, 2015.

[ST10] M. Skusa and B. Thalheim. Kohärente Softwareentwicklung - Grundlagen, Arbeitsumgebungen, Vorgehensweisen. Preprint 1018, Department of Computer Science, Kiel University, December 2010.

[ST12] E. Sivogolovko and B. Thalheim. Semantic approach to cluster validity notion. In *ADBIS (2)*, volume 186 of *Advances in Intelligent Systems and Computing*, pages 229–239. Springer, 2012.

[ST13] O. Sörensen and B. Thalheim. Semantics and pragmatics of integrity constraints. In *SDKB'11, LNCS 7693*, pages 1–17. Springer, 2013.

[ST15] K.-D. Schewe and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter Co-Design of Web Information Systems, pages 293–332. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.

[ST17] V. Storey and B. Thalheim. Conceptual modeling: Enhancement through semiotics. In *Proc. ER'17*, LNCS, 10650, page forthcoming, Cham, 2017. Springer.

[TD15] B. Thalheim and A. Dahanayake. A conceptual model for services. In *Invited Keynote, CMS 2015, ER 2015 workshop*, LNCS 9382, pages 51–61, Berlin, 2015. Springer.

[TD16] B. Thalheim and A. Dahanayake. Comprehending a service by informative models. In *Conceptual Modeling of Services*, LNCS 10130, pages 87–108, Berlin, 2016. Springer.

[TELT14] S. Torge, W. Esswein, S. Lehrmann, and B. Thalheim. Categories for description of reference models. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 229–240. IOS Press, 2014.

[TFTL^{+}14] M. Tropmann-Frick, B. Thalheim, D. Leber, G. Czech, and C. Liehr. Generic workflows - a utility to govern disastrous situations. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 417–428. IOS Press, 2014.

[Tha] B. Thalheim. Normal models and their modelling matrix. In *Formal foundations of technology*.

[Tha00] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.

[Tha09a] B. Thalheim. Abstraction. In *Encyclopedia of Database Systems*, pages 6–7. Springer US, 2009.

[Tha09b] B. Thalheim. Extended entity-relationship model. In *Encyclopedia of Database Systems*, pages 1083–1091. Springer US, 2009.

[Tha09c] B. Thalheim. Specialization and generalization. In *Encyclopedia of Database Systems*, pages 2746–2747. Springer US, 2009.

[Tha09d] B. Thalheim. Towards a theory of conceptual modelling. In *ER Workshops*, volume 5833 of *Lecture Notes in Computer Science*, pages 45–54. Springer, 2009.

[Tha10a] B. Thalheim. Model suites for multi-layered database modelling. In *Information Modelling and Knowledge Bases XXI*, volume 206 of *Frontiers in Artificial Intelligence and Applications*, pages 116–134. IOS Press, 2010.

[Tha10b] B. Thalheim. 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`.

[Tha11a] B. Thalheim. *Anwendungsorientierte Organisationsgestaltung*, chapter The Culture and Art of Conceptual Modelling, pages 127–144. baar, Hamburg, 2011.

[Tha11b] B. Thalheim. Integrity constraints in (conceptual) database models. In *The Evolution of Conceptual Modeling*, volume 6520 of *Lecture Notes in Computer Science*, pages 42–67, Berlin, 2011. Springer.

[Tha11c] B. Thalheim. The science of conceptual modelling. In *Proc. DEXA 2011*, volume 6860 of *LNCS*, pages 12–26, Berlin, 2011. Springer.

[Tha11d] B. Thalheim. 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*, chapter 17, pages 547–580. Springer, Berlin, 2011.

[Tha12a] B. Thalheim. The art of conceptual modelling. In *Information Modelling and Knowledge Bases XXII*, volume 237 of *Frontiers in Artificial Intelligence and Applications*, pages 149–168. IOS Press, 2012.

[Tha12b] B. Thalheim. The notion of a conceptual model. In N. Seyff and A. Koziolek, editors, *Modelling and Quality in Requirements Engineering*, pages 21–30. Verlagshaus Monsenstein und Vannerdat, 2012. ISBN 978-3-86991-740-5.

[Tha12c] B. Thalheim. The science and art of conceptual modelling. In A. Hameurlain et al., editor, *TLDKS VI*, LNCS 7600, pages 76–105. Springer, Heidelberg, 2012.

[Tha12d] B. Thalheim. Syntax, semantics and pragmatics of conceptual modelling. In *NLDB*, volume 7337 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2012.

[Tha13a] B. Thalheim. The definition of the (conceptual) model. In *Proc. EJC 2013*, pages 256–269, Nara, Japan, 2013.

[Tha13b] B. Thalheim. Open problems of information systems research and technology. In *Invited Keynote, BIR'2013. LNBIB 158*, pages 10–18. Springer, 2013.

[Tha14] B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.

[Tha15] B. Thalheim. Das Modell des Modelles. *Erwägen-Wissen-Ethik*, EWE-Heft, Heft 3, 2015, 26. Jg.:407–409, 2015.

[Tha17a] B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In *Encyclopedia of Database Systems*. 2017.

[Tha17b] B. Thalheim. General and specific model notions. In *Proc. ADBIS'17*, LNCS 10509, pages 13–27, Cham, 2017. Springer.

[Tha17c] B. Thalheim. Model-based engineering for database system development. In *Conceptual Modeling Perspectives*, pages 137–153, Cham, 2017. Springer.

[TN15] B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*. De Gruyter, Boston, 2015.

[TT13] M. Tropmann and B. Thalheim. Mini story composition for generic workflows in support of disaster management. In *DEXA 2013*, pages 36–40. IEEE Computer Society, 2013.

[TTF15] B. Thalheim and M. Tropmann-Frick. The conception of the conceptual database model. In *ER 2015*, LNCS 9381, pages 603–611, Berlin, 2015. Springer.

[TTF16a] B. Thalheim and M. Tropmann-Frick. Model capsules for research and engineering networks. In *New Trends in Databases and Information Systems*, volume 637 of *CCIS*, pages 189–200. Springer, 2016.

[TTF16b] B. Thalheim and M. Tropmann-Frick. Models and their capability. In C. Beierle, G. Brewka, and M. Thimm, editors, *Computational Models of Rationality*, volume 29 of *College Publications Series*, pages 34–56. College Publications, 2016.

[TTF16c] B. Thalheim and M. Tropmann-Frick. Wherefore models are used and accepted? The model functions as a quality instrument in utilisation scenarios. In I. Comyn-Wattiau, C. du Mouza, and N. Prat, editors, *Ingenierie Management des Systemes D'Information*, 2016.

[TTFZ14] B. Thalheim, M. Tropmann-Frick, and T. Ziebermayr. Application of generic workflows for disaster management. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 64–81. IOS Press, 2014.

Conceptual Model
The Notion of a Model in Conceptual Modeling

Bernhard Thalheim
Christian-Albrechts University Kiel, http://www.is.informatik.uni-kiel.de/∼thalheim/models.htm

SYNONYMS

Modeling, Model

DEFINITION

A model is a well-formed, adequate and dependable artefact that represents other artefacts based on criteria of adequacy and dependability commonly accepted by its community of practice within some context. A conceptual model incorporates concepts into the model. A conceptual database model is a conceptual model that represents the structure and the integrity constraints of a database within a given database system environment.

A model has - as an artefact - its background with a undisputable grounding of the sub-discipline and with a basis consisting of chosen elements from the sub-discipline. A model is functioning if it is combined with utilisation/deployment methods. A functioning model is effective if it can be successfully deployed according to its deployment scenarios and its portfolio. They thus function in the utilisation scenario and use spectrum.

MAIN TEXT

Conceptual modeling widely uses models for construction of (database or information) systems. It is a widely applied practice and has led to a large body of knowledge on constructs that might be used for modelling and on methods that might be useful for modelling.

Models are artifacts that are well-formed, adequate and dependable within a given context for some community of practice. They satisfy a purpose (or goal or function in some utilisation scenario). The profile of an artifact is based on the goal or purpose or function of the artifact. Models represent other artifacts or origins. For instance, they image or describe some reality and serve as a prescription for development of a (database) system.

Conceptual models are models that incorporate concepts into the model. Concepts are used as semantical units for classification. Model elements are associated with the concept's names. A concept is also typically given through an embedding into the application domain and into the knowledge space.

Models function as an instrument in some utilisation scenario. The main *function* of a conceptual database model is the description-prescription function. In this case, the instrument is used as a mediator between a reality and an augmented reality that developers of a database system intend to build. The application of a model in a utilisation scenario is initiated by a goal or purpose that is agreed within some community of practice in some context. Other functions of a model despite the description-prescription function are the explanation, the optimisation-variation, the validation-verification-testing, the reflection-optimisation, the explorative, the hypothetical, and the documentation-visualisation functions.

An artefact is well-formed if it satisfies a well-formedness criterion. If the artefact is devoted to its profile then the artefact is called purposeful. A well-formed artifact is *adequate* for a collection of artifacts if it is analogous to the artifacts to be represented according to some analogy criterion within the analogy threshold, it is more focused (e.g. simpler, truncated, more abstract or reduced) than the artifacts within the given focus for some focus criterion, and it is purposeful for the given profile.

An artifact is justified, i.e. (i) by an empirical corroboration (according to purpose of its use, background, etc.) for the representation of the artifacts that is supported by some argument calculus, (ii) by rational coherence and conformity explicitly stated through formulas, (iii) by falsifiability that can be given by an abductive and/or inductive logical system, and (iv) by stability and plasticity (depending on the scope, grounding, basis, context and quality) explicitly given through formulas. The artifact is sufficient by its quality characterisation for internal quality, external quality and quality in use or through quality characteristics such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc.

Encyclopedia of Database Systems, 2017

Justification and sufficiency characterise the signification of an artifact for deployment, reliability and degree of precision efficiency for satisfying the deployment necessities, and extent of coverage depending on deployment. It is typically combined with some assurance evaluation.

A well-formed artefact is *dependable* for some of the justification properties and some of the sufficiency characteristics if the quality criteria are satisfied based on the assurance evaluation, and it is justified by a justification and based on the assurance evaluation.

A *model* is a well-formed, adequate and dependable artifact that represents other artifacts based on criteria of adequacy and dependability commonly accepted by its community of practice within some context.

Any artefact can be used as a model. It faithfully represents other artifacts and must provide facilities or methods for its use. An artefact is implicitly based on its background consisting of (I) an undisputable and well-accepted grounding from one side, i.e. paradigms, postulates, restrictions, theories, culture, foundations, conventions, commonsense, basement, authorities, and (II) a basis from other side, i.e. concepts, conceptions, assumptions, foundations, language as carrier, routine, school of thought, thought community, pattern, methodology, good practices, guidelines, and the cargo. The basis is negotiable.

The model and the artifact are *functional* if there are methods for utilisation of the artifact in dependence on the profile of the artifact. Artifacts are used for application cases that are supported by the task portfolio which an artifact might serve. Typical tasks include defining, constructing, exploring, communicating, understanding, replacing, substituting, documenting, negotiating, replacing, reporting, and accounting. We call an artifact and a model *effective* if it can be deployed according to its portfolio.

Models satisfy typically properties:

(1) *Mapping* property: each model has an origin and is based on a mapping from the origin to the artifact.

(1') *Analogy* property: the model is analogous to the origins based on some analogy criterion.

(2) *Truncation* or reduction property: the model lacks some of the ascriptions made to the origin and thus functions as an Aristolean model by abstraction by disregarding the irrelevant.

(3) *Pragmatic* property: the model use is only justified for particular model users, the tools of investigation, and the period of time.

(4) *Amplification* property: models use specific extensions which are not observed in the original.

(5) *Distortion* property: models are developed for improving the physical world or for inclusion of visions of better reality, e.g. for construction via transformation or in Galilean models.

(6) *Idealisation* property: modelling abstracts from reality by scoping the model to the ideal state of affairs.

(7) *Carrier* property: models use languages and are thus restricted by the expressive capacity of these languages.

(8) *Added value* property: models provide a value or benefit based on their utility, capability and quality characteristics.

(9) *Purpose* property: models and conceptual models are governed by the purpose. The model preserves the purpose.

CROSS REFERENCE

I. Data Model
    a. Semantic data model
    b. Conceptual Modeling
    c. Entity-relationship model
    d. Conceptual Data Model
II. Database design
    a.Conceptual schema design

REFERENCES

B. Thalheim. The conceptual model ≡ an adequate and dependable artifact enhanced by concepts. *Series Frontiers in Artificial Intelligence and Applications,* 260, 241–254. IOS Press, 2014.

B. Thalheim. The conceptual framework to user-oriented content management. *Series Frontiers in Artificial Intelligence and Applications,* 154, 30–49. IOS Press, 30–49, 2007.

B. Mahr. Information science and the logic of models. *Software and System Modeling,* 8(3):365–383, 2009.

Model Capsules
for Research and Engineering Networks

Bernhard Thalheim and Marina Tropmann-Frick

Department of Computer Science, Christian-Albrechts-University Kiel,
Olshausenstrasse 40, 24098 Kiel, Germany
thalheim@is.informatik.uni-kiel.de

Abstract. Multi-model utilisation is a common practice in many sciences, e.g. computer science. Coherence and co-evolution of models is however still an open problem. Multi-model approaches suffer however from the impedance mismatch due to differences in modelling languages. The collaboration approach is based on preservation of local models and on explicit association of derived sub-models. Each discipline has developed its specific know-how in modelling and model deployment. Models evolve in dependence on the progress of the research work. If a model or one of its sub-models has been exchanged with a team member then this evolution must also be applied to models of the partner if those sub-models are used elsewhere.

We develop a novel approach to multi-model development and utilisation, to common use and utilisation of models and modelling experience, to systematic assessment of models and systematic extraction of the potential and capacity of models for a research community, and to the co-evolution of model networks.

1 Introduction

1.1 Complex Problems are Solved in Interdisciplinary Communities

Consider two typical research situations and problems that can be observed for interdisciplinary research in *interacting communities* of researchers.

(1) What are the causes for an inflammatory disease, especially for those triggered by dysfunction at boundary surfaces? Why is it a phenomenon of civilisation? How are cells and tissues infected? What kind of patient-specific treatment can be developed? How can life with a disease be improved and under what circumstances? What societal changes are required to move towards preventive medicine? Many branches of biology, medical science, economy, and social sciences participate in the research team (e.g. the Cluster of Excellence “Inflammation@Interfaces” at CAU Kiel). The collaboration relies on models that are exchanged within the team and that are the basis for a common understanding. The use of models is different. Such teams typically span over all four facets of scientific methods. Models are used in the way how empirical sciences use them, e.g. for exploration, experimentation, interpretation, and hypothesis exploration. At the same time models are used in the setting of theory-oriented sciences for explanations, for exploration, for illustration, for proofs and for concept surveys. In computational science models are used for instance for simulation, for emulation of

CCIS 637, 189--200. Springer, 2016

complex processes, for refinement of a general model by data, and for prognosis. Data sciences use models for detection of pattern, for mining of relations, and for generation of hypotheses. Models are the main exchange instrument for scientists in such teams.

(2) How climate is going to change in the future? How much will this change affect daily life? How should society and politics respond today? In order to answer such questions, teams with different backgrounds and from different sciences must be brought together. Teams must have in-depth expertise in their specific area, a common understanding, and a culture of collaboration. Each discipline and team member uses a specific background, a specific way of working, a specific language and a specific manage data and information. Team members need to exchange their insights and knowledge through models if they are to be easily understood and integrated in a multi-disciplinary manner. Reliable judgements would have to be made for example for climate change forecasting. To this date, this collaboration is not satisfactorily supported; resolving this issue will be a major research breakthrough.

The same situation can also be observed in Computer Engineering. Large systems typically consist of several components. They are developed in teams where a team member solves a certain development task with a specific scope and with an appropriate model. For instance, UML proposes several dozens of diagram languages for system development, e.g. use case, class, object, activity, package, interaction, sequence, time diagrams. Models developed vary in their scopes, aspects and facets they represent and their abstraction. Multi-modelling [3, 11, 20, 23, 24] is a culture in computer science. Maintenance of coherence, co-evolution, and consistency among models has become a bottleneck in development.

1.2 Multi-Modelling is the State-of-the-Art in Research and Engineering

Disciplines often use a combination of empirical research that mainly describes natural phenomena, of theory-oriented research that develops concept worlds, of computational research that simulates complex phenomena and of data exploration research that unifies theory, experiment, and simulation [10]. All these research methods use models as one of their main instruments. Typically, a suite or ensemble of models is simultaneously used due to the complexity of the real world, due to orientation on some of the aspects and facets, due to the abstraction level that fits best to the investigation goal, and due to the supporting instruments such as mathematics and visualisation.

Most disciplines integrate a variety of models or a *society of models*, e.g. [2, 14]. Models used in computer science are mainly at the same level of abstraction. It is already well-known for threescore years that they form a *model ensemble* (e.g. [8, 21]) or *horizontal model suite* (e.g. [3, 26]).

One of the main obstacles beside coherence of models is co-evolution of models within a model suite. However, this can be supported by strict or eager binding with some toleration of deviation. Coherence can be based on collaboration modi such as master-slave or handshake protocols. It is however an unsolved problem how shared elements can be managed within a model suite. At present, models are in some kind of coopetition (cooperation and competition) within a model suite. Often different languages, different backgrounds and different modelling styles are used and are not harmonised.

1.3 Overview of the Approach

In this paper we tackle the collaboration challenge by developing a flexible system to manage locally and to exchange globally models for collaboration in networks. In this case, models become thus a crosscutting concern to reflect competence for an interdisciplinary collaboration and for interactive research on complex society issues that cannot be solved within a singleton discipline.

We remind in Section 2 a novel notion of the model. This notion generalises the notions of models used in archeology, arts, biology, chemistry, computer science, economics, electrotechnics, environmental sciences, farming and agriculture, geosciences, historical sciences, humanities, languages and semiotics, mathematics, medicine, ocean sciences, pedagogical science, philosophy, physics, political sciences, sociology, and sport science.

Next we discuss in Section 3 model-based collaboration in research and development. This research can be supported by model suites. They establish coherence maintenance among models. The main novel contributions of the paper are the introduction of the notion of the **model capsule** in Section 4 and the proof of concept in Section 5.

2 The Notion of the Model

Disciplines have developed a different understanding of the notion of a model, of the function of models in scientific research and of the purpose of the model. Many different notions are used, e.g. [4, 12, 18]. There is however not yet a general notion of a model. Our definition of a model [32] summarises the bottom-up approach to models and modelling developed at CAU Kiel.

Models are often language based. Their syntax uses the namespace and the lexicography from the application domain. Semantics is often implicit. The lexicology can be inherited from the application domain and from the discipline. Models do not need the full freedom for interpretation. The interpretation is governed by the purpose of the model within the research scenario, is based on disciplinary concerns (postulates, paradigms, foundations, commonsense, culture, authorities, etc.) and is restricted by disciplinary practices (concepts, conceptions, conventions, thought style and community [5], good practices, methodology, guidelines, etc.). Models combine at least two different kinds of meaning in the namespace: referential meaning establishes an interdependence between elements and the origin ('what'); functional meaning is based on the function of an element in the model ('how'). The pragmatics of a model depends on the community of practice, on the context of the research task and especially on the purpose or function of the model.

2.1 A Model is a Well-Formed, Adequate and Dependable Instrument

A **model** *is a well-formed, adequate and dependable instrument that represents origins.*

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

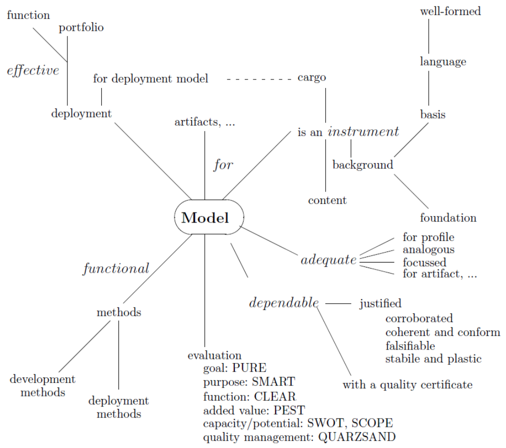
The model should be well-formed according to some well-formedness criterion. As an instrument or more specifically an artifact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form.

A well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*.

Well-formedness enables an instrument to be *justified* by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through formulas, by falsifiability, and by stability and plasticity.

The instrument is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics [31] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed instrument is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.



The model has a profile (goal or purpose or function), represents artifacts and is used for some deployment scenario. As an instrument, a model has its own background (e.g. foundation (paradigms, postulates, theories, disciplinary culture, etc.) and basis (concepts, language, assumptions, practice, etc.)). It should be well-defined or well-formed. Adequacy is based on satisfaction of the purpose, analogy to the artifacts it represents and the focus under which the model is used. Dependability is based on a justification for its usage as a model and on a quality certificate. Models can be evaluated by one of the evaluation frameworks. A model is functional if methods for its development and for its deployment are given. A model is effective if it can be deployed according to its portfolio, i.e. according to the tasks assigned to the model. Deployment is often using some deployment model, e.g. for explanation, exploration, construction, description and prescription.

A model can be used for different purposes and various usage scenarios. Therefore, a model is typically also extended by *views* or *viewpoints* that reflect certain parts of the model and that hide details which are not necessary. This reflection is often only provided in a non-systematic or implicit way. Additionally, we need a refinement notion, methods for combination and for evaluation of models.

2.2 Models as a Means in Research and Engineering Networks

A common understanding of the nature of models, of the methods and techniques that are used for model development and model deployment and of systematic approaches to modelling enables also model-based collaboration in networks.

Models are built and modelling is performed in a similar form, with similar background and theories and within similar investigation scenarios despite the variety of models, the variety of purposes, the complexity, the range from micro to macro, and the variety of solutions. Each discipline has been developing also specific solutions to modelling and model deployment. These solutions may also be used for other disciplines, may be combined with their solutions, or may replace their solutions.

Our notion of the model has been validated and verified against the model notions of many disciplines. The validation [33] brought an insight into the specific understanding of adequacy, dependability, functioning, and effectiveness used in each of these disciplines. The validation has also resulted in an understanding of the added value of the model within the discipline, in an evaluation of the model maturity, in detection of features which are missing and should be added to the model or which can be deleted from the model, and in restrictions to model deployment which must be observed.

In Section 4 the notion of the model is generalised in order to cope with the requirements for model-based collaboration.

2.3 Local-As-Design for Disciplinary Models

Disciplines have however also their own foundation, their own background, their own culture and their own way of model use. Therefore, it is infeasible to develop a holistic model for everybody in a research team. Models should remain in their local setting and should not be integrated into general global model that is commonly agreed and used. For this reason, we prefer a *local approach*. A model remains within its local environment. It is however enhanced in such a way that it can be used in a collaboration and thus support exchange of ideas and results. This local approach is similar to the global-as-view integration approach used for integration of database systems. The model enhancement needs however a *generalised-global-as-view* approach. The collaboration environment thus supports a peer-to-peer exchange of exchange sub-models.

3 Model-Based Collaboration in Interacting Communities

3.1 Models — The “Intergalactic” Communication Instrument

Collaboration on the basis of models preserves local models which have sub-models for collaboration activities thus providing an explicit association of derived sub-models. Models vary in their abstraction, their foci and scales, their scopes, their aspects and their purposes. They are deployed in different scenarios and are backed by heterogeneous data with different granularity and at different levels of abstraction. A model-based collaboration cannot be based on an exchange of models as they are. Models must be fitted to the partner. We use typically parts or abstractions of a model for exchange. This model transformation is not yet performed in a systematic manner. We might however develop an algebra for such model transformations. In this case we can generate derivatives or exchange sub-models of a model.

If a derivative of a model is used for exchange then the model of the partner can incorporate the derivative. The derivative is typically transformed to the model. It is then integrated into the model that is under revision in such collaboration activities. The derivative is associated to a sub-model of the new model. This association can be the basis for future communication. Modelling itself becomes now teamwork.

Models are simultaneously used in interdisciplinary teams for different interleaved purposes. For instance, a conceptual model of an information system is used for construction and inspiration in an implementation phase, for planning and resource allocation, for verification against the requirements model, for optimisation of the structure, for prognosis of behaviour of the system that is under construction, for explanation and understanding its components, and as the basis for system integration. Each of these functions can be used by different stakeholders at the same time. Typically, only some of model elements are of interest to different team members. These members should be better supported by specific views defined on top of the model. These views should be defined in dependence on the viewpoints that are requested by the partner. If the model is changed then these views must also be changed and the change must be communicated in an appropriate form. Model views are therefore exported to partners.

A change in one model may also result in a change to models of collaborators. The changes should be integrateable into the model. The result of integration by a partner should be communicated within a research team. Model views that are derived from one model are imported into model views of another model. Since models might use different languages the model view that is exported by one model must be transformed before integration into another model.

3.2 Model Suites

Model suites are an extension of model ensembles [22] used for distributed or collaborating databases [25].

A model suite [3, 26] consists

- of set of models $\{M_1, ..., M_n\}$,
- of an association or collaboration schema among the models,
- of controllers that maintain consistency or coherence of the model suite,
- of application schemata for explicit maintenance and evolution of the model suite, and
- of tracers for the establishment of the coherence.

Coherence describes a fixed relationship between the models in a model suite.

The collaboration style of a model suite is based on supporting programs, data access pattern, style of collaboration, and coordination workflows. Collaboration pattern generalize protocols and their specification [16].

Let us assume that a model is defined in a language that uses constructors $\mathfrak{C}$ for the structuring and defining a model M, i.e. $M \in Term(\mathfrak{C})$ for the set of all terms defined in $\mathfrak{C}$. These constructor can be combined with an algebra $\mathfrak{A}$ of expressions defined over $\mathfrak{C}$. Typical operations of the algebra are set operations such as union, difference and intersection, constructing operations such as join, projection, selection, nesting (or integration/combination) and unnesting (or disintegration), and abstraction (or more specifically aggregation) operations. This approach to algebras follows approaches for universal algebras [19].

Each operation can be classified as either an *identification-preserving* or an *identification-loosing* one. Identification-preserving operations are, for instance, difference, intersection, nesting, and unnesting. An expression is identification-preserving if all its sub-expressions have this property.

We may use additional identification auxiliaries, i.e. constructions that define together with the given construction an identification-preserving expression.

A sub-model of M can be either defined as a sub-expression of the expression that defines M or as the application of an expression $E(M) \in \mathfrak{A}(\mathfrak{C})$ with one free variable M. For collaboration networks we choose the second approach and call them exchange sub-models. Sub-models can be identity-preserving or identity-loosing. Given any model or sub-model, an expression $E(M)$ defined on this model can also be considered as a mapping from M to the resulting structure $E(M)$. Furthermore, we can use infomorphisms [9, 28] among models. Two models M_1, M_2 are E_1, E_2*-infomorph* though two transformations E_1, E_2 with $E_1(M_1) = M_2$ and $E_2(M_2) = M_1$ if any object o defined on M_i can be mapped via E_i to objects defined on M_j for $i, j \in \{1, 2\}$, $i \neq j$.

4 The Model Capsule

4.1 The Model Capsule $\equiv$ Model $\bigoplus$ Exchange Sub-Models

Models are extended by sub-models that are either

1. *abstractions* of the given model similar to roll-up or aggregation techniques used in database technology [17] or
2. *specialisations* to a more specific model similar to refinement techniques used for abstract state machines [1] or
3. *specific viewpoints* of the given model similar to view schemata [29].

Sub-model specification is based on an algebra for abstraction, refinement and filtering. The algebra is also used for transformation of sub-models. Sub-models to be exported to another model can be transformed before becoming imported by another model.

A **model capsule** consists of a main model and many exchange sub-models. An exchange sub-model is either an export or an import exchange-sub-model. If it is an import sub-model then it must be identity-preserving[1].

Exchange sub-models are used as mediator in research teams and provide all details that are necessary for collaboration (completeness) but only those (minimality). Exchange sub-models are either derived from the main model in dependence on the viewpoint, on foci and scales, on scope, on aspects and on purposes of partners or are sub-models provided by partners and transformed according to the main model. A team member thus can integrate an exchange sub-model in his/her main model, can propagate changes made by him/herself to other partners and can change the main model

---

[1] This restriction can be weakened if additional identification auxiliaries are used.

according to changes by partners. This model capsule is the main communication vehicle for collaboration. The propagation and transformation from and to partners can be based on contracts or protocols.

4.2 Collaboration Model Capsules

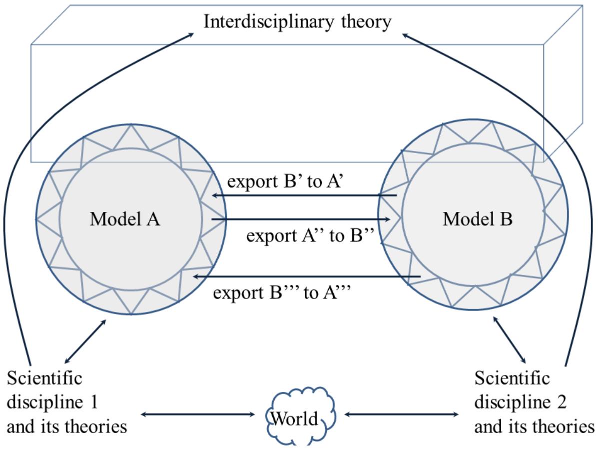
Model suites can be associated to each other based on exchange sub-models.

$$\begin{tikzcd}
M_1 \arrow[r, "\texttt{extract}", "E_{1,2}"'] & M_{1,2} \arrow[d, "\texttt{transform}"', "E^T_{1,2}"] & \\
& M^T_{1,2} \arrow[d, "\texttt{infomorphism}"', "{E_{2,1,2}, E_{2,2,1}}"] & \\
& M_{2,1} \arrow[r, leftrightarrow, "\texttt{exchange}", "E_{2,1}"'] & M_2
\end{tikzcd}$$

Given model suites $\mathcal{M}_i = (M_i, M_{i,1}, ...M_{i,n_i})$ with n_i exchange sub-models. A model suite $\mathcal{M}_1$ is bound to a model suite $\mathcal{M}_2$ via export/import sub-models $M_{1,2}$ and/or import sub-models $M_{2,1}$ if there exist expressions $E_{1,2}$, $E^T_{1,2}$, $E_{2,1,2}$, $E_{2,2,1}$, $E_{2,1}$ such that $E_{1,2}$ extracts the sub-model $M_{1,2}$ from M_1, the transformation expression transforms this sub-model to a model $M^T_{1,2}$ that is infomorph to the import sub-model $M_{2,1}$ of M_2, i.e. formally

· $E_{1,2}(M_1) = M_{1,2}$,
· $E_{2,1}(M_2) = M_{2,1}$,
· $E^T_{1,2}(M_{1,2}) = M^T_{1,2}$, and
· $M^T_{1,2}$ and $M_{2,1}$ are $E_{2,1,2}, E_{2,2,1}$-infomorph.

We notice without proof that the infomorphism can be integrated into the transformation expressions for some special cases.



Expressions we use for model association may be, for instance, aggregation or abstraction expressions, viewpoint expressions, specialisation expressions, or also combination expressions. Therefore, a sub-model of a first model that is used for association with a second model may be more abstract, or may be oriented on specific elements of the first model, or may extend the first model. Abstraction allows to form a kind of generalisation, i.e. a vertical hierarchy. The model capsule is bound vertically. Specific or extended models are typically defined on the same level of abstraction. The model capsule is then bound horizontally.

This approach is sufficiently general for model-based communication and reasoning in interacting research and engineering communities. Each branch of engineering or science uses its specific model suite. In order to collaborate, an interdisciplinary theory is formed. The interdisciplinary theory corresponds to the association in the real world. For instance, model capsules are based on models A and B that use corresponding scientific disciplines and corresponding theories as a part of their background. The models have three derived exchange sub-models that are exported to the other capsule and that are integrated into the model in such a way that the imported sub-model can be reflected by the model of the capsule. The two models and the two scientific disciplines are the kernel for an interdisciplinary theory.

5 Realisation and Implementation of the Approach

Model suites have already been investigated for UML-based software engineering in [26] on the basis of [30]. M. Skusa investigated the association among modelling languages based on language mappings. Each of the diagram types got its own profile. These profiles have been used for automatic derivation of associations among UML diagrams. The direction of enforcement follows in this case waterfall development strategies, i.e. requirements diagrams cannot be changed by conceptual diagram changes. He also developed controllers that maintain consistency of diagrams within a model suite. These controllers have been written as rules based on Abstract State Machines [1]. Since ASM rules run in parallel all controller run in parallel.

The Extract-Transform-Load paradigm can be enhanced by derivation of functions that provide the basic database system CRUD functionality [34]. Therefore, exchange sub-models support database processing similar to classical technology.

Traditional object-relational approaches only support singleton table views. To overcome this limitation we define a complex view as a collection of views that are associated through integrity constraints - mainly (pairwise) (generalised) inclusion constraints. The view classes are computed in the first step from the basic database using the view expression and then mapped to a database based on the association schema. They thus form a local database on their own.

The concept of view towers [15] has already been used for the generation of interfaces. Views of level i are schemata on their own and are incrementally constructed of the base database schema (level 0) and of views of level less than i. It has been shown that SQL and database technology nicely support such complex views [13]. The construction of view towers can be enhanced by a characterisation whether the view is updatable. A higher level view is strongly updateable if the algebraic expression that defines this view does not destroy updateability and each of its components of lower level is updateable. Views can be enhanced by auxiliary views that provide an enhanced updateability based on a combined view of the original one and the auxiliary ones that is itself updateable.

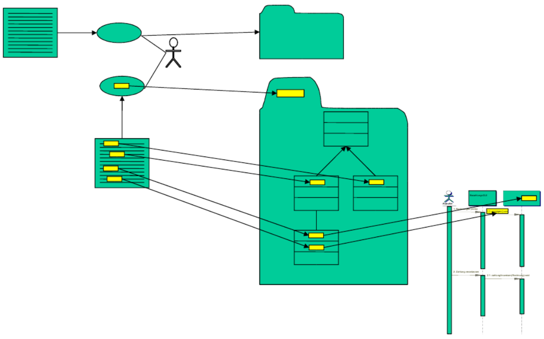
5.1 Realisation 1: Applicability of the Approach in Research Communities

Our approach supports collaboration for more complex applications discussed in Subsection 1.1. We define explicit transformation expressions. The notion of info-morphism becomes then however far more complex. Both research collaborations in the Clusters of Excellence at CAU Kiel are using ad-hoc model associations. In [6] model suites and views have been used for automatic recharge of archives. The development and maintenance of integrated, reusable and coherent archives for all data capturing project is a mandatory requirement issued by the German Research Council to integrated projects such as Clusters of Excellence. In [7], a general data store has been realised for all archeology and pre-historic data. The local projects have their import views to and export views from the general data store. The global data store consists of one component. This component contains all data from all projects in the Graduate School "Human Development of Landscapes" and a pair of an import and an export view for each of the projects. Project collaboration is based on collaboration export views for each collaborating community. The projects themselves have their own database schemata that correspond to the import view through an extract-transform-load feature.

Both database support projects are the basis and the background for the model capsule approach developed in this paper.

5.2 Realisation 2: Collaboration Model Capsules in Software Engineering

Let us now exemplify the concept for classical software engineering with an example adapted from [27].



Given a use_case diagram, a class diagram, a package diagram, and an interaction diagram. These four diagrams can be associated by exchange sub-model for a use_case-package association in the upper part and package-interaction, package-class, and class-use_case associations in the lower part. Controllers maintain the coherence of the different viewpoints. In the lower part, we consider the package diagram to be the leading diagram for the class and interaction diagrams and the class diagram as a leading diagram for the use case diagram. The class diagram has an export sub-model to the use_case diagram that has an identity-preserving sub-model as an import sub-model. Controllers may use a restrict, eager or lazy approach, i.e. a change in the class-diagram is allowed

· only if this can be directly reflected in the use_case sub-model (restrict) or if this changed be directly (eager) or at a later stage (lazy) propagated to the importing sub-model of the use_case model and

· the change modifies the export sub-model in the class diagram.

The application schemata are derived from controllers based on templates or pattern similar to integrity maintenance for referential inclusion constraints in databases. Tracers are then small demons that observe whether a model changes its export and import models.

6 Conclusion

This paper proposes an approach to models and model-based reasoning for interacting research and engineering communities. Models are an *"intergalactic" communication*

CCIS 637, 189--200. Springer, 2016

and reasoning instrument and a *crosscutting concern* in such networks. Model-based communication and reasoning is based on the concept of the model capsule that provides a flexible and powerful mechanism for model-based reasoning and collaboration. They provide a flexible system to manage locally and to exchange globally models for collaboration in teams. Models thus become a crosscutting concern to reflect competence for an interdisciplinary collaboration and for interactive research on complex society issues that cannot be solved within a singleton discipline.

The role and potential of models in networked research communities has not yet been systematically investigated, explored and generalised. This paper tackles the collaboration challenge based on model-based data exchanging collaboration. Model-based collaboration is only one kind of collaboration beside the data-based, concept-based, workpiece-based, process-oriented etc. collaborations. It seems however that models are a central instrument for any qualified and dependable collaboration.

As the next step, we aim at a general model description language ModelML that allows to collect models in networks in a form similar to an online interactive encyclopedia or model web. This model web supports systematic elicitation and exploration of modelling experience in research networks.

References

1. E. Börger and R. Stärk. *Abstract state machines - A method for high-level system design and analysis*. Springer, Berlin, 2003.
2. A. Coleman. Scientific models as works. *Cataloging & Classification Quarterly, Special Issue: Works as Entities for Information Retrieval*, 33, 3-4, 2006.
3. A. Dahanayake and B. Thalheim. Co-evolution of (information) system models. In *EMMSAD 2010*, volume 50 of *LNBIP*, pages 314–326. Springer, 2010.
4. U. Dirks and E. Knobloch, editors. *Modelle*. Peter Lang, 2010.
5. L. Fleck. *Denkstile und Tatsachen, edited by S. Werner and C. Zittel*. Surkamp, 2011.
6. D. Fleischer and K. Jannaschk. A path to filled archives. *Nature Geoscience*, (4):575 – 576, 2011.
7. F. Förster and B. Thalheim. An effectual approach for a data and information management for humanists. *Qualitative and Quantitative Methods in Libraries (QQML)*, (2):121–128, 2012.
8. N.P. Gillett, F.W. Zwiers, A.J. Weaver, G.C. Hegerl, M.R. Allen, and P.A. Stott. Detecting anthropogenic influence with a multi-model ensemble. *Geophys. Res. Lett.*, 29:31–34, 2002.
9. J. Goguen. Three perspectives on information integration. Kalfoglou, Y., et al., eds., Semantic Interoperability and Integration. Dagstuhl Seminar Proceedings 04391, 2005.
10. J. Gray. eScience: A transformed scientific method. Technical report, Talk given Jan 11, 2007. Edited by T. Hey, S. Tansley, and K. Tolle. http://research.microsoft.com/en-us/um/people/gray/talks/NRC-CSTB_eScience.ppt, Microsoft Research Publications, 2007.
11. E. Guerra, J. de Lara, D.S. Kolovos, and R.F. Paige. Inter-modelling: From theory to practice. In *MoDELS 2010, LNCS 6394*, pages 376–391, Berlin, 2010. Springer.
12. I.A. Halloun. *Modeling Theory in Science Education*. Springer, Berlin, 2006.
13. F. Holst. Konzeptuelle Sichten. Master's thesis, CAU Kiel, Institut für Informatik, Kiel, Juli 2015.
14. P. J. Hunter, W. W. Li, A. D. McCulloch, and D. Noble. Multiscale modeling: Physiome project standards, tools, and databases. *IEEE Computer*, 39(11):48–54, 2006.

15. H. Jaakkola and B. Thalheim. Multicultural adaptive systems. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 172–191. IOS Press, 2014.
16. H. König. *Protocol Engineering: Prinzip, Beschreibung und Entwicklung von Kommunikationsprotokollen*. Teubner, Stuttgart, 2003.
17. H.-J. Lenz and B. Thalheim. A formal framework of aggregation for the OLAP-OLTP model. *J. UCS*, 15(1):273–303, 2009.
18. L. Magnani, W. Carnielli, and C. Pizzi, editors. *Model-Based Reasoning in Science and Technology: Abduction, Logic, and Computational Discovery*. Springer, 2010.
19. A. I. Malzew. *Algebraic systems*. Nauka, Moscow, 1970.
20. J. Pardillo. A systematic review on the definition of UML profiles. In *MoDELS 2010, LNCS 6394*, pages 407–422, Berlin, 2010. Springer.
21. M. Pottmann, H. Unbehauen, and D.E. Seborg. Application of a general multi-model approach for identification of highly nonlinear processes – a case study. *Int. Journal of Control*, 57(1):97–120, 1993.
22. S. Prochnow and R. von Hanxleden. Statechart development beyond WYSIWYG. In *MoDELS*, volume 4735 of *Lecture Notes in Computer Science*, pages 635–649. Springer, 2007.
23. B. Rumpe. *Modellierung mit UML*. Springer, 2012.
24. R. Salay, J. Mylopoulos, and S. M. Easterbrook. Using macromodels to manage collections of related models. In *CAiSE*, volume 5565 of *Lecture Notes in Computer Science*, pages 141–155. Springer, 2009.
25. K.-D. Schewe and B. Thalheim. Development of collaboration frameworks for web information systems. In *IJCAI'07 (20th Int. Joint Conf on Artificial Intelligence, Section EMC'07 (Evolutionary models of collaboration)*, pages 27–32, Hyderabad, 2007.
26. M. Skusa. *Semantische Kohärenz in der Softwareentwicklung*. PhD thesis, CAU Kiel, 2011.
27. M. Skusa and B. Thalheim. *Kohärente Multi-Modell-Entwicklung. In [31]*, pages 431–454.
28. D. Song and P. Bruza. Discovering information flow using a high dimensional conceptual space. In *Research and Development in Information Retrieval*, pages 327–333, 2001.
29. B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
30. B. Thalheim. The conceptual framework to multi-layered database modelling. In *Proc. EJC*, pages 118–138, Maribor, Slovenia, 2009.
31. B. Thalheim. 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`.
32. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
33. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung*. De Gruyter, Ontos Verlag, Berlin, 2015.
34. M. Tropmann. Funktionale Integration heterogener Datenbanken. Master's thesis, CAU Kiel, Institut für Informatik, Kiel, Januar 2008.

CCIS 637, 189--200. Springer, 2016

Models and their Capability

Bernhard Thalheim and Marina Tropmann-Frick

Department of Computer Science, Christian-Albrechts-University Kiel, 24098 Kiel,
Germany

Abstract. Models are one of the main instruments in Computer Science. The notion of model is however not commonly agreed due to the wide usage of models. It is challenging to find an acceptable and sufficiently general notion of model due to the large variety of known notations. Such notion should incorporate all of the different notations and at the same time should allow to derive the specific notation from the general notion. We introduce a universal parameterised notion of the model. The parameters in this notion support adaptation of the universal notion of the model to the specific notation of interest. We finally apply this notion and this adaptation to development of business process models that are specified in BPMN.

1 The Model - an Artifact and an Instrument

Classical Computer Science research considers models as *artifacts*[1] that are constructed in certain way and prepared for their utilisation according to the purpose under consideration such as construction of systems, verification, optimization, explanation, and documentation.

Creation for a practical purpose means that the main target of model development is its application in utilisation scenarios. Models are considered to be artifacts in a stronger sense. We observe however that models are developed for their utilisation within some scenario. They are functioning in this scenario. That means models are instruments in these scenarios. The notion of an instrument[2] concentrates on this utilisation of models. Models are therefore mainly *instruments* that are effectively functioning within a scenario. The effectiveness is based on an associated set of methods and satisfies requirements of usage of the model.

1.1 Models - The Third Dimension of Science

Models are used as perception models, experimentation models, formal models, conceptual models, mathematical models, computational models, physical

---

[1] An artifact is "something that is created by humans usually for a practical purpose" or "something characteristic of or resulting from a particular human institution, period, trend, or individual" or "a product of artificial character due usually to extraneous (as human) agency" [16]. The last meaning of the notion of an artifact is not taken into consideration for models in most sciences and also in Computer Science.

[2] An instrument is among others (1) a means whereby something is achieved, performed, or furthered; (2) one used by another as a means or aid or tool [16].

Models of Rationality}, volume~29 of {\em College Publications Series, vol 29, 34--56 , 2016

models, visualisation models, representation models, diagrammatic models, exploration models, heuristic models, etc. Experimental and observational data are assembled and incorporated into models and are used for further improvement and adaptation of those models. Models are used for theory formation, concept formation, and conceptual analysis. Models are used for a variety of purposes such as perception support for understanding the application domain, for shaping causal relations, for prognosis of future situations and of evolution, for planning, for retrospection of previous situations, for explanation and demonstration, for preparation of management, for optimisation, for construction, for hypothesis verification, and for control of certain environments.

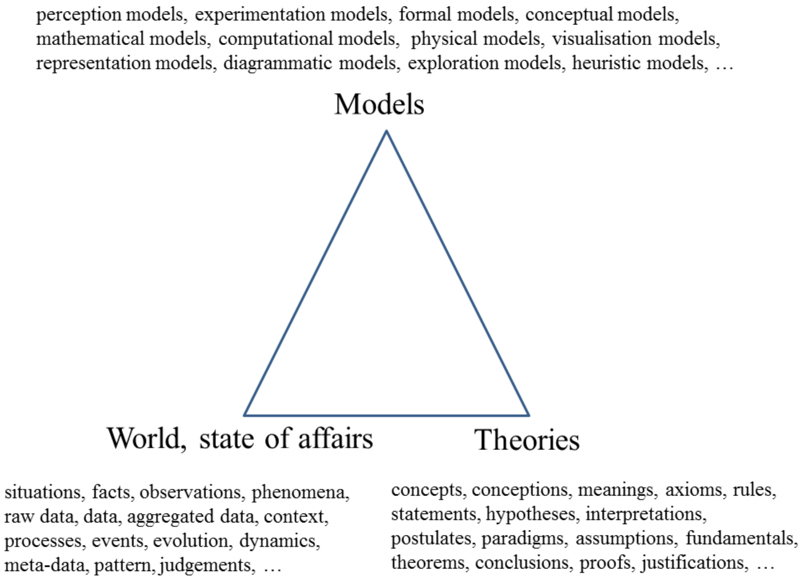


Fig. 1. Models - The third dimension of science

Models are one of the main instruments in scientific research. They are considered to be the *third dimension of science* [26][3] (Figure 1). They provide a tool for research and have an added value in research, e.g. for construction of systems, for education, for the research process itself. Their added value is implicit but can be estimated based on the capability, potential and capacity of the model. Models are common culture and practice in sciences. However, each discipline has developed its own modelling expertise and practice.

Models are often language based. Their syntax uses the namespace and the lexicography from the application domain. Semantics is often implicit. The lexicology can be inherited from the application domain and from the discipline. Models do not need the full freedom for interpretation. The interpretation is governed by the purpose of the model within the research scenario,

---

[3] The title of the book [4] has inspired this observation.

is based on disciplinary concerns (postulates, paradigms, foundations, commonsense, culture, authorities, etc.) and is restricted by disciplinary practices (concepts, conceptions, conventions, thought style and community [6], good practices, methodology, guidelines, etc.). Models combine at least two different kinds of meaning in the namespace: referential meaning establishes an interdependence between elements and the origin ('what'); functional meaning is based on the function of an element in the model ('how'). The pragmatics of a model depends on the community of practice, on the context of the research task and especially on the purpose or function of the model.

A model can be used for different purposes and various usage scenarios. Therefore, a model is typically also extended by views or viewpoints that reflect certain parts of the model and that hide details which are not necessary. This reflection is often only provided in a non-systematic or implicit way. Additionally, we need a refinement notion, methods for combination and for evaluation of models.

1.2 Scenarios of Model Utilisation

Models are used as an instrument in some utilisation scenario. At the same time, the model might be useless and not productive in other scenarios. Their function in these scenarios is a combination of functions such as explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualization, and description-prescription as a mediator between a reality and an abstract reality that developers of a system intend to build.

Traditionally, purposes or goals are considered first. The purposes and the goals are used to determine the functions of a model. This approach is centered around the purpose or goal and requires a definite understanding of the purpose and goal. Purposes and goals are often underspecified or blurry at the beginning. They become more clear after the model is being used. Compared to this approach, it is simpler to understand the application cases of a model and thus the utilisation scenarios. In this case we may derive the functions that a model has in these scenarios. Therefore, we use the approach that the functions of the model determine the *purposes* of the deployment of the model.

1.3 The Storyline of the Paper

The large variety of model notations (see, for instance, [13, 23, 30]) does not allow to transfer experience gained with one notation to other notations. Methods for utilisation or development are therefore mainly bound to one notation. Each subdiscipline has therefore its own understanding of modelling. It would however be beneficial to have a general notion of model that can be adapted to the specific notations of interest.

We introduce in Section 2 a universal notion of a model. This notion is based on the understanding of a model as an instrument in some utilisation scenarios. We only consider well-formed instruments since models must be intuitive and easy to understand. The model definition is based on two general parameter sets, adequateness and dependability. Each of the parameters can be instantiated in dependence of the function that the model should have in a given utilisation scenario within the sub-discipline. This instantiation facility is based on a conception frame for the model notion.

The approach is applied to BPMN modelling in Section 3. We describe the business process modelling approach and derive the capability of this modelling technique. We can now also explicitly describe the obstacles of BPMN modelling. Furthermore, we derive the evaluation procedure for the BPMN approach in Section 4.

This approach to modelling in Computer Science can now be used as a starting point of a theory of modelling (Section 5). We start with some, often implicitly given restrictions that a model has, esp. its burden by the background and by the directives. The evaluation of models also supports a statement on not-supported utilisations, called anti-profile. Finally, the conception frame can also be used for development of question forms that support model specification.

2 The Universal Notion of the Model

There are many notions of models. Each of them covers some aspects and concentrates on some properties such as the mapping, analogy, truncation, pragmatism, amplification, distortion, idealisation, carrier, added value, and purpose properties [11, 17, 18, 21]. The main property is however the *function property: The model suffices in its function in the utilisation scenarios that are requested.* This property results in the following notion of the model [25, 27, 29].

2.1 The Model Notion

Models have several *essential properties* that qualify an instrument as a model [22, 24]:

Definition 1. *An instrument is* well-formed *if it satisfies a well-formedness criterion.*

Definition 2. *A well-formed instrument is* adequate *for a collection of origins if (i) it is analogous to the origins to be represented according to some analogy criterion, (ii) it is more focused (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and (iii) it is sufficient to satisfy its purpose.*

Definition 3. *Well-formedness enables an instrument to be* justified*: (i) by an empirical corroboration according to its objectives, supported by some argument calculus, (ii) by rational coherence and conformity explicitly stated through formulas, (iii) by falsifiability that can be given by an abductive or inductive logic, and (iv) by stability and plasticity explicitly given through formulas.*

Definition 4. *An instrument is* sufficient *by a* quality *characterisation for internal quality, external quality and quality in use or through quality characteristics [20] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).*

Definition 5. *A well-formed instrument is called* dependable *if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.*

Definition 6. *An instrument is called* **model** *if it is* adequate *and* dependable*. The adequacy and dependability of an instrument is based on a* judgement *made by the community of practice.*

Definition 7. *An instrument has a* background *consisting of an undisputable grounding from one side (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities) and of a disputable and adjustable basis from other side (assumptions, concepts, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense).*

Definition 8. *A model is used in a* context *such as discipline, a time, an infrastructure, and an application.*

The model notion can be depicted in Figure 2 based on the following conceptions:

a fundament or ***background*** with
  - the grounding, and
  - the (meta-)basis,

four governing directives given by
  - the artifacts or better origins to be represented by the model,
  - the deployment or profile of the model such as goal, purpose or functions,
  - the community of practice (CoP) acting in different roles on certain rights through some obligations, and
  - the context of time, discipline, application and scientific school,

two pillars which provide
  - methods for development of the model, and
  - methods for utilisation of the model,

and finally

the model utilisation scenario for the deployment of the model in the given application.

The *model house* in Figure 2 abstracted from its full version [24, 27] displays these different facets of the model. The house consists of a cellar (basis in Figure 2) and a fundament (grounding in Figure 2), two pillars (development resp. utilisation methods), four driving or governing forces (origins, purpose of function, community of practice, context), and finally the deployment roof (utilisation scenario). The *grounding* is typically implicitly assumed and not disputable. It contains paradigms, the culture in the given application area, the background, foundations and theories in the discipline, postulates, (juristic and other) restrictions, conventions, and the commonsense. The *basis* is the main part of the background and is typically disputable.

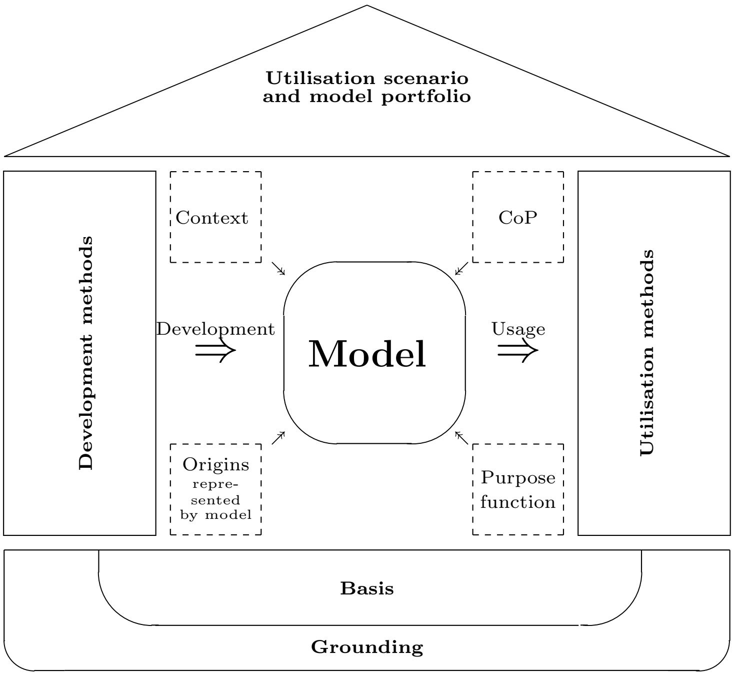


Fig. 2. Facets of the model notion

Definition 9. *A fully-specified model is* function-purpose-goal invariant *if the model can be used instead of the origins in the given scenario and have the same goal, the same purpose, and the same function. A model is* solution-faithful *if the solution of the problem solved with the model is analogous in the world of the origins based on the analogy criterion that is used for stating adequacy.*

2.2 The Conception Frame for the Model Notion

The model notion covers many different aspects. It might thus be of interest whether there is a guideline for development of models. Models are artifacts that can be specified within a W*H-frame [5] that extends the classical rhetorical frame introduced by Hermagoras of Temnos[4]. Models are primarily characterised by W^4: wherefore (purpose), whereof (origin), wherewith (carrier, e.g. language), and worthiness ((surplus) value). The secondary characterisation dimensions are given by: (1) stakeholder: by whom, to whom, whichever; (2) additional properties of the application domain: wherein, where, for what, wherefrom, whence, what; (3) solution: how, why, whereto, when, for which reason; and (4) context: whereat, whereabout, whither, when.

A practical guideline may just

---

[4] Quis, quid, quando, ubi, cur, quem ad modum, quibus adminiculis (Who, what, when, where, why, in what way, by what means), The Zachman frame uses a simplification of this frame.

1. start with fixation of two directives: origins to be represented and community of practice that accepts this model;
2. restrict the model utilisation scenario and the usage model to those that are really necessary and thus derive the purpose and function of the instrument;
3. define adequateness and dependability criteria of the instrument within the decision set made so far;
4. explicitly describe the background of the model, i.e. its undisputable grounding and the selected basis; and
5. explicitly specify the context for utilisation of the model.

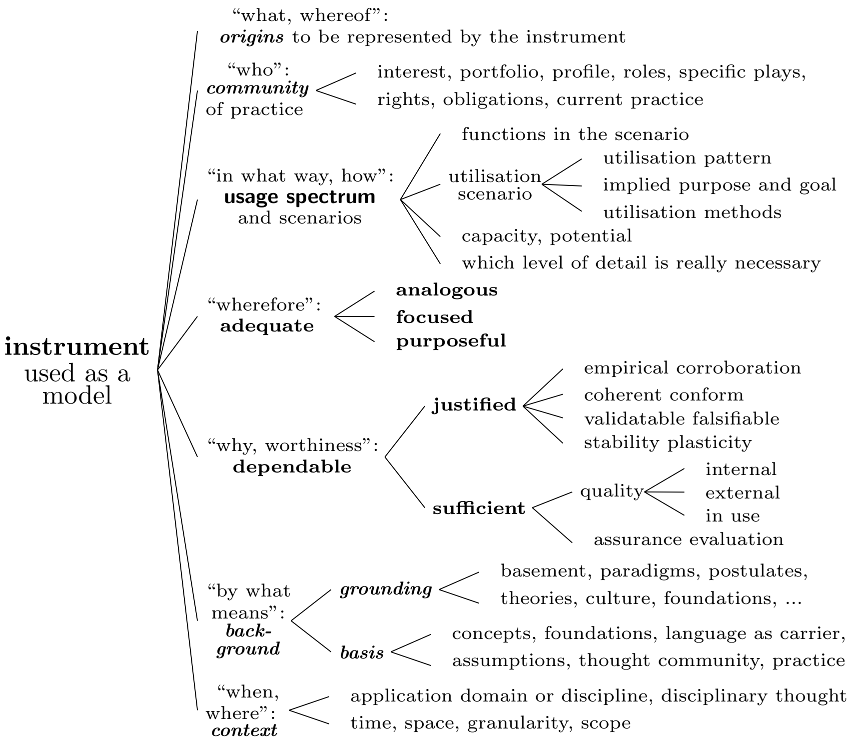


Fig. 3. Conception frame for systematic development of a model

The model development and utilisation depends in this case from:

Judgements of some members of the CoP to deploy the instrument as a model for some origin based on an assessment (deployability, rigidity, modality, confidence) within a CoP, utilisation scenario, and within a context.

Utilisation scenarios and use spectra accepted for the instrument with
functions of the instrument in utilisation scenario,
roles and deployment of the instrument in those scenario, and
resulting purposes and goals for the utilisation.

The instrument as such with some appreciation
  as a well-shaped instrument on the basis
    - of some criteria in dependence on intended utilisation and criteria for:
      - what is accepted in a CoP, and
      - what is syntactically, semantically, pragmatically well-shaped,
  that fits to the intended use, and
  is appropriate for the utilisation scenarios and the use spectra.

The orientation also reflect our understanding of a model as an instrument.

3 BPMN Diagrams as Models

The Business Process Modeling and Notation (BPMN) language [8, 14] is a conceptual business process specification language and is standardized by the Object Management Group (OMG). There are many different languages for description of business stories (e.g. SiteLang), of business rules (e.g. business use cases), and of workflows that are essentially specifications of business processes, activities of participants, utilisation with resources, and of communication among the participants. Languages such as S-BPM, BPMN, and EPC concentrate on different aspects of business processes, vary in scope and focus, use different abstraction levels, and are thus restricted in the capacity and potential for modelling. Most of the existing languages evolved over their lifespan and extensionally added features, more features, and other features again. BPMN is not an exception for this kind of overloading.

A business process consists of an ordered set of one or more activities (tasks) which collectively realize a business objective or policy goal. A workflow is the executable specification of a business process. It may describe all or some of the five aspects of business processes [15]:

(1) control flow description for the partial order of the activities, events or steps;
(2) organisation description with participants, theirs roles and plays within the processes, their rights and obligations, their resources, and their assignments;
(3) the data viewpoint description with an association to process elements and access rights for participants;
(4) the functional description that specifies semantics, pragmatics, and behaviour of each element of the workflow, e.g. the operations to be performed, pre- and postconditions, priority, triggers, and time frames for the operations;
(5) the operational assignment of programs that support all elements of the workflow.

The entire modelling process is based on a local-as-design perspective. A holistic or global view on a diagram collection is the task of a designer and becomes problematic in the case of specification evolution.

BPMN 2.0 defined four different kinds of diagrams for workflow specification. We shall briefly review these diagrams in the sequel. The diagram in Figure 4 combines these different aspects. It describes the accomplishment of requirements issued by a customer.

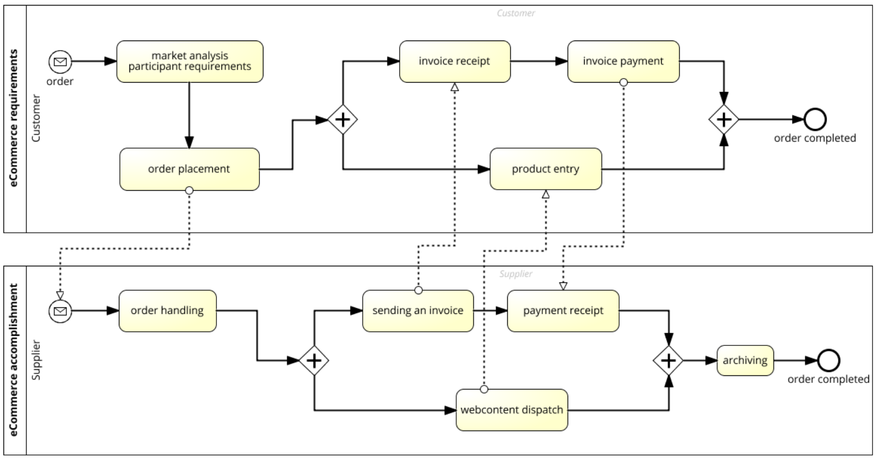


Fig. 4. Fulfillment of customer demands by vendors

3.1 Diagrams in BPMN

Process Diagrams. Process diagrams (also called orchestration diagrams) describe the stepwise task flow for one agent. A task flow might reflect different roles of an agent. These roles are separated by swimlanes. Processes are either public or private. Public processes can also be abstractions of private processes that represent the detailed control and task flow for a singleton agent. Main process elements are (a) atomic or complex activities for direct representation of stepwise actions of an agent, (b) gates for exclusive, non-exclusive, event-based or parallel splitting and joining of the control flow, (c) events for the start of a workflow, for the end of a workflow path, for the complete end of the entire workflow, and an intermediate event for representation of interaction events with agents outside the workflow, and (d) control flow arrows for representation of the order of process elements. Basic activities reflect abstract, service, send, receive, user, manual, business rule, or script tasks. Complex tasks reflect an entire sub-workflow, loops, parallel or sequential multiple executions, adhoc workflows, transactions, or specific exception handling workflows such as compensation. Interaction reflects message exchange, timer interaction, escalation enforcement, compensations, conditional interactions, links, and signals. Interaction can be sequential or parallel multiple. Interaction events may also be boundary events for a complex activity. All elements can be explicitly annotated by comments, by consumed data, or by produced data objects or data stores.

Choreography Diagrams. Choreography diagrams describe the message exchange among agents with reference to sending and receiving events, the message issue, and the graph-based representation of the partial order among these messages.

Collaboration Diagrams. Collaboration use choreography diagrams and process diagrams for explicit binding of senders and receivers of messages to black-box abstractions of agent workflows and abstract from message issues.

Conversation Diagrams. Conversation diagrams survey communication flow among agents as a birds view. They allow to derive dependences among process diagrams of agents.

3.2 Capability of BPMN Diagrams

BPMN modelling becomes nowadays a standard for typical business applications. Therefore, the capability of processes must be specified and well understood. It is thus necessary to know what is the ability to achieve a good model through a set of controllable and measurable features.

BPMN diagrams require a work-around for a number of conceptions such as macro-state, history, and system architecture. There are redundancies in the language itself that lead to flavour- or taste-oriented programming due to the overwhelming number of elements, construct excess and overload, e.g. groups, pool and lane, transformations, off-page connectors. The structuring becomes unclear since activities can be itself a workflow or a collection of workflows. This rather specific kind of abstraction should not be mixed with abstraction in general. Exception handling is completely confusing and only partially defined.

BPMN diagrams can represent only 8 out of 43 workflow resource pattern [10]. The data aspect is provided through properties of tasks, processes, and sub-processes. Their interrelationship is left to the developer community. It is the task of the developers to keep in mind the entire picture of the BPMN diagram collection.

BPMN uses an informal approach to semantics description what has been a matter of confusion. A formal approach to BPMN semantics can however be developed [1–3].

Furthermore, there is no conception of well-formed diagrams. Decomposition and composition is left to the developer. BPMN does not properly support the aspects (2), (4), and (5). The data aspect (3) is partially represented.

3.3 Deficiencies of Diagrams and Diagrammatical Reasoning

Diagrams are not universal for modelling. It is often claimed that diagrams are simple to use, are easy to interpret, have an intuitive semantics, are unique within a user community and have thus a unique pragmatics, and are thus powerful instruments. We observe however a number of obstacles that must be resolved before accepting a diagram as a model, e.g. the following ones:

- *Habituation versus unfamiliarity*: Diagram should be familiar to their users, have a unique semantics and pragmatics without any learning effort. Readers of diagrams must be literal with them.
- *Ambiguity of interpretation versus well-formedness*: Diagrams should not confuse by multiple interpretations (e.g. arrows), by instability and by context-dependence of form-content relations.
- *Incremental graphical construction*: Diagrams should follow the same construction pattern as the origin and should concentrate on typicality.
- *Naturalness of local reasoning*: Local-as-design approaches presuppose locality within the world of origins.
- *Unfamiliarity with non-linear behaviour*: Users are mainly linearly reasoning. Non-linear reasoning should be supported in a specific form.
- *Additional and supplementary elements without meaning*: Diagrams of use elements which do not have a unique or any meaning, e.g. colours, shapes, grid forms for lines etc.
- *Hidden dimensions within the diagram*: Diagrams cannot reflect all aspects although there are essential ones, e.g. time.
- *Representation as fine and visual art*: Finding a good representation is a difficult task and should be supported by a culture of modelling.

All these obstacles are observed in the case of BPMN diagrams [10].

Diagrams must be developed on the principles of visual communication, of visual cognition and of visual design [12]. The culture of diagramming is based on a clear and well-defined design, on visual features, on ordering, effect, and delivery, and on familiarity within a user community.

One of the main obstacles of diagrams is the missing abstraction. The simplest way to overcome it is the development of a model suite [19] consisting of a generic model and its refinement models where each of them is adequate and dependable. Generic models [31] reflect the best abstraction of all models within a model suite.

4 Evaluation of the BPMN Approach

BPMN is a powerful diagrammatic languages that uses more than 100 modelling elements. The same situation in the reality or the implemented system can be specified by a variety of diagrams. Since a theory of diagram equivalence is missing, [27] introduced seven evaluation methods for models:

- PURE – SMART – CLEAR evaluation for the goal-purpose-function evaluation of an instrument in a given application context, for given artifacts to be represented, for a given community of practice, and for a given profile (goal, purpose, and function) under consideration of the utilisation scenarios;
- PEST evaluation for assessment of internal, external, and quality of use;
- QUARZSAND evaluation for assessment of the model development, and
- SWOT – SCOPE evaluation for description of the potential of the model, i.e. the general properties of a given instrument or the modelling method.

Since we did not explore the directives in detail nor the adequateness and dependability of an instrument that is a candidate model, we concentrate on the last two methods in this section. The evaluation of adequacy and dependability has been developed in [28]. We concentrate here on the capacity and potential of the BPMN approach.

4.1 The Capacity of the BPMN Approach

Capacity is a strategic measure whereas the potential is a tactical one. The potential can be used to derive the added value of a utilisation of a model within a given scenario. The potential allows to reason on the significance of a model within a given context, within a given community of practice, for a given set of origins, and within the intended profile.

The capacity relates an instrument to utilisation scenarios or the usage spectra. We answer the questions whether the instrument functions well and beneficial in those scenarios, whether it is well-developed for the given goals and purposes, whether it can be properly, more focused, comfortably, simpler and intelligible applied in those scenarios instead of the origins, and whether the instrument can be adapted to changes in the utilisation. The answers to these questions determine the main content or cargo, the comprehensiveness, and the authority or general value of a model. Another important aspect is the solution-faithfulness of the instrument. The capacity is an essential element of the model cargo, especially of the main content of the model.

BPMN diagrams can be used in description, prescription, explanation, documentation, communication, negotiation, inspiration, exploration, definition, prognosis, reporting and other scenarios. We discover that communication, negotiation, and inspiration are supportable. Description, prescription, and definition can be supported if the BPMN diagrams are enhanced and a precise semantics of all BPMN elements is commonly used in all four kinds of diagrams. The adequacy and especially the analogy to the origins (i.e. storyboards or business processes) is assumed to be based on homomorphy what is rarely achieved. This homomorphy is suitable if all processes are completely and in detail specified and all variations and exceptions are consistent.

The general utility of BPMN diagrams becomes rather low if the specific background of the modelling approach is not taken into consideration. BPMN diagrams are process-oriented, based on an orthogonal separation of flow element into activities, gates, and events, differentiate actors within their roles, and support communication among actors based on message exchange. The execution semantics is based on a token interpretation of control flow. Actors are isolated in their execution if binding is not done through message exchange or implicit hidden resource conditions. Data and resource are however local. All processes are potentially executed in parallel. The local-as-design approach might be appropriate if business processes are not intertwined. The concentration on the same abstraction level restricts the applicability of BPMN modelling. Generic workflows [31] provide a solution to this limitation.

4.2 The Potential of the BPMN Approach

The potential describes the (in-)appropriateness of a modelling approach within the directives. The suitability of BPMN diagrams depends on whether the application and the context support the local-as-design approach, on whether the demands of the community of practice can be satisfied, on whether the instrument is adequate (analogous, focused, purposeful), on whether the goals can be achieved with the given instrument, on the fruitfulness of the instrument compared with other instruments, and on the threats and obstacles of utilisation of BPMN diagrams.

4.3 The SWOT Evaluation of the Potential

The SWOT analysis is a high-level method that allows to evaluate the general quality of an instrument and its general assumptions of deployment.

Strengths. The BPMN approach is standardised and uses a large variety of constructs. It thus allows development of detailed models. It has a high expressibility. Both intra- and inter-organisational aspects can be represented. The approach is well supported by tools.

Weaknesses. The large variety of competing elements is also a weakness. The complexity and integration of diagrams may cause solution-unfaithfulness. The language requires high learning efforts. Processes that are dynamic at runtime cannot be modelled. Exchange among tools is an open problem.

Opportunities. Most business processes can be adequately described due to the variety of elements. The standardisation provides at least a base semantics.

Threats and Risks. None-technical users might be unable to cope with diagrams. Work-arounds hinder comprehensibility. Vendors define their own extensions. The BPMN standard does not completely define the execution.

4.4 The SCOPE Evaluation of the Potential

The SCOPE analysis of a model embeds the model into the application context, refines the capacity evaluation of an instrument, and considers the community of practice and their specific needs.

Situation. BPMN diagram suites provide some kind of formalisation of business processes. Communication is specified to a certain degree. Control flow is well-represented.

Competence. BPMN diagrams must be combined with other models since the other four aspects (organisation, data flow, functions, operational assignment) are only partially reflected.

Obstacles. Typical challenges of BPMN modelling are the specification complexity, diagram coherence, exception handling, and the development of an execution semantics. There is no common agreement on well-formedness of diagrams.

Prospects. A separate BPMN diagram is easy to read and to interpret.

Expectations. The BPMN approach can be combined with local-as-design-oriented conceptual data models, storyboards, business rule specification and other modelling approaches as one kind of models within a model suite.

4.5 The Resulting Potential of the BPMN Approach

The BPMN diagram has a high potential for communication and negotiation utilisation scenario. The potential for system construction within a description-prescription scenario is however rather limited due to missing co-design support. A similar inappropriateness can be stated for explanation, prognosis, exploration, definition, and reporting scenario. The potential within a documentation scenario is rather small. The highest potential of the BPMN approach can be however observed for inspiration scenario. The process, choreography, conversation, and collaboration diagrams are an appropriate means for an implementation plan based on inspiring diagrams.

Similar to SPICE assessments [7], we may rate maturation of a model and a modelling approach to: (0) ad-hoc , (1) informal, (2) systematic and managed, (3) standardised and well-understood, and (4) optimising and adaptable, and (5) continuously improvable styles. The evaluation shows that the BPMN approach has not yet reached level (2). This observation leads us to the conclusion that PURE-SMART-CLEAR and PEST evaluations are heavily dependent from the directives for BPMN diagram modelling.

A model must be of high utility, must have a high added value, and should have a high potential. These parameters also depend on the well-formedness of the instrument. The BPMN approach can be enhanced by criteria for well-formedness for syntactical, semantical and pragmatical well-shaped diagrams [28].

5 Towards a Theory of Modelling

5.1 Models Burdened with Directives and Background

The directives and the background (see Figure 2) heavily influence the way how a model is constructed, what is taken into consideration and what not, which rigidity is applied, which basis and grounding is taken for granted, and which community of practice accepts this kind of model.

The model incorporates these influences without marking them in an explicit form. The model is laden or *burdened* by these decisions. Additionally, models are composed of elements that are selected, changed and adapted within a development process. Figure 5 depicts elements of this burden and this development history.

5.2 The Anti-Profile of an Instrument as a Model

We may now directly conclude that an instrument might or might not be adequate and dependable for any utilisation scenario due to its insufficiency to function in this scenario.

Definition 10. *A* utilisation pattern *of an instrument describes the form of usage of an instrument, the discipline of usage, the applications in which the instrument might be used, and the conditions for its utilisation.*

Definition 11. *A* utilisation scenario *consists of a utilisation pattern and a number of functions a specific instrument might play in this utilisation pattern.*

Definition 12. *A* usage spectrum *consists of collection of utilisation scenarios. A* portfolio *of an instrument combines the usage spectra.*

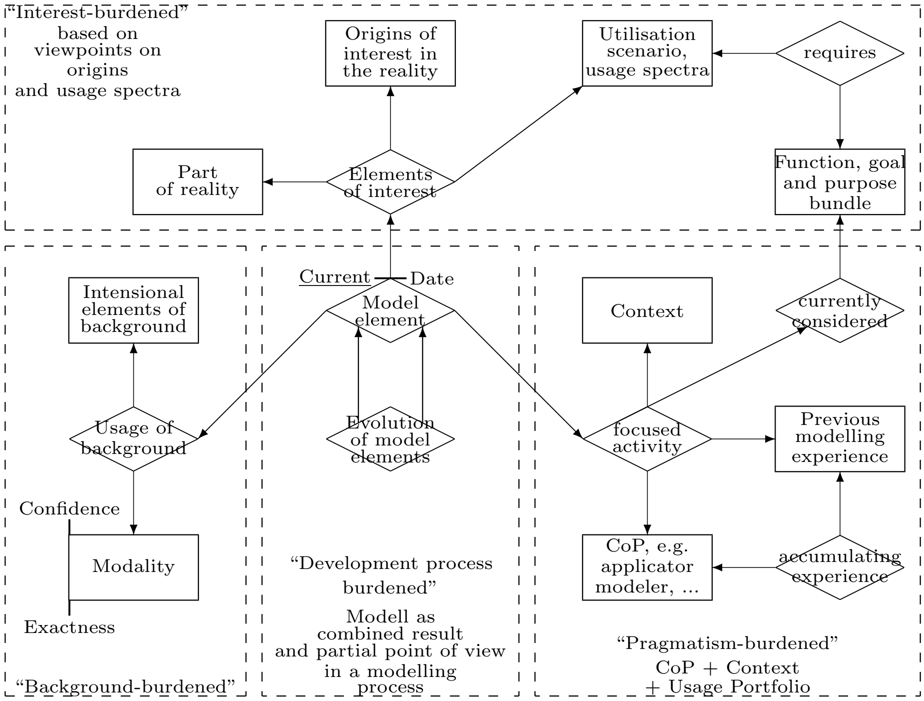


Fig. 5. Models are burdened by their development history, the background and the directives

Definition 13. *A* profile *of an instrument as a model consists of the goals, the purposes, and the functions of the instrument within a portfolio.*

We can now roughly describe an anti-profile of a model and resulting *utilisation proscriptions* of a an instrument as a model by answering the following questions:

- For which scenarios is the instrument useless?

- In which of the following scenarios efficiency and effectiveness is not given for the instrument: description & prescription, realisation & coding, theory development, theory refinement, causality consideration, inexplicability, demonstration, prediction, explanation, mastering of complexity, understanding, or ... ?
- Are essential parameters of the origins missing? Are some of the essential parameters only represented via mediating or dependable parameters? Are there dummy or pseudo dependences among the parameters?
- What cannot be adequately represented? Is the dependability really sufficient? In which case users need a special understanding and education? Which tacit knowledge is hidden in the instrument?
- In which cases the instrument cannot be effectively used? What must a user respect and obviate before using the instrument?
- Which biases and which background are palmed off? Which assumptions, postulates, paradigms, and schools of thought are hidden and not made explicit? Models might condition conclusions.

Since models are instruments their utilisation conditions conclusions and results. Therefore, it is appropriate to describe the anti-profile of a model as well.

5.3 Questions to Answer Before Using an Instrument as a Model

The rhetoric frame and its extension to the W*H frame [5] can now be used for derivation of questions one must answer before using the model:

- What is the function of the model in which scenario? What are consequential purposes and goals? What are anti-goals and anti-purposes?
- Which origins are going to be represented? Which are not considered? Does the model contain all typical, relevant and important features of the origins under consideration and only those?
- Is the instrument adequate and dependable within the utilisation scenario? What are the parameters for adequacy and dependability? How purpose-invariance and solution-faithfulness is going to be defined?
- What kind of reasoning is supported? What not? What are the limitations? Which pitfalls should be avoided?
- Do you want to have a universal model that contains all and anything? Would it be better to use a model suite where each of the models represent some specific aspects? What about the nonessential aspects?

6 Conclusion

A general understanding of the notion of a model has been already started with development of Computer Science. Milestones are the papers and books by H. Stachowiak (1980ies and 1990ies), B. Mahr (2000ies until 2015), W. Steinmüller (1993), and R. Kaschek (since 1990ies) [9, 11, 17, 18, 21]. These notions treat models from a phenomenological point of view through properties that a model should have (e.g. as main properties: mapping or analogy, truncation, pragmatic properties). We need however also an explicit definition of the notion of model. Such general notion has been developed in a series of papers, e.g. [22, 24, 25, 27, 29].

The model notion is universal one and based on two parameter sets for adequateness and dependability. The parameter sets seem to be complex and need a methodological support. This paper develops such a support facility based on the notion of a conception frame. The practicality of the approach is demonstrated for the workflow specification language BPMN. BPMN shares the positive treatment with most other formal or informal languages in Computer Science. The capacity and thus the restrictions or obstacles are not explicitly communicated. We see however that the evaluation, capacity, potential, and capability can be explicitly provided based on our approach.

Since the model notion is a mathematical definition, it seems to be achievable to develop a theory of modelling in the sense of a theory. In this paper, we only discuss two components of such a theory: the explicit description of the background of models and the anti-profile. The conception frame for the model definition may also be used for derivation of question forms that a modeller can use before delivering an instrument as a model to a community of practice. The development of a full theory is however a research issue for the next decades.

References

1. E. Börger, O. Sörensen, and B. Thalheim. On defining the behavior of or-joins in business process models. *Journal of Universal Computer Science*, 15(1):3–32, 2009.
2. E. Börger and B. Thalheim. A method for verifiable and validatable business process modeling. In *Software Engineering*, LNCS 5316, pages 59 – 115. Springer, 2008.
3. E. Börger and B. Thalheim. Modeling workflows, interaction patterns, web services and business processes: The ASM-based approach. In *ABZ*, volume 5238 of *Lecture Notes in Computer Science*, pages 24–38. Springer, 2008.
4. S. Chadarevian and N. Hopwood, editors. *Models - The third dimension of science*. Stanford University Press, Stanford, California, 2004.
5. A. Dahanayake and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter W*H: The conceptual Model for services, pages 145–176. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.
6. L. Fleck. *Denkstile und Tatsachen, edited by S. Werner and C. Zittel*. Surkamp, 2011.
7. ISO/IEC. Information technology - process assesment. parts 1-5. IS 15504, 2003-2006.
8. ISO/IEC. International organization for standardization: Information technology object management group: Business process model and notation. http://www.omg.org/spec/BPMN/ISO/19510/PDF/, 2013.
9. R. Kaschek. *Konzeptionelle Modellierung*. PhD thesis, University Klagenfurt, 2003. Habilitationsschrift.
10. F. Kossak, C. Illibauer, V. Geist, J. Kubovy, C. Natschläger, T. Ziebermayr, T. Kopetzky, B. Freudenthaler, and K.-D. Schewe. *A Rigorous Semantics for BPMN 2.0 Process Diagrams*. Springer, 2014.
11. B. Mahr. Information science and the logic of models. *Software and System Modeling*, 8(3):365–383, 2009.

12. T. Moritz. *Visuelle Gestaltungsraster interaktiver Informationssysteme als integrativer Bestandteil des immersiven Bildraumes.* PhD thesis, HFF Berlin-Babelsberg, 2006.
13. I. Nissen and B. Thalheim. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Bedeutung, Entwicklung und Einsatz, pages 3–28. De Gryuter, Boston, 2015.
14. OMG. Object management group: Business process model and notation (BPMN) 2.0. http://www.omg.org/spec/BPMN/2.0, 2011.
15. H. Pichler and J. Eder. Business process modeling and workflow design. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 8, pages 259–286. Springer, Berlin, 2011.
16. J.E. Safra, I. Yeshua, and et. al. *Encyclopædia Britannica.* Merriam-Webster, 2003.
17. H. Stachowiak. Modell. In Helmut Seiffert and Gerard Radnitzky, editors, *Handlexikon zur Wissenschaftstheorie*, pages 219–222. Deutscher Taschenbuch Verlag GmbH & Co. KG, München, 1992.
18. W. Steinmüller. *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik.* Wissenschaftliche Buchgesellschaft, Darmstadt, 1993.
19. B. Thalheim. *The Conceptual Framework to Multi-Layered Database Modelling based on Model Suites*, volume 206 of *Frontiers in Artificial Intelligence and Applications*, pages 116–134. IOS Press, 2010.
20. B. Thalheim. 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.`
21. B. Thalheim. 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*, chapter 17, pages 547–580. Springer, Berlin, 2011.
22. B. Thalheim. The science and art of conceptual modelling. In A. Hameurlain et al., editor, *TLDKS VI*, LNCS 7600, pages 76–105. Springer, Heidelberg, 2012.
23. B. Thalheim. The conception of the model. In *BIS*, volume 157 of *Lecture Notes in Business Information Processing*, pages 113–124. Springer, 2013.
24. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
25. B. Thalheim and A. Dahanayake. A conceptual model for services. In *Invited Keynote, CMS 2015, ER 2015 workshop*, LNCS 9382, pages 51–61, Berlin, 2015. Springer.
26. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung.* De Gruyter, Ontos Verlag, Berlin, 2015.
27. B. Thalheim and I. Nissen. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Ein neuer Modellbegriff, pages 491–548. De Gryuter, Boston, 2015.
28. B. Thalheim and I. Nissen. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Fallstudien zum Modellbegriff, pages 549–602. De Gryuter, Boston, 2015.
29. B. Thalheim and M. Tropmann-Frick. The conception of the conceptual database model. In *ER 2015*, LNCS 9381, pages 603–611, Berlin, 2015. Springer.
30. M. Thomas. Modelle in der Fachsprache der Informatik. Untersuchung von Vorlesungsskripten aus der Kerninformatik. In *DDI*, volume 22 of *LNI*, pages 99–108. GI, 2002.

31. M. Tropmann-Frick, B. Thalheim, D. Leber, G. Czech, and C. Liehr. Generic workflows - a utility to govern disastrous situations. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 417–428. IOS Press, 2014.

Enhancing Entity-Relationship Schemata for Conceptual Database Structure Models

Bernhard Thalheim and Marina Tropmann-Frick

Department of Computer Science, Christian-Albrechts-University Kiel, 24098 Kiel, Germany

Abstract. The paper aims at development of well-founded notions of database structure models that are specified in entity-relationship modelling languages. These notions reflect the functions a model fulfills in utilisation scenarios.

1 Utilisation Scenarios of Conceptual Models

Conceptual models are used as an artifact in many utilisation scenarios. Design science research [4] and ER schema development methodologies (e.g. [3, 8, 11]) developed so far a good number of such scenarios.

Communication and negotiation scenario: The conceptual model is used for exchange of meanings through a common understanding of notations, signs and symbols within an application area. It can also be used in a back-and-forth process in which interested parties with different interests find a way to reconcile or compromise to come up with an agreement. The schema provides negotiable and debatable propositions about the understanding of the part of the reality but does not have well-developed justificatory explanations.

Conceptualisation scenario: The main application area for extended entity-relationship models is the conceptualisation of database applications. Conceptualisation is typically shuffled with discovery of phenomena of interest, analysis of main constructs and focus on relevant aspects within the application area. The specification incorporates concepts injected from the application domain.

Description scenario: In a description scenario, the model provides a specification how the part of the reality that is of interest is perceived and in which way augmentations of current reality are targeted. The model says what the structure of an envisioned database is and what it will be.

Prescription scenario: The conceptual model is used as a blueprint for or prescription of a database application, especially for prescribing the structures and constraints in such applications. The schema proposes what the structure of a database is on the one hand and how and where to construct the realisation on the other hand. ER schemata can be translated to relational, XML or other schemata based on transformation profiles [11] that incorporate properties of the target systems.

LNCS 9381, 603--611, Berlin, 2015. Springer

These scenarios are typically bundled into *use spectra*. For instance, design science uses three cycles: the relevance cycle based on the design cycle based on description and communication and negotiation scenario, and the rigor cycle based on a knowledge discovery and experience propagation scenario. Database development is mainly based on description, conceptualisation, and construction scenarios. The re-engineering and system maintenance use spectrum is based on combination of documentation scenarios with an explanation and discovery scenario from one side and communication and negotiation scenario from the other side. Models are also used for documentation scenarios, explanation and discovery scenarios for applications or systems, and for knowledge experience scenario. We concentrate here on the four scenarios.

Contribution of the Paper

The first main contribution of this paper is an analysis whether the an entity-relationship schema is suffices as a model for database structures. We realise that the four scenarios require additional elements for the ER schema in order to become a model. The second main contribution of this paper is a proposal for an enhancement of ER schemata which allows to consider the artifact as a model within the given four scenarios. The paper partially presupposes our research (esp. [14], see also other papers in [15]).

2 The General Notion of a Model

Science and technology widely uses models in a variety of in utilisation scenarios. Models function as an artifact in some utilization scenario. Their function in these scenarios is a combination of functions such as explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualization, and description-prescription as a mediator between a reality and an abstract reality that developers of a system intend to build. The model functions determine the *purposes* of the deployment of the model.

The following notion of the model has been developed [16] after an intensive discussion in workshops with researchers from disciplines such as Archeology, Arts, Biology, Chemistry, Computer Science, Economics, Electrotechnics, Environmental Sciences, Farming and Agriculture, Geosciences, Historical Sciences, Humanities, Languages and Semiotics, Mathematics, Medicine, Ocean Sciences, Pedagogical Science, Philosophy, Physics, Political Sciences, Sociology, and Sport Science.

Definition 1. *A model is a well-formed, adequate, and dependable artifact that represents origins. Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios. As an artifact, a model is grounded in its community's sub-discipline and is based on elements chosen from the sub-discipline.*

This notion has been tested against the notions of a model that are typically used in these disciplines. We could state that these notions are covered by our notion. Origins of a model [7] are artifacts the model reflects. Adequacy of models has often been discussed, e.g. [6, 9, 10]. Dependability is only partially covered in research, e.g. [5].

Models have several *essential properties* that qualify an artifact as a model [14, 15]:

- An artifact is *well-formed* if it satisfies a well-formedness criterion.
- A well-formed artifact is *adequate* for a collection of origins if (i) it is analogous to the origins to be represented according to some analogy criterion, (ii) it is more focused (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and (iii) it sufficient satisfies its purpose.
- Well-formedness enables an artifact to be *justified*: (i) by an empirical corroboration according to its objectives, supported by some argument calculus, (ii) by rational coherence and conformity explicitly stated through formulas, (iii) by falsifiability that can be given by an abductive or inductive logic, and (iv) by stability and plasticity explicitly given through formulas.
- The artifact is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics [13] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).
- A well-formed artifact is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.
- An artifact is called **model** if it is *adequate* and *dependable*. The adequacy and dependability of an artifact is based on a *judgement* made by the community of practice.
- An artifact has a *background* consisting of an undisputable grounding from one side (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities) and of a disputable and adjustable basis from other side (assumptions, concepts, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense).
- A model is used in a *context* such as discipline, a time, an infrastructure, and an application.

The Taxonomy of Conceptual Models

The starting point in our investigation was the observation that there is no unique and commonly agreeable notion of the conceptual database structure model as such. The model supports different purposes and has different functions in utilisation scenarios. Therefore, we must have different notion of the conceptual database structure model.

Due to space limitations we concentrate on the first four utilisation scenarios. The other four scenarios are supported by specific conceptual models in a similar form.

3 Conceptual Database Structure Models for Communication and Negotiation

Communication aims at exchange of meanings among interested parties. The model is used as a means for communication. It truly represents some aspects of the real world. It enables clearer communication and negotiation about those aspects of the real world. It has therefore potentially several meanings in dependence on the parties. Communication acts essentially follow rhetoric frames[1], i.e. they are characterised through "who says what, when, where, why, in what way, by what means" (Quis, quid, quando, ubi, cur, quem ad modum, quibus adminiculis). In our case, the model ("what") incorporates the meaning of parties (semantical space; "who") during a discourse ('when') within some application with some purpose ("why") based on some modelling language.

Typically, artifacts used for communication and negotiation follow additional principles: Viewpoints and specific semantics of users are explicitly given. The artifact is completely logically independent from the platform for realisation. The name space is rather flexible. The model is functioning and effective if methods for reasoning, understanding, presentation, exploration, explanation, validation, appraisal and experimenting are attached.

Conceptual model for communication: *The conceptual database structure model comprises the database schema, reflects viewpoints and perspectives of different involved parties $\mathcal{U}$ and their perception models, and implicitly links to (namespaces or) concept fields of parties. Adequacy and dependability are based on the association of the perception models to viewpoints and of the viewpoints with the schema..*

A partial communication model does not use a schema and does not associate viewpoints to schema elements.

Therefore, the model can be formally defined as a quintuple

$$(\mathcal{S},\ \{\,(\mathcal{V}_i, \Phi_i) \,|\, i \in \mathcal{U}\},\ \{\,(\mathcal{P}_i, \Psi_i) \,|\, i \in \mathcal{U}\},\ \mathcal{A},\ \mathcal{D})$$

that relates elements of the conceptual schema $\mathcal{S}$ to the perception model $\mathcal{P}_i$ of the given party i. The perception model is reflected in the schema via viewpoints $\mathcal{V}_i$. It implicitly uses concept fields $\mathfrak{C}_i$ of parties i. The mapping $\Psi_i : \mathcal{P}_i \to \mathcal{V}_i$ associates the perception model of a given party i to the agreed viewpoint. In the global-as-design approach, the viewpoint $\mathcal{V}_i$ is definable by some constructor Φ_i defined on $\mathcal{S}$. The adequacy $\mathcal{A}$ is directly given by the second and third parts of the model. The justification $\mathcal{J}$ and the dependability $\mathcal{D}$ are extracted from the properties of Φ_i and Ψ_i.

The negotiation scenario can thus be understood as stepwise construction of the mappings, stepwise revision of the schema and the viewpoints, and analysis whether the schema represents the corresponding perception model.

---

[1] It relates back to Hermagoras of Temnos or Cicero more than 2000 years ago.

4 Conceptual Database Structure Models for Conceptualisation

Conceptualisation is based on one or more concept or conception spaces of business users. Given a business user community $\mathcal{U}$ with their specific concept fields $\{\mathfrak{C}_i | \, i \in \mathcal{U}\}$. Let us assume that the concept fields can be harmonised or at least partially integrated into a common concept field of users $\mathfrak{C}^{\mathcal{U}}$ similar to construction approaches used for ontologies.

Conceptual model for conceptualisation: *The conceptual database structure model comprises the database schema, a collection of views for support of business users, and a mapping for schema elements that associates these elements to the common concept field..*

Therefore, the model can be formally defined as a quintuple

$$(\mathcal{S}, \mathfrak{V}, \mathcal{M}, \mathcal{A}, \mathcal{D})$$

consisting of the conceptual schema $\mathcal{S}$ and a mapping $\mathcal{M} : \mathcal{S} \rightarrow \mathfrak{C}^{\mathcal{U}}$. The adequacy $\mathcal{A}$ is based on the mapping. The justification $\mathcal{J}$ and the dependability $\mathcal{D}$ are derived from the concept fields.

5 Conceptual Database Structure Models for Description and Prescription

An artifact that is used as a conceptual model for database system description can be either understood as a representation, refinement and amplification [13, 15] of situation or reality models or as a refinement and extension of the communication model. The main approach to conceptual modelling for system construction follows the first option. The second option would however be more effective but requires a harmonisation of the perception models. The first option may start with reality models that reflect the nature of the business in terms and in the language of the business. It includes also the top management view, a corporate overview, and a sketch of the environment. The reality models are reasonable complete, are described in terms of the business and use general categories that are convergent.

Conceptual model for description: *The conceptual database structure model comprises the database schema, a collection of views for support of business users, a collection of a commonly accepted reality models that reflects perception or situation models with explicit association to views, and the declaration of model adequacy and dependability..*

Therefore, the model can be formally defined as a quintuple

$$(\mathcal{S}, \mathfrak{V}, (\mathfrak{R}, \Psi), \mathcal{A}, \mathcal{D}) \ .$$

The conceptual model may be enhanced by an association Φ of views to the schema. This enhancement is however optional.

Descriptive models adequately explicate main concepts [12] from the reality models and combine them into views. The descriptive model reflects the origins and abstracts from reality by scoping the model to the ideal state of affairs.

Prescriptive models that are used for system construction are filled with anticipation of the envisioned system. They deliberately diverge from reality in order to simplify salient properties of interest, transforming them into artifacts that are easier to work with. They may follow also additional paradigms and assumption beyond the classical background of conceptual database structure models: Salami slicing of the schema by rigid separation of concern for all types; conformance to methods for simple (homomorphic) transformation; adequateness for direct incorporation; hierarchical architecture within the schema, e.g. for specialisation and generalisation of types; partial separation of syntax and semantics; tools with well-defined semantics; viewpoint derivation; componentisation and modularisation; integrity constraint formulation support; conformance to methods for integration; variations for the same schema for more flexible realisation etc.

Directives (or pragmas) [1] prescriptively specify properties for the realisation. Transformation parameters [11] for database realisation are, for instance, treatment of hierarchies, controlled redundancy, NULL marker support, constraint treatment, naming conventions, abbreviation rules, set or pointer semantics, handling of weak types, and translation options for complex attributes. Based on [2] we give an explicit specification of directives for the realisation. The prescription model also consists of a general description of a realisation style and tactics, of configuration parameters (coding, services, policies, handlers), of generic operations, of hints for realisation of the database, of performance expectations, of constraint enforcement policies, and of support features for the system realisation. These parameters are combined to the realisation template $\mathcal{T}$. The realisation template can be extended by quantity matrix for database classes $\mathfrak{Q}$ and other performance constraints $\mathfrak{C}$ and by business tasks and their reflections through business data units $\mathfrak{B}$. Directives can be bound to one kind of platform and represent in this case a technological twist, e.g. by stating how data is layered out. They are typically however bound to several platforms in order to avoid evolution-proneness of models.

Conceptual model for prescription: *The conceptual database structure model comprises the database schema, a collection of views for both support of business users and system operating, a realisation template, and the declaration of model adequacy and dependability..*

Therefore, the model can be formally defined as a quintuple
$$(\mathcal{S}, \mathfrak{V}, \mathcal{T}, \mathcal{A}, \mathcal{D}) .$$

6 Conclusion

This paper shows that the ER schema is a central unit in a conceptual database structure model. The conceptual database structure model contains also other elements in dependence on its function in utilisation scenarios. As long as we use a global-as-design approach, the ER schema is essential and the kernel of such database structure models.

We may combine the conceptual models to description/prescription models

$$(\mathcal{S},\, \mathfrak{V},\, (\mathfrak{R}, \varPsi),\, \mathcal{T},\, \mathcal{A},\, \mathcal{D})\ .$$

and to description/prescription models with conceptualisation

$$(\mathcal{S},\, \mathfrak{V},\, (\mathfrak{R}, \varPsi),\, \mathcal{M},\, \mathcal{T},\, \mathcal{A},\, \mathcal{D})\ .$$

The combination with communication/negotiation is more problematic since the corresponding models are based on divergent perception models that might represent the very personal viewpoint of business users in different context and work organisation.

The notion of the model for conceptual database structure models can be summarised in dependence on their utilisation scenario:

Table 1. Conceptual database structure models that extend the conceptual database schema in dependence on utilisation scenarios

Scenario	Model origin	Add-ons to the conceptual database schema
Communication and negotiation	Perception (and situation) models	Views representing the viewpoint variety and associated with the perception models
Conceptualisation	Perception and reality models	Associations to concepts and conceptions, semantics and meanings, namespaces
Description	Reality model	View collection, associations to origins
Prescription	Reality (and situation) models	View collection, realisation template

References

1. ISO/IEC JTC 1/SC 22. Information technology – Programming languages – C. ISO/IEC 9899:2011, Stage: 60.60, 2011.
2. B. AlBdaiwi, R. Noack, and B. Thalheim. Pattern-based conceptual data modelling. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 1–20. IOS Press, 2014.
3. C. Batini, S. Ceri, and S. Navathe. *Conceptual database design (an entity-relationship approach).* Benjamin/Cummings, Redwood City, 1992.
4. S. Gregor. The nature of theory in information systems. *MIS Quarterly*, 30(3):611–642, 2006.
5. I.A. Halloun. *Modeling Theory in Science Education.* Springer, Berlin, 2006.
6. R. Kaschek. *Konzeptionelle Modellierung.* PhD thesis, University Klagenfurt, 2003. Habilitationsschrift.
7. B. Mahr. Information science and the logic of models. *Software and System Modeling*, 8(3):365–383, 2009.
8. H. Mannila and K.-J. Räihä. *The design of relational databases.* Addison-Wesley, Wokingham, England, 1992.
9. H. Stachowiak. Modell. In Helmut Seiffert and Gerard Radnitzky, editors, *Handlexikon zur Wissenschaftstheorie*, pages 219–222. Deutscher Taschenbuch Verlag GmbH & Co. KG, München, 1992.

10. W. Steinmüller. *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1993.
11. B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
12. B. Thalheim. The conceptual framework to user-oriented content management. In *Information Modelling and Knowledge Bases*, volume XVIII of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2007.
13. B. Thalheim. 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`.
14. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
15. B. Thalheim. Models, to model, and modelling - Towards a theory of conceptual models and modelling - Towards a notion of the model. Collection of recent papers, http://www.is.informatik.uni-kiel.de/∼thalheim/indexkollektionen.htm, 2014.
16. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung*. De Gruyter, Ontos Verlag, Berlin, 2015.

Bus Inf Syst Eng
DOI 10.1007/s12599-016-0439-z

DISCUSSION

Theories in Business and Information Systems Engineering

**Martin Bichler · Ulrich Frank · David Avison · Julien Malaurent ·
Peter Fettke · Dirk Hovorka · Jan Krämer · Daniel Schnurr ·
Benjamin Müller · Leena Suhl · Bernhard Thalheim**

© Springer Fachmedien Wiesbaden 2016

1 Theories in Business and Information Systems Engineering

1.1 Introduction

Even though the idea of science enjoys an impressive reputation, there seems to be no precise conception of science. On the one hand, there is no unified definition of the extension of activities subsumed under the notion of science. According to the narrow conception that is common in Anglo-Saxon countries, science is restricted to those disciplines that investigate nature and aim at explanation and prediction of natural phenomena. A wider conception that can be found in various European countries includes social sciences, the humanities and engineering. On the other hand and related to the first aspect, there is still no general consensus on the specific characteristics of scientific discoveries and scientific knowledge.

1.2 Theory and Science

The demarcation problem in the philosophy of science is how to distinguish between science and non-science. Some argue that the demarcation between science and non-science is a pseudo-problem that would best be replaced by focusing on the distinction between reliable and unreliable

Prof. Dr. M. Bichler (✉)
Department of Informatics, Technical University of Munich,
Boltzmannstr. 3, 85748 Garching/Munich, Germany
e-mail: bichler@in.tum.de

Prof. Dr. U. Frank
Chair of Information Systems and Enterprise Modelling,
Institute for Computer Science and Business Information
Systems, University of Duisburg-Essen, Universitaetsstr. 9,
45141 Essen, Germany
e-mail: ulrich.frank@uni-duisburg-essen.de

Dr. D. Avison · Dr. J. Malaurent
Department of Information Systems, Decision Sciences and
Statistics, ESSEC Business School, Avenue Bernard Hirsch,
B.P. 50105, 95021 Cergy Pontoise Cedex, France
e-mail: avison@essec.fr

Dr. J. Malaurent
e-mail: malaurent@essec.edu

Prof. Dr. P. Fettke
German Research Center for Artificial Intelligence (DFKI
GmbH) and Saarland University, Campus D3 2,
Stuhlsatzenhausweg 3, 66123, Saarbrücken, Germany
e-mail: peter.fettke@iwi.dfki.de

Dr. D. Hovorka
Business Information Systems, The University of Sydney
Business School, The University of Sydney, Sydney, NSW 2006,
Australia
e-mail: d.hovorka@econ.usyd.edu.au

Prof. Dr. J. Krämer · D. Schnurr
Chair of Internet and Telecommunications Business,
Faculty of Business Administration and Economics,
University of Passau, Dr.-Hans-Kapfinger-Str. 12,
94032 Passau, Germany
e-mail: jan.kraemer@uni-passau.de

D. Schnurr
e-mail: daniel.schnurr@uni-passau.de

Dr. B. Müller
Innovation Management and Strategy, Faculty of Economics and
Business, University of Groningen, Nettelbosje 2,
9747 AE Groningen, The Netherlands
e-mail: b.mueller@rug.nl

Published online: 22 June 2016

knowledge, without bothering to ask whether that knowledge is scientific or not. Nevertheless, there seems to be one answer to Kant's question concerning the difference between scientific insights and the dreams of a ghost-viewer that is accepted by many: At its core, scientific knowledge is based on theories. Therefore, research should be aimed at the construction and testing of theories. However, this conclusion is satisfactory only at first sight, because the concept of theory itself lacks a unified and commonly accepted definition. There seem to be various reasons for this surprising lack of conceptual clarity at the foundation of an enterprise that is aimed at linguistic precision.

First, the term "theory" is used for different kinds of epistemological constructions. That makes it difficult to develop a satisfactory general conception. Philosophy of science does not provide us with an accepted concept of theory either (Godfrey-Smith 2003). Formal theories developed using the axiomatic method as it is subject of mathematics and logic are not necessarily motivated by observations from the empirical world. Their truth can be proved, i.e., they can be verified with respect to the underlying axioms. Theories in the empirical sciences usually aim at gaining reliable descriptions of reality. Therefore, their justification will depend on some form of confrontation with a conception of reality which is coined by underlying epistemological and ontological assumptions. In the case of (neo)positivist approaches, this kind of justification is based on the correspondence theory of truth, which in turn has its background in a (critical) realist view of the world. Some philosophers of science aim at a (partially) formalized conception of empirical theories. The semantic view (Suppe 1989) regards theories as being comprised of sets of mathematical models and sets of models with an empirical claim. (Testable) hypotheses then serve to link both kinds of models. The 'non-statement view' of theories aims at specifying a formal structure, also called an "architectonic", which should be suited to represent the "'essential' features of empirical knowledge ..." (Balzer et al. 1987, xvii). The formal structure comprises a set of so called potential models (interpretations) of the underlying conceptual framework. Hermeneutic approaches which are rather based on different forms of constructivism or idealism make use of the coherence or the consensus theory of truth. In addition to that it is questionable whether truth is always the only justification criterion (Frank 2006).

Second, the actual use of the term is not only ambiguous but also ambivalent. A clear distinction between scientific (theoretical) and non-scientific knowledge is not trivial, if not impossible (Laudan 1983). Furthermore, studies in sociology of science show that scientific knowledge contributions are not independent from external factors such as incentives, expected reputation or power games (Feyerabend 1993; Kuhn 1964; Latour and Woolgar 1986). Sometimes it may seem that a theory is the result of a social construction – somebody has named it as such and his proposal was legitimized by being published in a top tier journal – rather than an epistemological distinction.

1.3 Theories in Our Field

The lack of a satisfactory conception of theory is especially critical in Information Systems or Business and Information Systems Engineering (BISE), respective. The wide range of research topics in our field comprises not only empirical theories, but also formal theories and the design of elaborate artifacts. At the same time, leading journals emphasize the need for theories, thereby creating a situation that is suited to create confusion. Various publications are aimed at targeting this problem.

Especially Gregor (2006) helps clarifying the use of theories in Information Systems. However, her work is mainly restricted to (neo-)positivist ideas of theory (Popper, Hempel/Oppenheim) and does not account for the peculiarities of formal theories or those conceptions of theory found in our neighboring disciplines economics, informatics, and management science, and also of those in several sub-communities of BISE. Frank (2006) suggests a meta conception of scientific knowledge that covers empirical, formal and design contributions, but does not provide a correspondingly wide conception of theory.

The situation is even worse when it comes to criteria that help assessing the quality of theories – especially with respect to the epistemological value of probabilistic propositions that are used by the majority of theoretical contributions in our field (Lim et al. 2009) – and that Popper refused to accept as proper theories. The problems caused by an ambiguous conception of theory in our field have been known for some time. In a recent debate that was triggered by Avison and Malaurant (2014) who question what they call the "theory fetish in information systems", (Markus 2014, p. 342) comes to the conclusion "... that conflicting notions of theory and theoretical contribution, rather than sheer overemphasis on theory, may lie at the heart of the problem that Avison and Malaurent identified."

Prof. Dr. L. Suhl
Decision Support and Operations Research Lab, Department Business Information Systems, Universität Paderborn,
Warburger Str. 100, 33098 Paderborn, Germany
e-mail: suhl@upb.de

Prof. Dr. B. Thalheim
Information Systems Engineering (ISE), Institut für Informatik,
Christian-Albrechts-Universität zu Kiel, Olshausenstr. 40,
24098 Kiel, Germany
e-mail: thalheim@is.informatik.uni-kiel.de

A close look at theories relevant for our field results in a wide range of examples that are substantially different. For example, in informatics theoretical foundations such as automata theory, computability theory, complexity theory, or computational learning theory, which are typically based on the axiomatic method, constitute foundations for engineering sub-disciplines such as data engineering, data mining, and operations research. In those fields that focus on human behavior and action systems, many researchers follow a neo-positivist research paradigm with a concept of theory that leans on that common in the natural sciences. However, some researchers in these fields prefer hermeneutic approaches, e.g., for conducting case studies. Respective research methods do not only replace the idea of scientific objectivity with subjectivity, they sometimes deny the need for generalization.

The neo-positivist conception is challenged by a further principal concern that is directly related to a current subject of our research: the digital transformation. It is questionable whether research can provide an orientation for change if it is focused on actual or past patterns of developing and using IT. Instead, it may be more appropriate to emphasize the notion of theory ("the$\underline{o}$ría"): to transcend the "factual" world by contemplation. For us that means to look beyond current patterns of developing and using IT or – in other words: to develop justified (!) models of possible future worlds (Rorty 1999; Frank (2006) that serve those who live the future as an inspiration and a meaningful orientation. Respective constructions cannot be validated by confronting them with reality, since they are on purpose different from it.

Fields that make heavy use of formal models and methods are arguably very important for our discipline. They emphasize the power of mathematics and logic for representing scientific knowledge. While respective constructions come with obvious advantages as they allow for computing and proving, they come with the problem how to decide whether there is a valid empirical interpretation of socio-economic systems and whether actors can be expected to follow the rules of logic.

On the other hand, there are researchers that follow a more empiricist agenda, but aim to reconstruct their theories with formal models. This is particularly important in our field as human behavior cannot easily be characterized by a simple set of axioms. Empirical models of behavior can then be used to contrast axioms as they are used in theory. For example, independence of irrelevant alternatives is an axiom typically used in social choice theory. However, experimental research has found that human subjects often change their preferences over two alternatives if faced with an extended set of alternatives.

1.4 Theory and BISE Identity

The theoretical foundation of a scientific discipline has a substantial impact on its identity, and the identity of the IS discipline has led to significant discussion in the past. Some colleagues see themselves in the tradition of computer science and operations research, and they heavily draw on certain branches of mathematics, theoretical computer science (in particular algorithms and complexity theory), and statistics. Some colleagues are closer to economics and draw on economic theory, most notably microeconomics and industrial economics. Finally, the work of many colleagues is rooted in psychology and sociology, in particular when it comes to user perception and adoption of information systems.

Of course, the underlying theory has a substantial influence on the research being done and the criteria used to evaluate research. Some argue that IS needs to develop its own theories, which are distinct from reference disciplines. After all, it is not even easy to characterize what constitutes a theory, and the understanding of this is different in all of these reference disciplines. In any case, the current state of the discussion on theory in IS appears unsatisfactory.

Due to the fact that IT plays a role in more and more aspects of our lives, IS academics have looked into an ever growing number of subjects and IT-driven phenomena. Sometimes these phenomena are related to finance (e.g., crowd funding), sometimes to marketing (e.g., online shopping behavior), sometimes to systems engineering (e.g., enterprise architecture management), and sometimes to labor economics (e.g., online job markets). Nowadays research topics in BISE are largely interdisciplinary. While it is important to analyze all of these topics, our community is not the only one looking at these phenomena. It is important that we bring certain methods and theories to the table – a particular point of view that adds to the work of others in a valuable way. This is one, but of course not the only reason why it is important to be aware of the theoretical foundations of our work.

While some may regard a discussion of theories a mere philosophical exercise, we are convinced that a reflection on the foundations of our work – and its intended outcome – is essential. Without considering the existing variety of theory conceptions in our discipline, we cannot develop elaborate ideas of the ultimate goals of our work, of the justification and evaluation of research, of scientific progress and of proper ways to document scientific knowledge.

1.5 Contributions

We have collected the views of colleagues on the importance and nature of theories in their field. This was intended to not only lead to a summary of different theoretical streams relevant to our research, but it might also influence the discussion about curricula in our field. We asked them to account for the following questions:

- Which conception of theory is central to your area of research?
- How do you evaluate progress in your field and what would you describe as long-term goal?
- In which way does theory guide design and engineering in your field and how does it impact practice?
- How do you evaluate the quality of theories in your field?

The contributions we received confirm that a debate on theory in our field is both challenging and inspiring. It is challenging because there is a variety of clearly different perspectives on the subject that indicates not only that we lack a common conception of theory, but that it might even be illusive to aim at one. At the same time, such a debate promises that "the object of our thought becomes progressively clearer" (Berger and Luckmann 1966) through the multitude of perspectives on it.

David Avison and Julien Malaurent used the opportunity to comment on their contribution to a debate on theory they had organized earlier (Avison and Malaurant 2014). There they questioned the "theory fetish" they observed in IS research and suggested that research would benefit from a more relaxed notion of theory, which they referred to as "theory light". In their present contribution they emphasize that they did not mean to give up the quest for theory in IS research, but that there should be the opportunity for publishing ideas without referring to a rigorous notion of theory. Avison and Malaurent seem to assume that there is a common conception of theory in IS, since they do not discuss the conception of theory as such.

Peter Fettke focusses on particularities of research in Business and Information Systems Engineering (BISE) compared to IS. He argues that IS follows a model of research that has matured in the natural sciences, while BISE is rooted in engineering. While he regards referring to theories as a common, if not mandatory part of research in IS, he suggests that there are conceptual frameworks in BISE that are not called theory, but might as well qualify as such. While Fettke is reluctant to offer a definition of theory, he has a clear preference for a concept of theory that emphasizes the identification of cause–effect-relationships.

Dirk Hovorka proposes an inspiring relativist view on theory. He criticizes the common idea that a theory is a static linguistic structure that enables problem solving as misleading. Instead, he proposes a more dynamic view. Theories, as well as the conception of theory, are in a state of flux, they are representations of the ongoing discourse that constitutes the idea of science. Since such a discourse may stress a multiplicity of different perspectives on the subject of thought, theories may possess different forms and serve different purposes. Therefore, according to Hovorka, it would be inappropriate to aim at a common or integrated conception of theory. At the same time, such a view on theory implies giving up the common idea of scientific progress, because it denies the existence of criteria that would allow a clear discrimination of competing contributions to a common knowledge base.

In their research, Jan Krämer and Daniel Schurr follow a micro-economic paradigm that makes heavy use of mathematical models. Therefore, it does not come as a surprise that the conception of theory they suggest shows clear similarities to the notion of theory in mathematics. They regard models as interpretations of formal theories that help mediating between abstract structures and reality. To serve this purpose, models need to be designed with assumptions about the targeted domain in mind, which in turn requires some sort of empirical analysis. Hence, they claim that models serve as an instrument to develop appropriate formal theories that can be turned into theories with an empirical claim. They do not, however, advocate a pure realist conception of models. Instead, they regard models as analytical tools that may on purpose deviate from factual properties of reality.

Benjamin Müller distinguishes between positivist and non-positivist conceptions of theory and poses the question which one is more appropriate. He argues that scientific progress is likely to result from integrating and consolidating findings that are brought about by different research methods and paradigms. Consequently, he proposes that accounting for multiple perspectives should be a pivotal criterion for evaluating the quality of theories. He also advocates the conduction of research on post-adoption, that is to go beyond simplified models of technology adoption and focus on new patterns of (inter-) action that may emerge after the adoption of new technologies.

Leena Suhl's view on theories reflects her work in operations research. She argues that operations research calls for enriching formal theories with empirical theories from the targeted domains, especially from economics, but also from fields such as manufacturing or marketing. Suhl suggests that the use of different types of theories contributes to the strength of the field, because it requires looking at the research subject from different perspectives. Therefore, she advises against aiming for a common conception of theory or even a comprehensive unified theory in Business and Information Systems Engineering. Instead, she suggests building and maintaining a common repository of relevant theories and methods that foster reuse.

Bernhard Thalheim argues that conceptual models are indispensable instruments of research in our field.

Therefore, he proposes a general model theory that is suited to guide the more reflected construction, use and evaluation of models. For this purpose, he suggests a conception of model and discusses its relationship to the concept of theory. Since he regards models as primary subjects of scientific thought, he recommends supplementing a general model theory with a theory of reasoning that would include foundational elements of reasoning about the construction, analysis, and use of models.

Prof. Dr. Martin Bichler
Technical University of Munich

Prof. Dr. Ulrich Frank
University of Duisburg-Essen

2 A Call for 'Theory Light' Papers

In our original paper published in *Journal of Information Technology* (Avison and Malaurant 2014), we argued that papers in our top journals need not only emphasize theoretical contributions, but could also, for example, emphasize new arguments, facts, patterns and relationships and thereby be 'theory light' and yet still make a major contribution to the discipline of information systems (IS). We gave some examples of such papers from IS and other management disciplines. We also provided several reasons for our concern about the present stress on theory in our journals, giving full explanations in that original paper:

1. Authors may be tempted to revert to 'ideal types' in our understanding process to make sense of the data within a theoretical framework.
2. Authors may be tempted to distort the description of the research setting so that it fits better to the chosen theory or theories.
3. There is no 'recipe' to help authors somehow fit the data to a theory and too few reflective accounts about how any potential gap between theory and data can be addressed, so that authors may be tempted to choose only those data that fit the story.
4. Authors may be tempted to choose theories that might be more related to 'fashion' or the fact that a theory developed in another discipline has yet to be 'borrowed' into IS, in order to provide an 'original' theoretical contribution, rather than to select a theory on the grounds of suitability considerations.
5. The requirement to emphasize theory in all our published papers has an opportunity cost as authors loose the opportunity to make other valuable contributions fully because of space issues. To move into 'unexplored territories and arguments' requires supporting explanations etc. to make the contributions convincing.
6. The requirement of a theoretical contribution in every paper makes some of these 'contributions' somewhat trivial. Many papers may contain 'theoretical filling' rather than making a substantial theoretical contribution. It is this 'window dressing' which downplays theory as it does not give theory the weight it deserves and suggests that IS is 'weak theoretically'. Thus IS papers that do stress theory should deepen IS theory rather than simply 'add to the mass'.

As we stated in our original paper, all these concerns are not about appropriate emphasis on theory, but about the danger of inappropriate emphasis or inappropriate use of theory or theoretical frameworks. We therefore argued for (and provided examples of) some papers being 'theory light' where theory plays (or pretends to play) no significant part in the paper and the contribution lies elsewhere.

We are particularly concerned that too few papers published in the top journals of our discipline impact practice. Articles published are often posteriori interpretations of cases or datasets and the connections between academic IS researchers and practitioners remain too limited and uncertain. For this reason we have been particularly keen to promote the use of action research (Avison et al. 2016).

Our paper has had the impact to lead, for example, to six rich commentaries published in the same issue of *Journal of Information Technology*, but it has also sometimes been misinterpreted and misrepresented. For that reason we now emphasize what we did not say! For example:

1. We did not argue for a theoretical or theory-free research. This suggests an anti-theoretical stance that we do not share. We argue for papers to be accepted in our top journals that either make an excellent theoretical contribution or that make an excellent contribution elsewhere.
2. Our position is not the same as that of a grounded theorist who might start from a tentative theory-free stance but when making sense of the data is expected to create theory. Therefore papers based on the grounded theory approach are expected to discuss theoretical contributions of the research.
3. We did not argue that theory should not be a key element of doctoral studies. Doctoral students should have a thorough grasp of theory. They need to demonstrate knowledge and use of theory as part of their qualification.
4. We did not suggest that 'anything goes' in 'theory light' papers. Indeed, we suggested that authors and reviewers ask themselves ten questions which might apply to all qualitative papers, but are especially important in 'theory light' papers. These questions are: (1) Is it interesting? (2) Is it original? (3) Is it rigorous?

(4) Is it authentic? (5) Is it plausible? (6) Does it show criticality? (7) Is there access to the original data? (8) Is the approach appropriate? (9) Is it done well? (10) Is it timely? Again, each of these questions is discussed in the paper.

5. We do not regard writing ‘theory light’ papers to be easier to research or write, nor did we imply a less rigorous reviewing process, a lowering of standards for our leading journals, or an easier read. On the contrary, responding positively to our ten questions above suggests that these contributions need to be especially good ones.

The acid test for any paper (including ‘theory light’ ones) is the following high barrier: Is it probable that the paper will stimulate future research that will substantially alter IS theory and/or practice? Following this path we should see more papers in our leading journals that are truly original, challenging, and exciting, and less – dare we say – formulaic.

Dr. David Avison
Dr. Julien Malaurent
ESSEC Business School

3 Towards a Coherent View on Information Systems

Scientists have odious manners, except when you prop up their theory; then you can borrow money of them. – Mark Twain

3.1 Business Informatics as an Academic Field of Inquiry

Talking about theories depends on the underlying notion of theory. First, I would like to point out that academic fields of inquiry have developed very different understandings of what science and an acceptable theory are. It is impossible to give a complete overview of all answers. However, I would like to open the discourse and make some important preliminary remarks.

Table 1 shows four triples of corresponding words in English, French and German. This synopsis clearly shows that that for the English word “science” different terms are used in German and French (McCloskey 1984). This fact is of major importance because it makes indisputably clear that the terms “science” and “Wissenschaft” are not interchangeable in all sentences without altering the truth value of statements. Hence, speakers from different language communities, particularly from English and German speaking ones, have different conceptions in mind when talking about science or *Wissenschaft*. According to (McCloskey 1984, p. 97), while in German and French the science word “merely means ‘disciplined inquiry,’ as distinct from... journalism or common sense”, in English, the “august word connotes of numbers, laboratory coats, and decisive experiments publicly observed”. In fact, whenever German speakers use the term “*Wissenschaft*” in the sense of *Geisteswissenschaft* or *Ingenieurwissenschaft*, English speakers do not use the term “science” at all.

Therefore, if we talk about Information Systems or Business and Information Systems Engineering (BISE) as a science, our understanding of science has to be clarified. While Information Systems is strongly rooted in science, BISE has its origin in engineering. In the following, I use the term “Business Informatics” – in analogy to Bioinformatics or Health Informatics – as an umbrella term for Information Systems and BISE. Table 2 summarizes the foci of different academic disciplines studying information systems.

3.2 What is a Theory in Business Informatics?

Analyzing the usage of the term “theory” in different communities is one approach to answer the question what a theory is in Business Informatics. Table 3 aggregates the results of two quantitative literature reviews conducted by Lim et al. (2009) (with a focus on Information Systems) and Houy et al. (2014) (with a focus on BISE).

Table 2 Focus of different academic disciplines studying information systems

Natural sciences	Social sciences	Humanities	Engineering
Information systems			
			Business and Information Systems Engineering
Business informatics			

Table 1 Synopsis of terms denoting academic fields of inquiry in different languages (based on McCloskey 1984)

English	French	German
Natural *sciences*	Les *sciences* naturelles	Die Natur*wissenschaften*
Social *sciences*	Les *sciences* sociales	Die Sozial*wissenschaften*
Humanities	Les *sciences* humaines	Die Geistes*wissenschaften*
Engineering	L’ingénierie	Die Ingenieur*wissenschaften*

Table 3 Most cited theories in Business Informatics (The ranking points are calculated as the arithmetic mean of the ranking points a theory obtained by the two rankings. A theory ranked first gets 1 point, ranked second gets 2 points etc.)

Theory	Lim et al. (2009)	Houy et al. (2014)	Ranking points
Technology acceptance model	1	3	2.0
Game theory	4	1	2.5
Transaction cost theory	5	2	3.5
Resource-based view	2	6	4.0
Systems theory	–	4	4.0
Organizational theory	–	5	5.0
Diffusion of innovations	6	9	7.5
Graph theory	–	8	8.0
Theory of planned behavior	6	11	8.5
Theory of reasoned action	3	18	10.5
Decision theory	16	6	11.0
Principal agent theory	21	10	15.5
Organizational learning theory	8	28	18.0
Social cognitive theory	10	44	27.0
Dynamic capabilities	8	89	48.5

These results show:

- Pluralistic orientation: Table 3 only depicts the most cited theories in Business Informatics, in total, more than 200 theories were identified. This result shows that there exists no clear and distinct theoretical research paradigm in the sense of Kuhn (1996). Although there are some competing theories (e.g., resource-based view versus market-based view), most theories have different application areas and can be seen as complementary.
- Theory as an umbrella term: Sometimes the term "theory" is used as an umbrella term for different theoretical approaches, e.g., organization theory, decision theory or systems theory include very different theoretical approaches.
- Different reference disciplines: Theories used in Business Informatics are rooted in different academic fields of inquiry, e.g., microeconomics (game theory), strategic management (resource-based view), or organizational sciences (organizational theory).
- Mathematical and empirical theories: Some theories have an empirical content, e.g., transaction cost theory. The empirical content of other theories is debatable, e.g., systems theory or game theory. Other theories, e.g., graph theory, do not have any empirical content at all.
- Descriptive and normative theories: The term "theory" is used in a descriptive as well as a normative sense. For instance, it is well-known that decision theory has two different branches, normative/rational decision theory and descriptive decision theory.

Although such quantitative literature analyses can give important and interesting insights into the usage of the term "theory" in Business Informatics, it is also clear that such results should be critically reflected: (1) The presented analysis is based on the premise that a theory is present wherever the term "theory" is used. Although the idea that the meaning of a word is given by its usage is appealing, it should be remarked that it would be a classical logical fallacy to derive a normative notion of what a theory is solely from a descriptive analysis. (2) Since the term "theory" is used very differently, it is prima facie plausible that there exists not only one conception of the idea "theory". My following contribution relies on the premise that the term "theory" can be explicated differently.

3.3 Two Major Design Theories in Business Informatics

The analysis above shows that design theories are clearly underrepresented in the top Business Informatics theories (Gregor 2006). However, it cannot be concluded from this result that there are no important design theories in Business Informatics. Note that there are many important theories in other branches of academic inquiry which do not carry the term "theory" in their name, e.g., geometry, thermodynamics or evolution. In fact, some very important research results in Business Informatics are not labeled as theory at all. Let me introduce two examples which have major influence within the German Business Informatics community:

- Model of Integrated Information Systems (IIV) developed by Mertens (2012): The work on this model started in the late 1960s and was further developed for more than 40 years. This model shows how different application systems in the manufacturing industry are conceptually integrated.

- Architecture of Integrated Information Systems (ARIS) developed by Scheer (1994): Scheer developed the ARIS as an instrument to systematize different aspects to describe and develop information systems. For each aspect and layer particular instruments are introduced and integrated. This model was developed in the late 1980s and is still used in different versions.

Although both works can easily be criticized for several reasons (e.g., bias towards manufacturing industry, not every construction step is explicated), the mentioned examples are two major instances of design theories. This is not merely my opinion; the statement can easily be substantiated by taking a look on the history of these contributions (work of Mertens developed up to the 18th edition, Scheer's major work on ARIS is translated into English, Chinese, Russian and other languages). There are numerous examples of dissertations and research articles which are based on the design theories developed by Mertens and Scheer, although the literature analysis shows that they are not explicitly labeled as theory. Furthermore, at many German-speaking universities, these works provide the classical textbook for an introductory course into Business Informatics.

To summarize, although both design theories are not explicitly called "theories" and therefore do not appear in the above mentioned literature analysis, it would be a mistake not to subsume this work under the umbrella term "(design) theories" of Business Informatics.

3.4 Theoretical Progress: A Multi-Perspective Understanding of Theory

At large, there are good arguments to question the idea of scientific progress in general (Kuhn 1996). However, when understanding academic inquiry as a problem solving activity by following a particular research paradigm, I think it is possible to see some important developments which can be called progress. With respect to different research traditions, such a progress can have very different roots and epistemic qualities (Hacking 1983). Figure 1 provides an overview of four main perspectives.

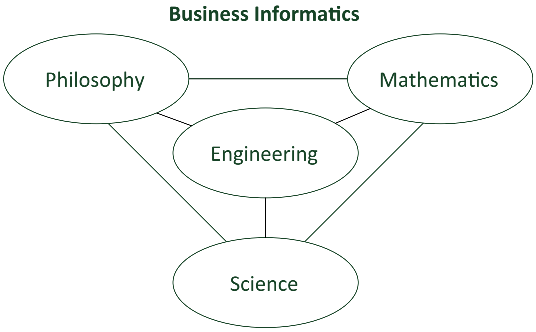


Fig. 1 Different perspectives on Business Informatics

- Business Informatics as mathematics: From the perspective of mathematics, the formal structure of information systems is of major importance. Empirical insights are out of scope of this perspective. As a primary method, a formal proof is used. Progress is achieved by formalizing general ideas and proving interesting statements. Example: Seminal paper by Kindler (2006) introduces and formalizes a framework for formal execution semantics for Event-driven Process Chains (EPC). The significant progress of this work is a mathematically sound definition of the non-local behavior of EPC.
- Business Informatics as a science: Real phenomena are described, explained, understood and often generalized by using a theory about these phenomena. Experiments are the scientific method par excellence. From this perspective, there are different areas for improvement, mainly a theoretical progress [finding a new theory explaining a phenomenon), an empirical progress (identifying or describing a (new) phenomenon] and a methodological progress (improving an existent or inventing a new method). Example: Seminal Paper by Davis (1989) explaining the acceptance of information technology. Davis shows that perceived ease of use and perceived usefulness are high predictors for user acceptance of information technology (theoretical progress). Additionally, he develops and validates measurement instruments for all introduced constructs (methodological progress).
- Business Informatics as engineering: New, more powerful and astonishing information technologies are created in academic or industrial laboratories and ultimately tested in reality. Research and development respectively prototyping are primary research methods. Example: Seminal work by Scheer (1994) on the Architecture of Information Systems (ARIS). The significant contribution of Scheer's work is a comprehensive framework for describing and developing business information systems. Furthermore, a powerful software package was developed which demonstrates the feasibility and usefulness of this innovative approach. The experiences with this prototype provides the foundation for the development of the ARIS Platform which later became the market-leading system for business process management.
- Business Informatics as a philosophy: Developing new ideas and perspectives and criticizing well-known approaches is important for the philosophy of information systems. Speculation, discourse, analysis,

> argument and debate are the major elements of methods used from this perspective. Example: Wand and Weber (1988) present the idea of using ontology as a foundation of information systems research and set the philosophical starting point and foundation for a broad research stream (Fettke 2006). Another example on the meta-level of research on Business Informatics is the seminal work by Hevner et al. (2004) who explicitly discuss the importance of design science research in information systems. Both works mentioned offer very fresh and fruitful views for and on research in Business Informatics. The significant contribution of Wand and Weber is a completely new fundament for conducting research. Hevner et al. introduce clear guidelines for conducting design science.

Again, I would like to point out that the different perspectives often stress different aspects of progress. However, the ultimate goal is to provide a *coherent* view on information systems. Identified contradictions in practice or theory are an important sign of a lack of coherence and call for more research. Furthermore, different perspectives on information systems have to be *integrated*. Such an integration provides a richer picture of how information systems are, can be, or should be.

As I stated before, different academic fields of inquiry have developed different understandings of what a theory is. However, I would like to mention that there exists a standard view on theory in the philosophy of science, which I would like to discuss in more detail in the following.

3.5 A Narrower View on Theory: The Standard View in Philosophy of Science

If you talk about what a theory is, there are of course different answers to this question (Fettke and Loos 2004). In the broadest sense, a theory is the result of an academic inquiry. As such it can be understood as justified true beliefs which are framed and often specifically named. However, the term "theory" is often used in a narrower sense. For example, compare the five theory types described by Gregor (2006), namely theory for: (I) analysis, (II) explaining, (III) predicting, (IV) explaining and predicting and (V) design & action.

Compared to the concept of theory introduced by Gregor, the standard view of philosophy of science is much narrower (Bunge 1998b; Ladyman 2001). According to the standard view, a theory is a cumulating point of scientific endeavor. A theory is a hypothetical-deductive system which contains presumptions and at least one scientific law statement covering a cause–effect-relationship (formalized as $A \rightarrow B$). The Euclidian geometry theory was for a long time the ideal formulation of a theory. However, in the meantime it is well known that Euclid's geometry does not fit together with the real world, other geometry theories have been developed. Furthermore, Newtonian mechanics is an example of another theory in this sense. However, we know that this theory is still successfully applied in everyday reasoning, although it is not correct when very large velocities or very big masses are involved. Under this assumption, relativity theory must be used for correct reasoning.

From my point of view, there are good reasons to identify cause–effect-relationships at the core of an academic discipline or theory (Note that this statement is not a contradiction to my preliminary remarks as long as you accept the unproblematic premise that there are different conceptions of what theory is.). However, as an application-oriented discipline, solely quarrying for cause–effect-relationships is not sufficient. Business Informatics should not only be interested in cause–effect-relationships, but should also research means–end-relationships (Bunge 1998a; Chmielewicz 1994; Zelewski 1995).

3.6 The Importance and Foundation of Technological Rules

Business Informatics investigates information systems. Such investigations aim at representing and explaining *existing* information systems. According to the standard view of theory, a scientific law constitutes the core of a theory. In contrast, an application-oriented discipline such as Business Informatics is not only interested in scientific laws but in technological rules [formally: "B per A!", (Bunge 1998a; Maaß and Storey 2015)]. In other words, Business Informatics works on new, *possible* information systems [Frank (2006); Müller (1990), p. 8]. Two design types can be distinguished. First, a new system can be described ("to-be system"). Although not every time explicitly mentioned, the modus of description is: "It is possible that ...". Such a description represents an information system as it could or should be. Second, a new process can be described ("to-be process"). A planned process describes an action plan of how a possible system can be implemented or how an objective can be achieved.

Technological rules do not represent existing systems; they guide the development of new information systems. It is impossible to assign truth values to statements about possible systems by comparing the stated possibility with actual reality. Instead, one can only ask whether it is possible to implement or to realize such designs or whether it is desirable to make a planned system reality.

Typical examples for technological rules are (Fettke 2008):

- Business Model Engineering: "Customer-orientation improves profit!" (Davenport and Short 1990).
- Business Process Engineering: "Using processes models is more efficient!" (Scheer 1994).
- Business Software Engineering: "Adding people to a late project makes it later!" (Brooks 1975).

The most important question is how such technological rules can be justified. Or, more generally: What is the interdependence between theories (in the narrower sense) and technological rules?

Often, from the perspective of pure science, it is argued that engineering is only an application of such law statements. Although some renowned proponents, e.g., Popper (1957), formulate the idea that theories can easily be transformed into technological rules by so-called tautological transformations, I believe the interrelationship between both concepts is much more complex (Houy et al. 2010, 2015). For example, the following aspects must be taken into account: (1) "Man has known how to make children without having the remotest idea about the reproduction process" (Bunge 1998a, p. 143). (2) Theories are sometimes still used for design purposes even when it is widely accepted that they are not true, e.g., Newtonian mechanics is still used for the calculating satellite orbits. (3) Not every law statement can effectively be used by a technological law statement, e.g., if one has no means to make the antecedent of the law true, it is impossible to use the law by a simple tautological transformation. Nevertheless, knowing the law might be useful for technological purposes. (4) Particularly in Business Informatics it is questionable whether all known empirically identified patterns or regularities qualify as causal relationships. For example, it is debatable whether the construct "perceived ease-of-use" of the *Technology Acceptance Model* has a *causal* effect on system acceptance. (5) Social systems engineers have to deal with self-fulfilling or self-defeating predictions.

To conclude, from an application-oriented perspective it *does* make sense to conduct academic inquires which are *not* theory-grounded (in the narrower sense) but practically successful.

3.7 On the Quality of Theories in Business Informatics

Lack of cumulative research, following short-lived fads and missing long-term, ambitious research goals are well-known shortcomings of our field which many others have criticized before (Hirschheim and Klein 2003; Steininger et al. 2009). Instead of repeating these still relevant deficits, I would like to put more emphasis on another aspect.

In his contribution to this discussion, Dirk Horvoka already referenced Kuhn's concept of the disciplinary matrix which constitutes not only the identity of discipline but also the values of a research community. In other words, it is interesting to have a more detailed look on our disciplinary matrix in order to elaborate on the quality of theories in our field.

The textbooks of a discipline are one important factor constituting the disciplinary matrix. First, textbooks are major sources for introducing students to a field and demonstrating what is well-known and well-accepted in that discipline. Second, textbooks are also useful for practitioners as points of references to most significant results. Metaphorically speaking, they are symbols for the body of knowledge of a discipline.

A few years ago, some colleagues conducted a detailed analysis of Business Informatics textbooks and obtained remarkable and thought-provoking results (Frank and Lange 2004; Schauer and Strecker 2007). I do not want to recapitulate and update this analysis here. Instead, I would like to pose the following question: *How do our textbooks deal with theories?*

Without conducting a detailed analysis of how theories are referenced and described in our textbooks, I conjecture that the theories mentioned before do not play a central role in these introductory texts. This might have different reasons, e.g., it might take some time until a theory that is newly introduced by a major research outlet is included in a textbook.

As said before, there are also well-established theories in Business Informatics (e.g., *Technology Acceptance Model* and the two design theories by Mertens and Scheer mentioned above). I know there are some textbooks which adequately cover these theories. However, other textbooks do not describe or even mention these well-known theories at all. What can be the reason for this omission?

If we exclude the explanation that these textbooks do not represent the disciplinary matrix adequately, one explanation may be that the authors of these textbooks do not identify the mentioned theories as part of the disciplinary matrix of our discipline. If my assumption is true, then it can be concluded that our disciplinary matrix is not coherent anymore, but might be cracked.

3.8 Conclusion

When discussing what theory is and its role in academic inquiry, it must be clear that different fields of inquiry have very different answers to these questions. From the wider perspective of scientific progress, it can be argued that this situation can be harmful but also very productive. However, it is necessary that different fields of knowledge create a coherent view of what information systems are.

According to the standard view of theory in philosophy of science, a theory is a set of statements with at least one nomological law. Such statements are of major importance for the understanding and design of information systems. Although there are some candidates for such statements in the context of Business Informatics, it is clear that there are very few examples which are able to constitute the core of our discipline. However, there exist well-known examples for (design) theories which can be seen as the core of Business Informatics.

In the future, it is necessary to develop a more coherent picture of different approaches to information systems. I propose to distinguish between two types of approaches, namely *black box* and *white box theorizing*. In a black box approach, technology is viewed as a black box whose inner components are invisible to the theory; they are abstracted. Typical examples for black box theories are the *Technology Acceptance Model* or studies on success factors of ERP systems. Such an approach to theorizing has its strengths. It provides a higher level of abstraction because the concrete implementation is not regarded as important for the theory. Furthermore, the complexity of real information systems is effectively reduced.

However, black box approaches are established on the premise that technology is simply *given*. Such approaches are blind with respect to design decisions inside the black box, which might have a huge impact on theorizing about it. Per definitionem, they do not generate knowledge about the inner structure and functions of technology. What our discipline needs are more *white box* theories providing a coherent view on information systems and its *inner* components.

Prof. Dr. Peter Fettke
German Research Center for Artificial Intelligence (DFKI)
and Saarland University

4 Science as Practice: Theory-as-Discourse

4.1 Introduction

When Latour's climate scientist explains why his own claims and not those of the climate-change deniers should be believed, he does not invoke theory or models. He does not summon explanatory power or predictive accuracy. Nor does he retreat to an argument about instruments or data or simulations. Rather he responds, "If people don't trust the institution of science, we're in serious trouble" (Latour 2013, p. 3). He appeals to the fragile and ill-defined institution that engages a specific form of discourse. It is the discourse of science this essay highlights, and the disciplinary context in which the concept of theory makes any sense at all.

The assertions that theory is the pinnacle of research (Gregor 2006; Straub 2009), that scientific knowledge is based on theories, and that the primary contribution to research is theory have become IS folklore and are only rarely contested [for examples see: Avison and Malaurant (2014), Hambrick (2007)]. The claim that "conflicting notions of theory and theoretical contribution, rather than sheer overemphasis on theory" (Markus 2014, p. 342) is the cause of problems for the field assumes that a unitary view of theory is desirable. Further, it obscures the differences among the discursive, material, and instrumental contexts in which theory makes sense. Many authors discuss theory as a thing-in-itself, as an isolated entity to be reified, bounded and celebrated above all else. This preoccupation diminishes the other disciplinary research contributions that are required for a theory to be cogent (Hovorka and Boell 2015). Certainly theory is important and requires attention, but it is critical to position our understanding of theory within the distinctive disciplinary contexts through which theory, as a discourse, is created, critiqued, evolved, and adjudicated.

Through historical analysis, Kuhn captures this discourse in his original sense of *paradigm*, a term he subsequently abandoned for the broader concept of *disciplinary matrix*. This matrix is composed, at least in part, of symbolic generalizations, models, exemplars, instruments, and values (e.g., precision, prediction, generalizability, design). While Kuhn acknowledges that the list is incomplete, its components illustrate some of the shared commitments of a scientific practice.

It is noteworthy that in Kuhn's extensive writing *theory* is not prioritized as a defining component of disciplinary integrity or legitimacy. Instead, disciplines are characterized by their *paradigm* or *disciplinary matrix*. The primary meaning of paradigm (and a key component of the disciplinary matrix) is the *exemplar:* the texts, teaching cases, and narratives which "contain not only the key theories and laws, but also...the applications of those theories in the solution of important problems, along with the new experimental or mathematical techniques (such as the chemical balance in *Traité élémentaire de chimie* and the calculus in *Principia Mathematica*) employed in those applications." (Bird 2011). Theory and models are important but not "king" or the primary contribution of research. The elevation of theory as the premier contribution in scientific practice and the basis of knowledge misrepresents the role of theory in the broader discourse of scientific inquiry.

In Kuhn's *normal science*, scientists are occupied with matching facts and observations to extant theory, and with articulating what is implicit with theory. Scientists must "premise current theory as the rules of the game. His objective is to solve a puzzle... at which others have failed and current theory is required to define the puzzle…" (Kuhn 1965). Theory becomes fixed as a reified entity used to solve specific problems. Discussion in IS frequently focuses on the normal science image of theory as a reified object with essential characteristics. But during *revolutionary science*, in which the fundamentals of a disciplinary matrix change, Kuhn reveals fluidity in the conception of theory among practicing scientists who share the same commitments. The interpretation of a theory and even what it means to be a theory, is subject to situated contestation and revision and is specific to the scientific problem at hand. Kuhn's normal-revolutionary science distinction reveals that there is no clean separation of *a theory* from the disciplinary matrix, the discourse, in which it is embedded. As communities develop and change, theories are contested, supported/rejected, critiqued, expanded or simplified. Accounts of revolutionary science reveal an image of contestation, where ontological perspectives, theories of instruments and measurement, observations, ideas, things, marks, practices, and truth vie for recognition.

From this we can see that theory cannot be cleanly separated from the discourse regarding observations, instruments, measurements, methods, and the values by which scientific activity is evaluated. Every theory *is a discourse* composed of the individual papers which, taken together, present argumentation for a specific account of a phenomenon. This account is only understood by the community based on disciplinary matrix which the community shares and within which the theory is grounded.

The introduction to this special section and some of the contributors acknowledge that IS, BISE, informatics, management science, and other specializations are overlapping, yet distinctive, fields of inquiry. As new research communities and subspecialties proliferate over time (e.g., Big Data, Q-BISE, DSR) there will perforce be many theory discourses between and within disciplines. Within each community, what counts as factual, as a construct, as valid, or as explanatory also changes. The set of publications, conference talks, teaching materials – the discourse – becomes an intellectual space where ideas clash. The theory-as-discourse is an area defined by what we know, but it is also a zone of contestation, not of revolution, but of ideas competing against each other to disclose what worlds are created by theory.

The consequence of conceptualizing theory as an ongoing discursive-instrumental argument rather than a category used to include/exclude specific instances is that there is no essential characteristic form or function of theory. One of Kuhn's central contributions was the recognition that practicing scientists do not follow a set of rules that enable coordinated research activity. Rather, the shared disciplinary matrix of each community is exhibited in the exemplars used to enroll researchers into the practice. Theory and models are only a part of the community's exemplars and are embedded in the discourse in each community. Thus theory-as-discourse takes on a multiplicity of forms and functions including:

- An aspiration – what we wish we knew.
- A condensation – what we think we know.
- A compounding – (nothing accumulates in an unaltered form).
- A guide – what is worthy of our time.
- A value – what is worthy of knowing.

4.2 Reflections on this Special Section

The variety in conceptions of theory as exhibited in this special section evidence the primary argument I have put forward. In summary, different intellectual communities articulate *theory* in a variety of ways. Theory is viewed: (a) as a law-like cause-effect relationship that may be used to develop practical technological rules (Thalheim, in this section), (b) as a set of models, which are themselves simplified abstractions of reality (Kraemer and Schnurr, in this section), and (c) as a foundation for specific domain-oriented sub-disciplines (Suhl, in this section). There is some agreement among these papers that theory differs among disciplines (Avison and Malaurent as well as Fettke in this section). In addition, Mueller (in this section) notes the relationship between different onto-epistemologies that disclose different phenomenon, and the theorizing that identifies and accounts for those phenomenon. For example, the phenomenon of *IS use,* which is grounded in a Cartesian separation of user and object (Weber 2012), is de-centered in a non-dualist ontology (Barad 1996; Riemer and Johnston 2012).

These different conceptions do not present a compelling argument that IS/BISE and design communities should search for a unifying conceptual ground upon which to construct "theory for everyone," or for an integrated conception of theory across communities. Rather they evidence the position argued in this essay that different conceptions of theory are not only inevitable, but are essential, for the different communities within IS/BISE, design and engineering to progress. It is not possible or desirable to reconcile or to integrate the many descriptions of theory such that every science community would agree on a single set of normative criteria. For example, IS is composed of multiple intellectual communities (Larsen et al. 2008). These communities have differing goals and values, and their different ontological foundations disclose different phenomena. Some communities in IS and BISE focus on explaining and predicting known phenomena. Recognizing the multiple forms and interpretations of explanation (Hovorka 2004) and of prediction (Hacking 1999) renders Gregor's (2006) theory types equivocal in that the development and assessment of explanatory or predictive theories differs depending on the specific form of explanation or prediction implicated in the theory discourse. IS design- and engineering-oriented communities are more like architectural practice (Lee 1991) in their focus on creating new realities and emergent phenomenon rather than retrospective explanation or specific future predictions. But they are different practices and consider theory quite differently. In each community the resulting *theory-as-discourse* has different criteria for development, for contribution, for progress, and for adjudication of quality. In some communities, increasing the absolute accuracy of prediction is valuable. In other communities, increasing the business utility of prediction indicates progress. For some the creation of novel or problem-solving artifacts constitutes contribution and intellectual progress. But often *progress* can only be judged in retrospect as technologies or new processes derived from scientific inquiry come to dominate the landscape. Broadly, there are multiple distinctions for progress, including increasing correspondence of representations to observed phenomenon, of coherence of a set of beliefs held to be true, and of pragmatism. These adjudications further illustrate the inevitability of different theory discourses within and among the IS/BISE and design communities as each community enacts theory-as-discourse in relation to its own shared commitments to knowing the world.

A flexible and many-valent theory-as-discourse does not lead to arbitrary or relativistic conceptions of theory. The instrumental and discursive *theory-as-discourse* proposed here is implied by Pickering's "mangle" (Pickering 1995) and by the "motley of science" of Hacking (1992). The dialectic of resistance and accommodation in scientific practice provides severe criteria for objectivity at both community and individual levels. These may include demands for falsifiability, avoidance of post-hoc and ad-hoc modifications, and the preference for theory which predicts new phenomenon over theories that explain what is already known (Pickering 1995). These, and other shared commitments of the *institution* of science are the background upon which communities adjudicate the quality of each theory-as-discourse. As scientific practice is enacted, the instruments, symbolic generalizations, models and values are challenged, supported critiqued, and evolve. The material phenomena themselves resist and push back, revealing a realm in which the researcher and their instruments struggle to make things work (Pickering 1992). Material reality resists capture by experiments, denies measurement, and confounds instruments. Accommodation occurs when researchers enact conceptual, instrumental or other reconfiguration to overcome resistance (Pickering 1995). The dialectic of resistance and accommodation thus results in further changes in the theory discourse. When material resistance becomes extreme, a theory-as-discourse will longer elaborate "a distinct realm of facts, phenomena, and understandings of the world" (Pickering 1995, p. 202), and it is abandoned. For example, Wegener's theory of continental drift (Wegener 1966), first published in 1915, was dismissed as being eccentric, footloose, preposterous, and improbable. But new instruments (e.g., sonar, magnetometers), disclosure of new phenomenon (e.g., ocean ridges and trenches, earthquake zones), new theory (e.g., sea-floor spreading, magnetic field reversal), and new models (e.g., continental drift; lithosphere dynamics) entered the theory-as-discourse resulting in the abandonment of contracting-earth theory and the broad acceptance of Plate Tectonics – albeit 50 years later.

The theoretical discourse culminating in Plate Tectonics illustrates that the phenomenon itself changed as symbolic generalizations, instruments, models and new exemplars become part of the disciplinary matrix. It is only within this discourse, in its entirety, that Plate Tectonics theory makes the world comprehensible. Theory-as-discourse acknowledges the variety of contributions composing a community's *disciplinary matrix* and contextualizes the social-political-material-discursive practice of scientific institutions. This position liberates us from an unresolvable debate on what theory *is* or *should be*. In rejuvenating the discussion of the full spectrum of potential research contributions which constitute a disciplinary matrix, we may restore theory to an appropriate position and regain confidence in the institution of science itself.

Dr. Dirk Hovorka
University of Sydney

5 Microeconomically Founded Information Systems Research

5.1 Introduction

It is our fundamental understanding that the main purpose of IS research, like most other research disciplines, should be the development of robust theories, which can then inform us about the likely answers to our research questions. What is notable, although not unique about IS research is that the research questions we pursue are not only concerned with the understanding, explanation and possibly prediction of real world phenomena, but also with how we can shape the institutions (North 1991; Roth 2002) that govern these phenomena in order to achieve a certain goal (cf. Gregor 2006). In this regard, IS research takes a theory-guided engineering perspective.

Consider the domain of electronic markets, for example. IS research may be interested in why an observed (e.g., technology induced) market behavior occurs, which market outcomes are likely under a given scenario, but also how markets should be designed in order to achieve a desirable outcome.

In the following we will develop and discuss what we call an idealized microeconomically founded IS research process cycle, depicted in Fig. 2, which reflects our view that fruitful IS theories can be built upon formal, analytic models. Such models are in turn founded upon both, stylized facts that are derived from empirical regularities observed in reality, as well as the existing body of knowledge stemming from robust theories. With reality, we denote the object and processes of investigation that research intents to describe or understand. Scientific inquiries are either concerned with realizations of the past or with potential future states. Researchers perceive reality through empirical observation and data gathering, which is naturally constrained and imperfect. Models, which in themselves are the foundation of theory, can then be used to explain, predict and design instances of the real world. Finally, models, and thus also theory, are evaluated and refined with respect to their ability to inform us about past or future real world phenomena. This can be achieved in field or laboratory studies either by validating or falsifying theory-guided hypotheses, comparing a theory's predictions with actual future outcomes or by evaluating the success of theory-informed design proposals and engineering approaches in actual applications.

The herein described research paradigm is more specific than (but not contradictory to) more general IS research paradigms (cf. Frank 2006), such as design science (cf., e.g., Hevner et al. 2004). Nevertheless, we will argue that theories developed under this framework are suitable to pursue all four fundamental goals of IS research, namely analysis, explanation, prediction, and prescription/design (cf. Gregor 2006). It is not our intention, however, to evaluate or judge different IS research approaches, but rather to motivate why we believe that the proposed microeconomically founded research paradigm is one of several appropriate means to rigorously develop relevant IS theories.

5.2 The Building Blocks of Microeconomically Founded Theory Development

5.2.1 *Theory as a Set of Models*

In general, theory has been characterized as the "basic aim of science" (Kerlinger 1986, p. 8) and is often referred to as "the answer to queries of why" (Kaplan and Merton cited by Sutton and Staw (1995), p. 378). According to (Weick 2005, p. 396) a theory may be measured in its success to "explain, predict, and delight".

In explaining our precise understanding of "theory", we start from the premise that the main task of theory is the integration of findings of individual studies into a modular, but coherent body of knowledge that connects research agendas based on a shared terminology and which provides a microfoundation. Revision and extension of theory is achieved in iterative steps through new or modified models that may either re-investigate central assumptions, thus deepening theory's microfoundation, or create meta-

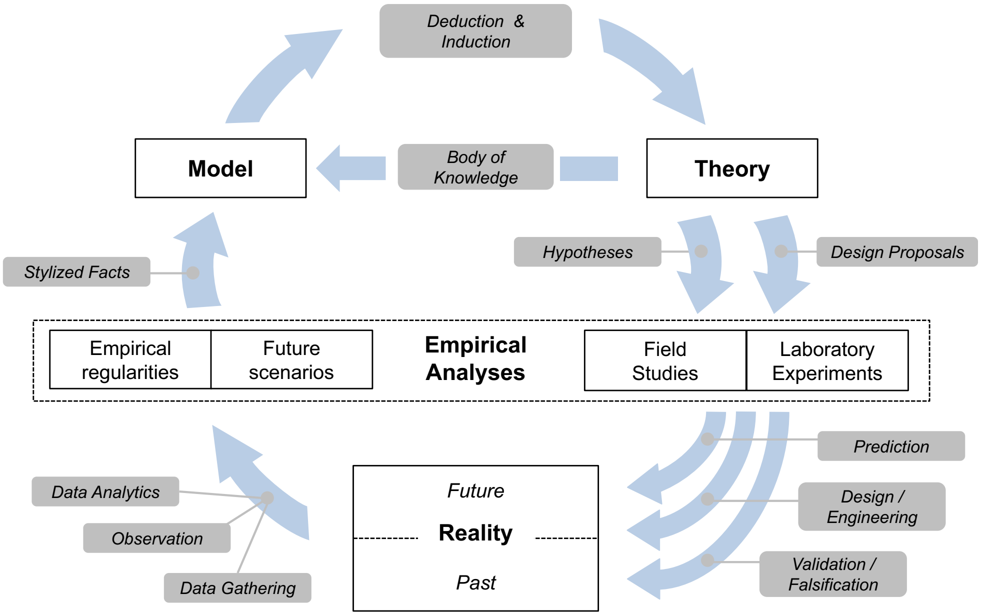


Fig. 2 Idealized microeconomically founded IS research process cycle

models by further abstraction based on the existing body of knowledge. By this means, a mircofounded theory serves as an anchor (Dasgupta 2002) and provides building blocks for new research projects and further theory-building.

In our view, robust theories are the result of deduction and induction from a host of formal models. Therefore, theory can be viewed as a classified set or series of models (Morgan and Knuuttila 2012). In philosophy of science this integral role of models as a part of the structure of theory has been supported by the Semantic View and has been further emphasized by the Pragmatic View (Winther 2015). Consequently, a clear distinction between theory and its models is difficult in general, and even more so if the analysis of theoretical models is deemed as the central part of scientific activity.

At the extreme, a single model can already be the foundation of a theory, although probably not a very robust one. In this regard, the understanding of a robust theory in the social sciences may differ from the understanding of a robust theory in the natural sciences, because theory in the social sciences can be very context dependent, as subjectivity of decision makers, i.e., their beliefs, information, and view of the world substantially shape their choices and actions (Hausman 2013). For example, (Dasgupta 2002, p. 63) noted that "the physicist, Steven Weinberg, once remarked that when you have 'seen' one electron, you have seen them all. [...] When you have observed one transaction, you have not observed them all. More tellingly, when you have met one human being, you have by no means met them all". This is why a robust theory in the social sciences should regularly be built upon a set of models, each of which takes a different perspective on a particular issue and explores a slightly different set of assumptions, such that the boundaries of the theory become transparent.

5.2.2 *Models as the Mediator Between Theory and Reality*

This understanding of theory shifts our attention to the development of suitable models. Models as idealizations (Morgan and Knuuttila 2012) serve as representations of reality that are obtained by simplification, abstraction (see, e.g., the work of Cartwright 2005; Hausman 1990) and/or isolation (Mäki 1992, 2012). But they may also be created as pure constructions, i.e., exaggerated caricatures (Gibbard and Varian 1978), fictional constructs (Sugden 2000), or heuristic devices that "mimic [...] some stylized features of the real system" (Morgan and Knuuttila 2012, p. 64). Gilboa et al. (2014) suggested that economic models serve as analogies that allow for case-based reasoning and contribute to the body of knowledge through inductive inference rather than through deductive, rule-based reasoning. We advocate the use of formal, analytic models in this context, because such models allow to make the assumptions transparent that may lead to a proposition and possibly a normative statement upon which a robust theory, and ultimately a robust explanation or prediction can be built. Note that mathematical formalization is a sufficient, but not a necessary prerequisite to develop a formal model, because it allows to precisely formulate its subject domain, making it an "exact science" (Griesemer 2013, p. 299). Moreover, (Dasgupta 2002, p. 70f.) argued that in building a theory "prior intuition is often of little help. That is why mathematical modeling has proved to be indispensible". The analytic approach provides researchers with a toolbox to deal with especially hard and complex problems. By the means of logical verification, propositions can be shown to be internally true with regard to the underlying assumption.

In general, the goal of a model is to "capture only those core causal factors, capacities or the essentials of a causal mechanism that bring about a certain target phenomenon" (Morgan and Knuuttila 2012, p. 53). Such an abstraction is the prerequisite for conducting a deductive analysis within a particular scenario of interest. What we consider to be particularly important in order to develop relevant models is that a model's microfoundation should contain elements of both theory and reality. On the one hand, a model's assumptions should reflect stylized empirical facts that are well grounded in observed empirical regularities or relevant future scenarios. Such empirical facts can be derived directly from gathered data (most likely with measurement error), may already be the result of extended data analysis, e.g., in the form of detected patterns or correlations, or may be identified by means of a literature review (Houy et al. 2015). However, stylized empirical facts need not (yet) be supported by any theory. This enables us also to incorporate insights of theory-free empirical analysis [particularly (big) data analytics or machine learning] into formal models, which may then lead to a theory that can explain the empirical regularities.[1] On the other hand, a model's assumptions may also be derived from the existing body of knowledge, i.e., from theory. This exemplifies the dual view on the relationship between models and theory: Although models are used to advance theory, theory is also used to produce and inform models.

---

[1] In this context, it is worth mentioning that although data analytics may be able to predict what will happen in a specific context, similar to a theory, it is still theory-free, because it is generally not able to explain why it happens. Without theory, however, it must remain unknown whether these predictions can be generalized and to what extent they are robust to other application scenarios. Therefore, data analytics differs from the traditional paradigm of empirical analysis, which centers around the falsification or validation of hypotheses, which again requires a theory (although not necessarily in the same sense as proposed here – see, e.g., Diesing (2008) for a more elaborate discussion of the relationship between empirical and formal theory) from which these hypotheses are derived in the first place.

A main line of attack against analytic models is to argue that they are not realistic and thus, model-driven theory is useless, because there is nothing to learn about reality. This criticism is amplified in the field of social science, where models are context dependent, as argued above. This naive understanding, however, falls short. First, as we have just mentioned, good models should be grounded in stylized empirical facts. Second, there is an inherent trade-off between accuracy and generality, achieved through simplicity (Gilboa et al. 2014). Scholars experienced in the domain of modeling generally agree on the fact, that too much complexity in fact impedes the explanatory power and the interpretability of models. For example, (Schwab et al. 2011, p. 1115) stated that in order "to formulate useful generalizations, researchers need to focus on the most fundamental, pervasive, and inertial causal relations. To guide human action, researchers need to develop parsimonious, and simple models that humans understand". In the words of (Lucas 1980, p. 697) "a 'good' model [...] will not be exactly more 'real' than a poor one, but will provide better imitations". In this context, the statistician George Box coined the famous phrase that "all models are wrong, but some are useful" (Box 1979, p. 2), clarifying that a model must inherently be unrealistic in a dogmatic sense (see Mäki 2012 for a discussion), but that models in fact enable us to understand real phenomena by abstracting from the complexity of reality. To exemplify this, (Robinson 1962, p. 33) argued that "a model which took account of all the variegation of reality would be of no more use than a map at the scale of one to one". Of course, an interesting model must also exceed a pure tautology, i.e., the results that can be deduced from its assumptions are usually not a priori clear, but may represent surprising results (Koopmans 1957; Morgan and Knuuttila 2012). This requirement can be paraphrased by a quote that is supposedly due to Einstein: "Everything should be made as simple as possible, but not simpler".

Furthermore, we wish to emphasize that over and beyond the explanatory function of formal models, the modeling process itself may prove to exhibit value for understanding a particular scenario. Moreover, a model is an instrument to express an individuals' perception of a problem and may therefore serve as a communication device. (Gibbard and Varian 1978, p. 669) stated that "perhaps, it is initially unclear what is to be explained, and a model provides a means of formulation".

5.2.3 *Empirical Analyses as the Means to Evaluate Theory*

According to our theory-centric research view, empirical analysis serves two core functions: (1) As described above, empirical analysis is a means to derive stylized facts in order to motivate model assumptions, or likewise, to evaluate the plausibility of proposed assumptions. (2) As will be described next, empirical analysis is also a means to evaluate the quality of a theory as a whole. In the context of IS research, we conceive three main ways in which evaluation of theory can be done.

First, empirical analysis, foremost field and laboratory studies, can be employed in order to falsify [in the spirit of Lakatos and Popper (Hausman 2013; Backhouse 2012)], and more ambitiously to validate, theoretically derived hypotheses. While field studies have the advantage of high external validity, they can be generally challenged on the premises that it is difficult to establish causal effects due to problems of (unobserved) confounding variables and endogeneity. At a fundamental level, this gives rise to doubts whether empirical observations are able to falsify (a fortiori validate) theory at all. These concerns are magnified due to the context-specific nature of field studies and a lack of control over the environment that encompasses investigations. Laboratory experiments may be able to mitigate some of these concerns through systematic variation of treatment conditions, randomization of subjects and augmented control of the researcher. Based on a high internal validity, although at the cost of lack of external validity, isolation of causal relationships is facilitated and falsification of theoretical propositions is more easily justifiable (Guala 2005). Furthermore, laboratory experiments facilitate the process of de-idealization (Morgan and Knuuttila 2012), i.e., the generalization of the model context beyond its well-defined assumptions by successively relaxing the assumptions until the theory's established hypotheses begin to break down. Ultimately, however, laboratory and field studies are complementary means to a similar end.

Second, empirical analysis can evaluate the accuracy of theory-driven predictions over time. Although hypotheses may also be regarded as model predictions, the focus here lies less on falsification of suggested causal relationships, but more on the correct qualitative assessment of the impact of future scenarios. With regard to its ability to predict future states of reality [in the sense of Friedman 1953], a microfounded theory draws from its ability to explain observations at the macro level, based on an understanding of the underlying mechanisms and the necessary conditions. By this means, theory-driven predictions are likely to be more robust to changes of real systems as underlying causes can be identified and theory can be modified accordingly (Dasgupta 2002). Moreover, formal analysis allows for experimentation and evaluation of counterfactuals. Two remarks should be made in this context: First, it must be noted that there exists an inherent trade-off between a theory's simplicity and its predictive accuracy. While a simple model or theory may apply more generally and is able to make more robust qualitative predictions, it will also almost certainly be too simple to make accurate quantitative predictions. In turn, the reverse holds true for complex models. This is akin to what is known as the bias-variance-trade-off in statistics (cf. Hastie et al. 2009). Second, even if a theory's prediction may be accurate, this does not "prove" in a deductive sense that it is valid. We may only apply what is known as abductive inference here, that is we can infer that a theory was sufficient to predict the phenomenon of interest, but not that it was necessary, i.e., the only possible theory to be sufficient.

Third, and possibly most interesting in the context of IS research, empirical studies can serve as a testbed for theory-driven design proposals. In this context, laboratory experiments can be seen as an intermediate economic engineering step, similar to a wind tunnel in traditional engineering, where the design proposals (e.g., a proposed market design or regulatory institution) can be evaluated under idealized conditions that mirror those assumptions under which the theory was developed. If the proposed design performs well (relative to the intended goal) in the laboratory then it should be taken to the field for further evaluation. If, however, the proposed design already fails to perform in the laboratory, then there is little reason to believe that it would perform well in the field (Plott 1987). Consequently, the design, and most probably also the underlying theory, would need revision already at this stage.

5.3 Conclusions

Recently, several scholars in the fields of management (Locke 2007; Hambrick 2007) and IS (Avison and Malaurant 2014), among others, have criticized excessive adherence to theory and argue that a scientific contribution can also be made without the need for theory. While we are sympathetic with this view, we strongly believe that the development of robust theories is at the core of scientific endeavor. However, we also believe that these models and theories should be both, (1) well grounded in stylized empirical facts that are the result of inductive research efforts, as well as (2) evaluated and refined through empirical analyses based on field studies and laboratory experiments. To this end, we have motivated and discussed a microeconomically founded IS research paradigm that we deem suitable to develop theories in our field that are rigorous and relevant. In this spirit, we deem the long term goal of microeconomically founded IS research to be the development of robust and stable theories that have been developed and refined through several repetitions of the depicted research process cycle.

Prof. Dr. Jan Krämer
Daniel Schurr, M.Sc.
Universität Passau

6 Theory in the Age of Post-Adoption

6.1 Introduction

To put first things first: I think of theory and theorizing as the key task of any science and feel that our discipline's attention is increasingly shifting in that direction. This is evidenced by seminal contributions (e.g., Burton-Jones et al. 2015; Gregor 2006; Weber 2012), special sections in key journals (e.g., MISQ and JAIS), and dedicated conference tracks (esp. at ICIS, ECIS, and HICSS). In my opinion, this is a welcome shift from methods to theories – or from *how* to *what* we research – that brings a dormant discussion to the center stage: what is theory?

This shift also comes with controversy: While I personally don't agree to the "theory fetish" Avison and Malaurant (2014) diagnose, I think they do our discipline a great service by recognizing this discussion. However, I believe that this issue's editorial points in the right direction when it refers to Markus' (2014, p. 342) observation that "conflicting notions of theory and theoretical contribution, rather than sheer overemphasis on theory, may lie at the heart of the problem [...]." In light of an increasing recognition of the debate about what theory is, it comes as no surprise that Becker et al. (2015) find that "rethinking the theoretical foundations of the IS discipline" is among the top three grand challenges in our discipline's future development – both in terms of relevance and impact.

6.2 The Field of Post-Adoption

One arena I believe this challenge to be particularly true for is post-adoption. As a response to criticism of simple models of technology adoption, the post-adoption research community is shaping up to develop more elaborate models for what happens across multiple levels once technology starts to interact with individuals' actions and larger organizational, market, and societal structures. The resultant research opportunities resonate with the German Informatics Society's grand challenge of omnipresent human-computer interaction, and socio-technical issues, broadly speaking, are among the key issues in the BISE community as well (Becker et al. 2015). Outside of academia, post-adoption research comes at a time when many organizations are thinking about how to engage in digital transformation in order to leverage modern information and communication technologies.

Of course, this is not a new issue. Its roots date back to the 1970s (esp. Bostrom and Heinen 1977a, b) and beyond (e.g., Emery and Trist 1960; Woodward 1958). Recently, however, post-adoption research has mainly been characterized by an intense ontological and epistemological debate and a resultant fragmentation of its results – that is, its theories.

(1) *Which conception of theory is central to your area of research?*

The two main contestants in this debate come with different conceptions of theory: those advocating ontological separability of social and material aspects on one side, and those promoting ontological inseparability on the other (Mueller et al. 2012). Recently, these camps have begun to rally under new banners such as "critical realism" versus "agential realism" (Leonardi 2013) or "weak socio-materiality" versus "strong sociomateriality" (Jones 2014) respectively.

While the interested reader can find more elaborate explanations of these camps in Leonardi (2013) and Jones (2014), the camps' assumptions about ontology and epistemology are central to the debate on theory. On the one hand, the separability camp subscribes to a realist ontology and a mostly representational epistemology. For them, material and social aspects exist independently of any actor and the theorist's key job is to determine which is which and how they interact once they meet in practice. Works by Mutch (2010, 2013) and Mingers (2000) – who strongly draw on Bhaskar (1979) – investigate how such a paradigmatic setup can facilitate the study of technology in social systems, and papers by Burton-Jones and Grange (2013) or Volkoff, Strong, and colleagues (e.g., Strong and Volkoff 2010; Volkoff et al. 2007) deliver excellent exemplars of how this philosophical position helps develop theoretical models of post-adoption mechanisms and processes.

On the other hand, the inseparability camp grants ontological equality of all entities involved in a phenomenon. These entities, however, do not depend on any objective reality nor are they an attribute of human (or, for that matter, non-human) agency. They rather emerge within entanglements through material-discursive practices. Such an entanglement, or phenomenon, is the ontological entity that is sociomaterial. This means that, ontologically speaking, all phenomena are inseparably social *and* material and that any attempt to separate the two is an arbitrary decision by an agent – be it an actor in one of our studies or the researcher. This stresses a deviation from the representational epistemology discussed above and suggests a shift towards performative (and diffractive) thinking. This camp, rooted in Barad's (2003) work, was made popular in IS by Orlikowski and Scott (esp. Orlikowski 2010; Orlikowski and Scott 2008) and studies by Scott and Orlikowski (2014) themselves and by Schultze (2011) illustrate the tenets of this paradigmatic position and its conception of theory.

While my own thinking increasingly gravitates towards the realist position (e.g., Lauterbach et al. 2014) – mainly because I find respective field studies easier to design – I strongly believe that both positions should not be seen as fundamentally irreconcilable opposites. Rather, I would like to think that there is a level beyond the current discussion on which we could explore how insights from these two perspectives complement each other. However, the (seeming) opposition between the camps creates a key challenge to this (perhaps naive) belief: Are the mostly positivistic conceptions of theory and theorizing still useful (let alone valid) in the neo-positivist world of the critical realists or in the non-positivist world of the agential realists or – particularly – in a world that seeks to move beyond their distinction?

(2) *How evaluate progress in your field? What is a long-term goal?*

It is this challenge that also drives progress: For the last five years, progress in this domain is probably best described by the emergence of new theoretical perspectives and our discipline's increasing command of the underlying paradigmatic positions. While the former is evidenced by a growing number of studies employing some form of sociomaterial thinking (e.g., Hultin and Mähring 2014; Introna and Hayes 2011; Johri 2011; Jones 2014), the latter is underlined by the various attempts to better structure the debate's philosophical roots (e.g., Jones 2014; Leonardi (2013).

However, a challenge I see in this is the fact that many seem to have been motivated by some instance of paradigmatic inconvenience to develop an own variant of the ontological and epistemological foundations. Looking at the larger body of sociomaterial studies published recently, irreconcilable differences seem to hamper our discipline's ability to integrate and synthesize theoretical findings I argued for above – an essential prerequisite for the development of a cumulative tradition and a competition of theories to retain the most powerful explanations (Weick 1989).

Consequently, a long-term goal I think worthy of exploration is to turn away from a theory for every one towards a theory for everyone – even if we may have to stop calling it theory then. That is, carefully discussing if and how paradigmatic differences influence our findings, what we mean when we talk of theory, and our ability to compare, contrast, and combine insights into the interplay of technology, social structures, and individuals' behaviors. Hovorka (this section) makes an excellent observation when he points out that the different communities involved in such an integration effort will likely also realize differences in what they mean by theory and how they judge its progress and quality. Nevertheless, I feel that this plurality of perspectives still gravitates around the interplay of technology, social structures, and individuals' behaviors as a common phenomenon. Wouldn't it thus seem logical to try to learn from each other? To me, this thought resonates with the debate between Avison and Malaurant (2014) and Markus (2014) as much as it seems to be on Barad's (2003) mind. Also, a debate seeking to transcend philosophical differences seems a promising approach to not simply reproduce the philosophical discussions from outside the BISE community, but to actually contribute to advancing these debates – a concern for this domain that can be traced back as far as Williams' and Edge's (1996) seminal paper. Consequently, in order to help the post-adoption domain and its theories grow, revisiting paradigmatic assumptions to explore options for complementarity of findings is an essential prerequisite for integrating and consolidating our various findings towards a shared understanding.

(3) *How is theory guiding design and engineering and how does it impact practice?*

While much of the debate in this field might seem esoteric, I see three important links between this paradigmatic debate and practice. First, I believe that our research in this domain enables managers to better express their experiences. This is inspired by a steering committee meeting I attended three years ago in which I pitched the post-adoption research my team and I intended to do to a potential host company. While the team and I expected that the philosophical aspects might be ill-matched to the audience, the participating executives quickly adopted the concepts presented to them and retold their experiences in this newfound language. The ensuing discussion allowed them to make sense of each other's experiences, pinpoint problems, and devise solutions – and resulted in exciting insights for research.

Second, I see important links to the design and engineering of future systems. Insights from this domain of IS research are beginning to shed light on how people interact with technology, make sense of it, and transform what they do through it (e.g., Burton-Jones and Grange 2013; Liang et al. 2015) as well as on how we design the projects that introduce these technologies (e.g., Strong et al. 2014; Wagner et al. 2010). While not yet prominent, some IS research hints towards this research's impact on how we design technologies and their interfaces, particularly when recognizing material properties and their impact on resultant practices (e.g., Jones 2014; Leonardi 2012). I like the thought Brynjolfsson and McAfee (2014) introduce: Increasingly, we will have to think of technology and how we design it not (only) as a potential replacement for human work, but as a meaningful augmentation that complements human work. This will lead to new forms of technology and interface design just as much as to new patterns of interaction between humans and technology. In the long run, this understanding will inform the development of truly intelligent and self-adapting technologies.

Third, on a more abstract but all the more important level, better understanding of what technology is, how we relate to it, and how it shapes our lives also has an ethical dimension. While underexplored in our field thus far, technology is in the process of fundamentally reshaping our life and how we live it.

Taking these three together, advanced sensemaking and expression will allow for expanded description, analysis, and explanation of the interplay of technology, social structures, and individuals' behaviors. Such an improved understanding of post-adoption research's key phenomenon will transform technologies, behaviors, and social structures. Thus there seems to be nothing quite so practical as a sound understanding of what technology means for us, how we relate to it, and how it influences our behaviors; all of which needs to ground on a sound paradigmatic understanding of the theories we develop to help explain these issues.

(4) *How do you evaluate the quality of theories in your field?*

Much like elsewhere, the basic evaluation of theories in the post-adoption field is conducted through a social process towards consensus among a panel of reviewers, editors, and authors. The key tenet of this process to me, especially for conceptual pieces mostly focused on theory and theorizing, is to see if a new theory proposed succeeds in convincing peers. To this end, its power to transform our thinking is one of the key aspects I believe to be important in new theoretical contributions. This resonates strongly with DiMaggio's (1995) idea of theory as narrative with a touch of enlightenment as well as with my own steering committee experience I shared above.

As such, the question of whether a new theoretical perspective helps to make sense of things we observe in practice, but cannot quite explain so far, seems like a key aspect of a theory's quality. For this, Popper (1980) develops the metaphor of theories as "[...] nets cast to catch what we call 'the world'; to rationalize, to explain and to master it" (p. 59). Again, DiMaggio (1995) offers a brilliant perspective on theory as being constructed "post hoc," which to me suggests that many theories might best not be evaluated by any quantitative indicator, but by their potential to inspire and transform thinking.

This also alerts us to the fact that no theory should be looked at in isolation. Beyond any one single theory alone, a good theory also engages in a detailed discussion of rivalry explanations, boundary spanning constructs, and its own boundaries. While often neglected in complex manuscripts already pressured for space, this engagement with what else we know is essential to link any theoretical insight back to the larger discourse and its attempt to build a cumulative core of knowledge on the phenomenon we study. Based on own experiences (e.g., Mueller and Raeth

2012), I particularly appreciate multi-paradigmatic and multi-theoretical work that consciously compares and contrasts what we can see from one perspective with what we would see from another. In the long run, such comparative working will contribute to what Weick (1989) calls disciplined imagination, that is, theorizing as a process of variation, selection, and retention.

Of course the ability to do so depends on understanding the underlying paradigmatic assumptions and on being willing to focus on commonalities and overlaps rather than differences. Above, I hinted towards my belief that the post-adoption community is not yet at a point where such a synthesis is possible. The last five years rather seem to inspire the metaphor of the "Tower of Babel" instead of letting us hope for the coming of a "Babelfish" for theories and insights (as borrowed from Douglas Adams' best-selling "Hitchhiker's Guide to the Galaxy" series).

6.3 Challenges on the Way Ahead

In the next five years, however, I am confident that this domain will witness a tremendous discussion and – hopefully – advance of theory and theorizing. Regardless of which of the above mentioned camps researchers subscribe to, both will likely be united in their quest for post-positivist theories; neo-positivist, realist scholars on one side and non-positivist scholars on the other. This will come with a shift away from the conceptual monopoly positivistic, representational constructions of theory have held in the discourse so far. In fact, the upcoming working conference of the IFIP working group 8.2 to be held this December just before ICIS has set out to explore "new encounters with technology and organization" that go "beyond Interpretivism" (from the call for papers) and I am excited to see what this will produce.

Future debates like this will have to address a wide spectrum of issues: from the redefinition of basic theory taxonomy (e.g., is the term "construct" also applicable to describe theories that do not follow a realist ontology and a representational epistemology?) to quite practical concerns (e.g., means of representation; Gregor 2006). This will also lead to an intense debate on what theory really is and new quality criteria that theories have to live up to, preferably also across paradigmatic positions (see, e.g., Burton-Jones et al. 2015 or Lee 2014 for notable early contributions). Reading Hovorka's contribution to this section, I feel that the post-adoption community is on the brink of realizing and discussing its theories-as-discourses – both in terms of their contents (immediate theories) as well as on a philosophical level (meta-theoretical considerations). While the current fragmentation of these discourses seems to hamper the integration of our various understandings of the post-adoption phenomenon, its heterogeneity must not be seen as something evil per se. Quite to the contrary, I join Scott and Orlikowski (2013) in appreciating the plurality of current studies and also think that Lyytinen and King (2004) make an excellent point when they advocate plurality as a driver of innovation that makes sure that a discipline stays current and maintains a reasonable level of plasticity to adapt to changes in the phenomena it studies.

At the end of the day, all research in this domain strives to better understand the interplay (or intraplay) of technology, social structures, and individuals' behaviors. In the years ahead, I personally hope that the focus will not only be on the content (i.e., the theory itself), but also on two equally important aspects: First, the meaning of theory – or what comes beyond theory – in order to help integrate what we learn about post-adoption. Second, the process of theorizing in order to help aspiring theorist – like myself – hone the skills and crafts of writing and reasoning that are theorizing.

Dr. Benjamin Müller
University of Groningen

7 Business and Decision Analytics in BISE: How much Theory do we Need?

As a scientific discipline, BISE is based on a theoretical foundation that includes different theories depending on the focus and perspective of a given subcommunity. The BISE subcommunity, due to its focus on analytical methods and decision support systems, uses quantitative methods to build and analyze descriptive, predictive and prescriptive models that support decision makers in practice. Here we use the term "Business and Decision Analytics" for this subarea. The quantitative methods draw from a rich theoretical basis in mathematics, statistics, computer science, and operations research, among others. It is not a main goal of BISE researchers to develop new theories in mathematics or operations research, but they need understanding of theory in order to be able to select a right solution approach for each problem and research task. As a generalization and abstraction, new theoretical findings can be established based on BISE research in this area.

Theories in statistics, artificial intelligence, and data modeling form the basis of business and decision analytics, and researchers develop new models and methods to analyze data and compute various indicators to guide business decisions. Mathematics, algorithm theory, and software engineering are important to guide business analysts and software developers in building optimization systems to compute optimal or near-optimal solutions for complex decision problems in business applications.

The models that represent decision problems from practice tend to be quite large and difficult, so that solution methods are needed which can cope with large models and can scale these according to the needs from practice. Knowledge of complexity theory helps researchers to classify algorithmic solution methods and be able to judge their suitability for a given decision problem. It is not a main goal of a BISE researcher to prove worst-case complexity of an algorithm, but rather to assess which methods are able to generate best possible solutions that can be realized in practice with today's technologies.

Fuzzy set theory or alternative uncertainty theories, including stochastics, can be the basis for modeling approaches with respect to preference elicitation and optimization, when the data available is uncertain. Discrete event simulation traditionally uses stochastic distributions to model uncertain data. Decision theory can be used as a basis for designing systems for multicriteria decision support. Some decision support approaches can be built using game theory to represent autonomous actors in agent-based systems.

Modeling is a very important step in developing solutions for decision situations. The best modeling approach should be selected based on the structure and goals of the decision problem. Optimization models, simulation models, data mining models and multicriteria decision models, among others, have their own application areas, and each modeling technology requires a certain structure of the decision problem. A unified modeling theory is still missing and would be helpful for selecting a suitable modeling approach (see Thalheim, in this section).

A main challenge the business and decision analytics subcommunity faces today is the increasing complexity of decisions in the progressively dynamic environment of today's business, especially in supply, manufacturing and service networks (see Fink et al. 2015; Mertens et al. 2015). The increasing interaction of various entities in complex business networks is not yet well understood. Simultaneously today's powerful information technology allows for the use of large amounts of structured digital data for decision-making. "Big data" together with cloud technologies provide much more opportunities to analyze and generate supporting information for decision makers than has been realized until now.

A main research goal of the business and decision analytics subcommunity is to develop new models, methods and systems to be able to model and analyze the complex networks and interactions of their entities. New approaches are needed that include uncertainties and consider robustness aspects, thus providing support to help practitioners improve decision making. To achieve this goal, an interdisciplinary approach is necessary. We need expertise in modeling, algorithms, software engineering, and business theories.

Long-time research goal of the business and decision analytics subcommunity is thus to develop and improve models and methods that help to understand and analyze the dynamic environment of today's business. Evaluation of research progress should therefore assess to what extent new decision models cover relevant areas in business that have not been fully understood until now, as well as how good the methods are which have been proposed to solve and analyze the models. The models and methods developed should be evaluated considering problem structure and needs from the business world, and the same should be done simultaneously with the scientific state-of-the-art and relevant theory. The natural goal is thus to combine rigor and relevance and to produce relevant research results on a high level of scientific rigor.

An expert in research and/or practice of business and decision analytics needs interdisciplinary skills and usually combines knowledge of several disciplines such as information systems, mathematical models and methods, business processes, computer science, software engineering, and data science with decision support techniques. In these disciplines theories have been developed that build a theoretical foundation and thus establish the discipline as a scientific research area. Some of the relevant theories are domain-specific and focus on a given application domain, such as ERP, revenue management or recommender systems, and others are of general nature, such as graph theory or complexity theory.

Besides theoretical knowledge, a business and decision analytics professional needs awareness of all competences necessary to complete modeling and system development projects that provide support for business decision makers and processes. Typically, the following competencies are needed:

- To understand the domain and the specific decision problem.
- To select a suitable modeling approach: simulation, optimization, MCDM, data analysis etc.
- To set up a correct model, combining domain knowledge with modeling knowledge and experience.
- To select the right solution approach, its implementation, and configuration.
- If necessary, to develop and test new solution methods.
- To integrate new quantitative models into an existing business information system, incl. design of database interfaces, user interfaces, communication networks, etc.
- To interpret the solution for the decision makers.

Typical textbooks for decision support systems and operations research contain most of the relevant areas (see for

Table 4 Examples of components to be included in a classification system for business and decision analytics

Relevant areas	Examples
Application area	Production, marketing, revenue management, vehicle routing
Specific decision problem	Optimization of movements in operational inbound logistics, Simulation of customer behavior in a company
Modeling approach	Mathematical optimization model, Network model, Stochastic time-based simulation, Monte-Carlo simulation, Clustering, Association analysis
Solution method	Branch-and-cut, Genetic algorithm, Discrete-event simulation, Monte-Carlo simulation, k-means clustering, Apriori algorithm
Solution implementation	MIP-Solver, Library of basic genetic algorithms plus self-development, Software package for discrete event simulation, Data Analytics package ...
Integration into enterprise IS	Database interfaces, UI interfaces, ...
Decision support tools	What-if-analysis, Pie chart, Gantt chart, graphical Pareto front...
Interpretation for decision makers	Recommendations and alternatives from the business point of view

ex. Turban et al. 2014), however, they mostly focus on methodical aspects and ignore many areas that are important from the information systems point of view.

The question arises whether the subarea business and decision analytics in BISE involves or needs its own theories, or if it is sufficient to be based on theories of neighboring disciplines, the combination and integration of which no doubt is a very challenging task in every single project. To my understanding it does not seem promising to try to develop one unified comprehensive theory for the complete subcommunity, it would simply be too multifaceted as well as constantly evolving and without sharp boundaries. Its basis would be many theories from the neighboring disciplines, and an expert should have an understanding of the most important ones and be able to combine various aspects of them in each single research and development project.

However, it might be possible and helpful to develop a classification or taxonomy of business and decision analytics that could be called a theory. Such a structured and comprehensive view (though not necessarily covering all aspects) would help to understand the area and to select the right approach and right methods for a given problem.

Individual researchers and practitioners have collected a lot of experience and established strict rules as well as heuristic thumb rules that help structuring certain decision problems, selecting the right models and methods, and embedding the system components into an existing IS environment. This knowledge and experience may build the basis for a theory in the sense of classification, taxonomy and/or rule system. Such a taxonomy would ideally involve aspects such as application areas, modeling and solving methods, decision support components, as well as integration into business information and communication systems (see Table 4).

A comprehensive taxonomy would be helpful in introducing the area to students and professionals and in communicating the concepts of business and decision analytics. In practice, many objects can be assigned to two and more classes. However, the classification would help assigning an object and selecting the right approach to solve a given business decision task.

Prof. Dr. Leena Suhl
University of Paderborn

8 Towards a Theory of (Conceptual) Models

8.1 Introduction

A *theory* is in general any systematic and coherent collection of ideas that relate to a specific subject. The notion of theory varies in dependence on scientific disciplines (Kondakov 1974; Seiffert and Radnitzky 1992; Thiel 2004).

1. A theory can be understood as a practice-oriented apprenticeship, as a counterpart of acting and of practice, as a systematic generalization of experience, and a system of main ideas.
2. A (scientific) theory is a “systematic ideational structure of broad scope, conceived by the human imagination, that encompasses a family of empirical (experiential) laws regarding regularities existing in objects and events, both observed and posited. A scientific theory is a structure suggested by these laws and is devised to explain them in a scientifically rational manner. In attempting to explain things and events, the scientist employs (1) careful observation or experiments, (2) reports of regularities, and (3)

systematic explanatory schemes (theories).” (Bosco et al. 2015).

3. A theory can also be understood as an offer, i.e., a scientific, an explicit and systematic discussion of foundations and methods, with critical reflection, and as a system of assured conceptions providing a holistic understanding. Many scientific and engineering disciplines use this constructive understanding of the notion of theory. A constructive theory is a collection of settled instruction conceptions (e.g., concepts, rules, laws, conditions) for (system) development within practical (technical) and quality (esthetic) norms, according to the goals of construction, and guided by some background. A theory is understood as the underpinning of engineering similar to architecture theory (Semper 1851) and the approaches by Vitruvius and L. B. Alberti. Constructive theories in Computer Science and Business Informatics use as their sources four kinds of methods: systematic (deductive mathematical or inductive logical), engineering-oriented abductive or compositional, application-driven, and electronics-oriented component methods.

A theory in the third sense combines explicative and prognostic functions. It is applicative, explicate, exploitative, expiative, explorative, and implicative from the one side, and it is preindicating, prognosticative, and predictive from the other side. Gregor (2006) associates models with construction-oriented theories for the area of information systems. She distinguishes (1) theories for analyzing, (2) theories for explaining, (3) theories for predicting, (4) theories for explaining and predicting, and (5) theories for design and action. Her main attitude is, however, construction models for analysis, explanation, prediction, and construction.

8.2 Models – The Third Dimension of Science

Models are one of the – if not *the* – central elements of Computer Science and Business Informatics. The research in these disciplines considers models as artifacts that are constructed in a certain way and prepared for their utilization. Models might also be mental models and thought concepts. Models are used in utilization scenarios such as construction of systems, verification, optimization, explanation, and documentation. In these scenarios they function as instruments[2].

Given the utilization scenarios, we may use models as perception models, mental models, situation models, experimentation models, formal model, mathematical models, conceptual models, computational models, inspiration models, physical models, visualization models, representation models, diagrammatic models, exploration models, heuristic models, informative models, instructive models, etc. They are a means for some purpose (or better: function within a certain utilization scenario), are often volatile after having been used, are useful inside and often useless outside the utilization scenario.

8.2.1 Elements of a General Modeling Theory

A general theory of model should provide answers to questions such as: What is a model? What are its essential elements? Which kinds of models reflect which task and support a solution of which problems? Which methods must be provided for a proper use of the model? Which methods support development and modernization of models? In which cases is the model adequate? What are the limits and where should this model not be used? In which case we can rely on a model? What are good models? Which models are effective? Which properties can be proven for models? How can models be integrated and composed? What are the correct activities for modeling? What is the added value of a model? Who can use the model how? What are the background theories of modeling? Why should this model be used where it is used? In what way? And by what means?

A general modeling theory generalizes the variety of model notions. In this case language matters, e.g., it enables or disables. The theory allows for managing a complexity of models and methods. Model development methods and model utilization methods should be defined in a similar way as in natural sciences. The theory should also refer to good utilization stories and to best practices.

8.2.2 Models Within the Dichotomy of Theory and State of Affairs

Classical science and also Computer Science and Business Informatics consider models to reflect a certain state of affairs, a certain part of reality, or certain observations. They might also depict parts and pieces of a theory. So, models seem to be placed between the state of affairs and theories. Figure 3 shows the classical understanding of this dichotomy.

This two-dimensional reasoning seems, however, too simple. Models form a further and orthogonal means and are different from theories and also different from the state of affairs.

8.2.3 The Development of Sciences

Disciplines often use a combination of empirical research that mainly describes natural phenomena, of theory-

---

[2] An instrument is among others (1) a means whereby something is achieved, performed, or furthered; (2) one used by another as a means or aid or tool (Safra et al. 2003).

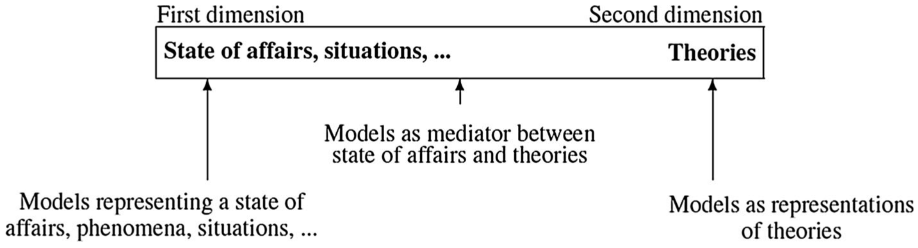


Fig. 3 Models as characterization of situations, representation of a theory, or a mixture of both

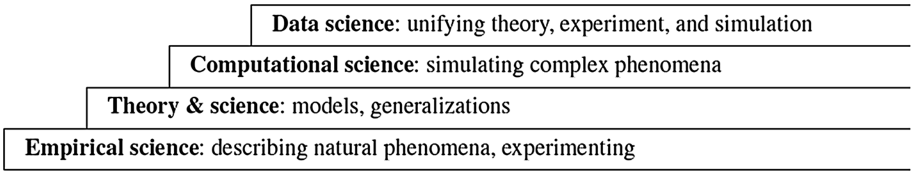


Fig. 4 The four generations of sciences

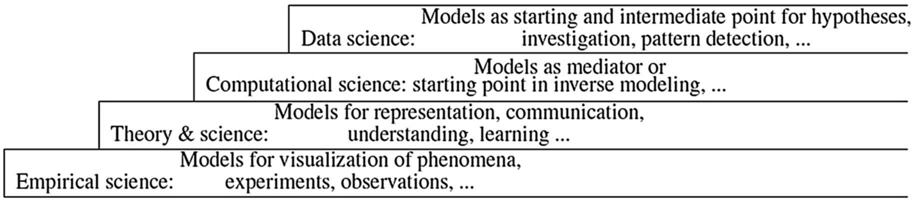


Fig. 5 Some model functions in the four generations of sciences

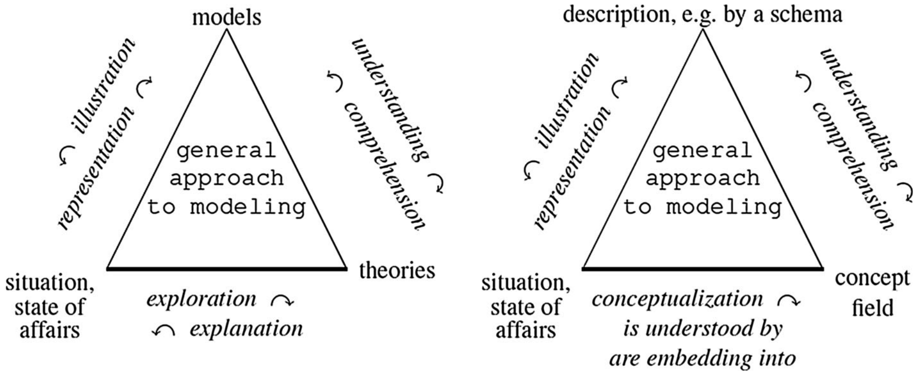


Fig. 6 Models – the third dimension of science and more specifically models in Business Informatics

oriented research that develops concept worlds, of computational research that simulates complex phenomena, and of data exploration research that unifies theory, experiment, and simulation (Gray 2007). Thus Fig. 4 distinguishes four generations of sciences.

Models are a main instrument in all four generations. Their function, however, is different as illustrated in Fig. 5.

8.2.4 Extending the Two-Dimension of the Dichotomy by a Third Dimension

The classical dichotomy of reality and theories should be extended by a third dimension. Theories explain the state of affairs. They are results of explorations of the reality. Models provide an understanding of a theory and illustrate the reality. For Computer Science and Business Informatics, the relationship is similar. We might, for instance, use schemata as models. The theory behind could be, for instance, a concept theory.

Models are therefore the third dimension of science (Thalheim and Nissen 2015a)[3]. Figure 6 depicts this understanding.

8.3 The Conception of the (Conceptual) Model

A model is a well-formed, adequate, and dependable instrument that represents origins.

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of

---

[3] The title of the book (Chadarevian and Hopwood 2004) has inspired this observation.

practice within some context and correspond to the functions that a model fulfills in utilization scenarios.

The model should be well-formed according to specific well-formedness criteria. As an instrument or more specifically an artifact, a model comes with its background, e.g., with paradigms, assumptions, postulates, language, thought community, etc. The background is often given only in an implicit form.

A well-formed instrument is adequate for a collection of origins if it is analogous to the origins to be represented according to specific analogy criteria, it is more focused (e.g., simpler, truncated, more abstract or reduced) than the origins being modeled, and if it sufficiently satisfies its purpose.

Well-formedness enables an instrument to be justified by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through formulas, by falsifiability, and by stability and plasticity.

The instrument is sufficient by its quality characterization for internal quality, external quality and quality in use or through quality characteristics (Thalheim 2010) such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed instrument is called dependable if it is sufficient and justified for some of the justification properties and some of the sufficiency characteristics.

8.3.1 Scenarios and Functions of a Model

Models function as an instrument in some usage scenarios and a given usage spectrum. Their function in these scenarios is a combination of functions such as explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualization, and description-prescription functions. The model functions effectively in some of the scenarios and less effectively in others. The function determines the purpose and the objective (or goal) of the model. Functioning of models is supported by methods. Such methods support tasks such as defining, constructing, exploring, communicating, understanding, replacing, substituting, documenting, negotiating, replacing, optimizing, validating, verifying, testing, reporting, and accounting. A model is effective if it can be deployed according to its objectives.

8.3.2 Conceptual Models

An information systems or database 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 the origin's characteristics.

Conceptual models are models enhanced by concepts and integrated into a space of conceptions[4]. Conceptional modeling is modeling with associations to concepts and conceptions. A conceptual model incorporates concepts into the model. Hence, Fig. 6 can now be revisited for this case and we arrive at Fig. 7.

8.3.3 Reasoning Theory within a Theory of Models

A general theory of reasoning must therefore cover many different aspects. We may structure these aspects by a pattern for specification of reasoning support for modeling acts or steps as follows (Thalheim 2011, 2012b, 2014; Thalheim and Nissen 2015b):

- the modeling acts with its specifics (Thalheim 2010);
- the foundation for the modeling acts with the theory that is going to support this act, the technics that can be used for the start, completion and for the support of the modeling act, and the reasoning techniques that can be applied for each step (Thalheim 2012a);
- the partner involved with their obligations, permissions, and restrictions, with their roles and rights, and with their play;
- the aspects that are under consideration for the current modeling acts;
- the consumed and produced elements of the instrument that are under consideration during work;
- the resources that must be obtained, that can be used or that are going to be modified during a modeling act.

Consider, for instance, the reasoning that aims at realization objectives. It includes specific facets such as

- to command, to require, to compel, and to make someone do something by means of supporting acts such as communicating, requesting, bespeaking, ordering, forbidding, prohibiting, interdicting, proscribing;
- to ask, to expect, to consider obligatory, to request and expect by means of specific supporting acts such as transmitting, communicating, calling for, demanding;
- to want, to need, to require by means of supporting acts of wanting, needing, requiring;

---

[4] White (1994) distinguishes two different meanings of the word 'concept': (1) Concepts are general categories and thing of interest that are used for classification. Concepts thus have fuzzy boundaries. Additionally, classification depends on the context and deployment. (2) Concepts are all the knowledge that the person has, and associates with, the concept's name. They are reasonable complete in terms of the business. Murphy (2001) and Thalheim (2007) define concepts in a more sophisticated form. According to White (1994), conceptions are systems of explanation.

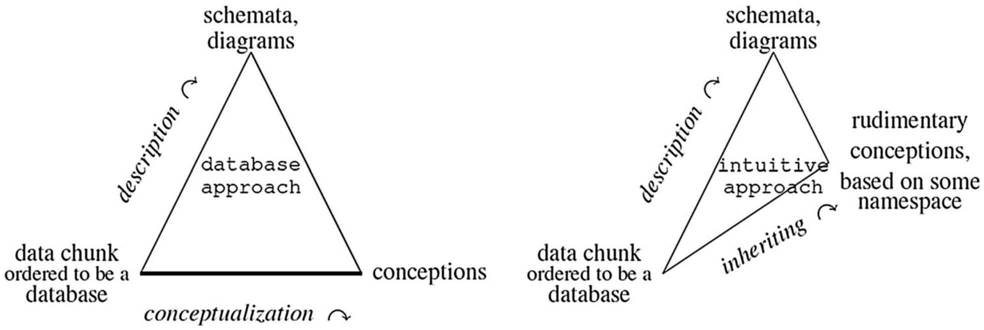


Fig. 7 Conceptual modeling as descriptive or rudimentary conceptual modeling for database models

- to necessitate, to ask, to postulate, to need, to take, to involve, to call for, to demand, to require as useful, to just, or to proper.

The reasoning that is geared towards operating, relevant properties, model objectives, the model itself, towards construction and assessment and guarantees can be characterized in a similar form.

8.4 Theories and (Conceptual) Models

Thalheim and Nissen (2015a) distinguish between ‘models’ (models as representations or artifacts), ‘to model’ (methods of model development and model utilization), and ‘modeling’ (systematic and well-founded matured model development and model utilization; abbreviated as MMM).

8.4.1 Art, Science, and Culture of Modeling

Art (in the broader sense, e.g., used in D.E. Knuth’s “Art of Programming”) is based on creative skills and imagination in the MMM community and produces models as instruments for an easy and simple way of utilization in given scenarios. It requires conscious development of well-formed models. It intends to be contemplated or appreciated as adequate and dependable. We claim that an MMM art has already been developed but is not yet compiled into a holistic body of knowledge.

However, engineering requires a creative application of scientific principles to the design or development and utilization of models, to forecast the effect of model application, and to effectively handle co-evolution of systems and models according to the function of models in utilization scenarios. It requires an MMM science and culture.

An MMM science additionally contains methodologies, matured guidelines for modeling practice, well-founded algorithms and methods for development and utilization of models beyond MMM theories. Culture is “a system of shared values, which distinguishes members of one group or category of people from those of another group; culture is therefore intrinsic in the mind of individuals and it can be measured” (Hofstede et al. 2010). An MMM culture is the collective programming of the mind in one MMM community of practice. It will be different in different areas of Computer Science and Business Informatics.

8.4.2 The MMM Theory as a Lacuna of CS and BI Research

Hartmann and Frigg (2014) consider models and modeling as one of the lacunas in modern research: “Models play an important role in science. But despite the fact that they have generated considerable interest among philosophers, there remain significant lacunas in our understanding of what models are and of how they work.” The book of Thalheim and Nissen (2015a) tries to close this gap on the basis of surveys of models, of approaches to the modeling activities, and of modeling in various sciences (archeology, arts, biology, business informatics, chemistry, computer science, economics, electrotechnics, environmental sciences, farming, geosciences, historical sciences, languages, marine science, mathematics, medicine, ocean sciences, pedagogical science, philosophy, philology, physics, political sciences, sociology, and sports). An MMM theory is still one of the difficult research topics in Computer Science and Business Informatics. The development of a settled conception of models is the first step. The next step is the treatment of modelling activities and of modeling. An MMM culture seems to constitute the task of the next decade.

Bernhard Thalheim
Christian-Albrechts-Universität zu Kiel

References

Avison D, Malaurant J (2014) Is theory king?: questioning the theory fetish in information systems. J Inf Technol 29(4):327–336

Avison D, Kock N, Malaurent J (2016) Information systems action research. Special issue of the Journal of Management Information Systems, Call for papers. http://www.jmis-web.org/cfps/JMIS_SI_CFP_AR.pdf. Accessed 15 May 2016

Backhouse RE (2012) The rise and fall of popper and lakatos in economics. In: Mäki U (ed) Philosophy of economics. Handbook of the philosophy of science. Elsevier, Amsterdam, pp 25–48

Balzer W, Moulines CU, Sneed JD (1987) An architectonic for science. Kluwer, Dordrecht

Barad K (1996) Meeting the universe halfway: realism and social constructivism without contradiction. In: Nelson LH, Nelson J (eds) Feminism, science and the philosophy of science. Kluwer, London, pp 161–194

Barad K (2003) Posthumanist performativity: toward and understanding of how matter comes to matter. Signs J Women Cult Soc 28(3):801–831

Becker J, vom Brocke J, Heddier M, Seidel S (2015) In search of information systems (grand) challenges. Bus Inf Syst Eng 57(6):377–390

Berger PL, Luckmann T (1966) The social construction of reality. A treatise in the sociology of knowledge, 1st edn. Doubleday, Garden City

Bhaskar R (1979) The possibility of natutralism, 1st edn. Harvester, Sussex

Bird A (2011) Thomas Kuhn. Stanford encyclopedia of philosophy. http://plato.stanford.edu/entries/thomas-kuhn/. Accessed 5 Jan 2016

Bosco S, Braucher L, Wiechec M (2015) Encyclopedia britannica. Ultimate Reference Suite, Merriam-Webster

Bostrom RP, Heinen JS (1977) MIS problems and failures: a socio-technical perspective part ii: the application of socio-technical theory. MIS Q 1(4):11–28

Bostrom RP, Heinen JS (1977b) MIS problems and failures: a socio-technical perspective. Part I: the causes. MIS Q 1(3):17–32

Box GE (1979) Robustness in the strategy of scientific model building. In: Launer RL, Wilkinson GN (eds) Robustness in statistics. Academic Press, New York, pp 201–236

Brooks FP (1975) The mythical man-month – essays on software engineering. Addison-Wesley, Reading

Brynjolfsson E, McAfee A (2014) The second machine age: work, progress, and prosperity in a time of brilliant technologies, 1st edn. W.W. Norton & Company, New York

Burton-Jones A, Grange C (2013) From use to effective use: a representation theory perspective. Inf Syst Res 24(3):632–658

Bunge M (1998a) Philosophy of science – from explanation to justification, Rev edn. Transaction, New Brunswick

Bunge M (1998b) Philosophy of science – from problem to theory, Rev edn. Transaction, New Brunswick

Burton-Jones A, McLean ER, Monod E (2015) Theoretical perspectives in IS research: from variance and process to conceptual latitude and conceptual fit. Eur J Inf Syst 24(6):664–679

Cartwright N (2005) The vanity of rigour in economics: theoretical models and Galilean experiments. In: Fontaine P, Leonard R (eds) The experiment in the history of economics. Routledge, London

Chadarevian S, Hopwood N (2004) (eds) Models – the third dimension of science. Stanford University Press

Chmielewicz K (1994) Forschungskonzeptionen der Wirtschaftswissenschaft, 3rd edn. Schäffer-Poeschel, Stuttgart

Dasgupta P (2002) Modern economics and its critics. In: Mäki U (ed) Fact and fiction in economics: models, realism and social construction. Cambridge University Press, Cambridge, pp 57–89

Davenport TH, Short JE (1990) The new industrial engineering: information technology and business process redesign. Sloan Manag Rev 31(4):11–27

Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q 13(3):319–340

Diesing P (2008) Patterns of discovery in the social sciences. AldineTransaction, New Brunswick

DiMaggio PJ (1995) Comments on “what theory is not”. Adm Sci Q 40(3):391–397

Emery FE, Trist EL (1960) Socio-technical systems. In: Churchman CW, Verhulst M (eds) Management sciences, models and techniques, vol 2. Pergamon Press, London, pp 83–97

Fettke P (2006) Referenzmodellevaluation – Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Gütekriterien. Logos, Berlin

Fettke P (2008) Empirisches business engineering. Grundlegung und ausgewählte Ergebnisse. Habilitationsschrift. Universität des Saarlandes, Saarbrücken

Fettke P, Houy C, Loos P (2010) On the relevance of design knowledge for design-oriented business and information systems engineering. Bus Inf Syst Eng 2(6):347–358

Fettke P, Loos P (2004) Referenzmodellierungsforschung. WIRTSCHAFTSINFORMATIK 46(5):331–340

Feyerabend PK (1993) Against method. Verso, London

Fink A, Kliewer N, Mattfeld D, Mönch L, Rothlauf F, Schryen G, Suhl L, Voß S (2014) Model-based decision support in manufacturing and service networks. Bus Inf Syst Eng 6(1):17–24

Frank U (2006) Towards a pluralistic conception of research methods in information systems research. ICB Research Reports No. 7, Institute for Computer Science and Business Information Systems, University Duisburg-Essen, ISSN 1860-2770

Frank U, Lange C (2004) Einführende Lehrbücher für Information Systems aus dem Blickwinkel der Wirtschaftsinformatik – Vorbild oder Bedrohung? Institut für Wirtschaftsinformatik der Universität Koblenz Landau

Friedman M (1953) Essays in positive economics. University Press, Chicago

Frigg R, Hartmann S (2014) Models in science. http://plato.stanford.edu/entries/models-science. Accessed 21 Sept 2014

Gibbard A, Varian HR (1978) Economic models. J Philos 75(11):664–677

Gilboa I, Postlewaite A, Samuelson L, Schmeidler D (2014) Economic models as analogies. Econ J 124(578):F513–F533

Gray J (2007) eScience: a transformed scientific method. http://research.microsoft.com/en-us/um/people/gray/talks/NRC-CSTB_eScience.ppt. Accessed 22 May 2016

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

Griesemer J (2013) Formalization and the meaning of “theory” in the inexact biological sciences. Biol Theor 7(4):298–310

Godfrey-Smith P (2003) Theory and reality. An introduction to the philosophy of science. University of Chicago, Chicago

Guala F (2005) The methodology of experimental economics. Cambridge University Press, New York

Hacking I (1983) Representing and intervening: introductory topics in the philosophy of natural science. Cambridge University Press, Cambridge

Hacking I (1992) The self-vindication of the laboratory sciences. Sci Prac Culture 30

Hacking I (1999) The social construction of what?. Harvard University Press, Cambridge

Hambrick DC (2007) The field of management’s devotion to theory: too much of a good thing? Acad Manag J 50(6):1346–1352

Hastie T, Tibshirani R, Friedman J, Franklin J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer, New York

Hausman DM (1990) Supply and demand explanations and their ceteris paribus clauses. Rev Polit Econ 2(2):168–187

Hausman DM (2013) Philosophy of economics. In: Zalta EN (ed) The Stanford encyclopedia of philosophy

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105

Hirschheim R, Klein HK (2003) Crisis in the IS field? A critical reflection on the state of the discipline. J Assoc Inf Syst 4(5):237–293

Hofstede G, Hofstede GJ, Minkow M (2010) Cultures and organizations: software of the mind: intercultural cooperation and its importance for survival. McGraw-Hill, New York

Houy C, Fettke P, Loos P (2015) Stylized facts as an instrument for literature review and cumulative information systems research. Commun Assoc Inf Syst (CAIS) 37(1):225–256

Houy C, Frank J, Niesen T, Fettke P, Loos P (2014) On the usage of theories in the field of Wirtschaftsinformatik – a quantitative literature analysis. In: Loos P (ed) Veröffentlichungen des Instituts für Wirtschaftsinformatik. Saarbrücken: IWi-Heft Nr. 198, Institut für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)

Hovorka DS (2004) Explanation and understanding in information systems. In: 10th Americas conference on information systems, AIS, New York

Hovorka DS, Boell SK (2015) Cogency and contribution in IS research. Paper presented at the 36th international conference on information systems, Ft. Worth, TX

Hultin L, Mähring M (2014) Visualizing institutional logics in sociomaterial practices. Inf Org 24(3):129–155

Introna LD, Hayes N (2011) On sociomaterial imbrications: what plagiarism detection systems reveal and why it matters. Inf Org 21(2):107–122

Johri A (2011) Sociomaterial bricolage: the creation of location-spanning work practices by global software developers. Inf Softw Technol 53(9):955–968

Jones M (2014) A matter of life and death: exploring conceptualizations of sociomateriality in the context of critical care. MIS Q 38(3):895–A896

Kerlinger FN (1986) Foundations of behavioral research. Holt, Rinehart and Winston, Fort Worth

Kindler E (2006) On the semantics of EPCs: resolving the vicious circle. Data Knowl Eng 56:23–40

Kondakov NI (1974) Teorija. In Logics dictionary. Nauka, Moscov

Koopmans TC (1957) Three essays on the state of economic analysis. McGraw-Hill, New York

Kuhn TS (1964) The structure of scientific revolutions. University of Chicago Press, Chicago

Kuhn T (1965) Logic of discovery or psychology of research. In: Lakotos I, Musgrave A (eds) Criticism and the growth of knowledge, vol. 4. Proceedings of the international colloquium in the philosophy of science, vol. 4. Cambridge University Press, Cambridge, pp 1–23

Kuhn T (1996) The structure of scientific revolutions. University of Chicago Press, Chicago

Ladyman J (2001) Understanding philosophy of science. Taylor & Francis, Milton Park

Larsen K, Monarchi D, Hovorka D, Bailey C (2008) Analyzing unstructured text data: using latent categorization to identify intellectual communities in information systems. Decis Support Syst 45(4):884–896

Latour B, Woolgar S (1986) Laboratory life: the construction of scientific facts. Princeton University Press, Princeton

Laudan L (1983) he demise of the demarcation problem. Physics, philosophy and psychoanalysis. Springer, Dordrecht, pp 111–127

Lauterbach J, Kahrau F, Mueller B, Maedche A (2014) What makes "the system" tick? – explaining individuals' adaptation behavior towards effective use in an enterprise system implementation. In: 35th International conference on information systems (ICIS 2014), Auckland

Latour B (2013) An inquiry into modes of existence. Harvard University Press

Lee AS (1991) Architecture as a reference discipline for MIS. Management Information Systems Research Center, Curtis L, Carlson School of Management, University of Minnesota

Lee AS (2014) Theory is king? But first, what is theory? J Inf Technol 29(4):350–352

Leonardi PM (2012) Materiality, sociomateriality, and socio-technical systems: what do these terms mean? how are they different? do we need them? In: Leonardi PM, Nardi BA, Kallinikos J (eds) Materiality and organizing: social interaction in a technological world. Oxford University Press, Oxford, pp 25–48

Leonardi PM (2013) Theoretical foundations for the study of sociomateriality. Inf Org 23(2):59–76

Liang H, Peng Z, Xue Y, Guo X, Wang N (2015) Employees' exploration of complex systems: an integrative view. J Manag Inf Syst 32(1):322–357

Lim S, Saldanha T, Malladi S, Melville NP (2009) Theories used in information systems research: identifying theory networks in leading IS journals. In: ICIS proceedings, Phoenix, Arizona

Locke EA (2007) The case for inductive theory building. J Manag 33(6):867–890

Lucas RE (1980) Methods and problems in business cycle theory. J Money Credit Bank 12(4):696–715

Lyytinen K, King JL (2004) Nothing at the center? Academic legitimacy in the information systems field. J Assoc Inf Syst 5(6):220–246

Maaß W, Storey VC (2015) Logical design patterns for information system development problems. In: Johannesson P, Lee ML, Liddle SW, Opdahl AL, Pastor López Ó (eds) ER 2015 (LNCS 9381). Springer, Berlin, pp 134–147

Mäki U (1992) On the method of isolation in economics. In: Dilworth C (ed) Idealization IV: intelligibility in science. Rodopi, Amsterdam

Mäki U (2012) Realism and antirealism about economics. In: Mäki U (ed) Philosophy of economics. Handbook of the philosophy of science, vol 13. Elsevier, Amsterdam, pp 3–24

Markus LM (2014) Maybe not the king, but an invaluable subordinate: a commentary on Avison and Malaurant's advocacy of 'theory light' IS research. J Inf Technol 29(4):341–345

McCloskey DN (1984) The literary character of economics. Daedalus 113(3):97–119

Mertens P (2012) Integrierte Informationsverarbeitung. Operative Systeme in der Industrie, 18th edn. GWV Fachverlage, Wiesbaden

Mertens P, Barbian D (2015) Beherrschung systemischer Risiken in weltweiten Netzen. Inform Spektrum 38(4):283–289

Mingers J (2000) The contribution of critical realism as an underpinning philosophy for OR/MS and systems. J Oper Res Soc 51(11):1256–1270

Morgan MS, Knuuttila T (2012) Models and modelling in economics. In: Mäki U (ed) Philosophy of economics. Handbook of the philosophy of science, vol 13. Elsevier, Amsterdam, pp 49–87

Mueller B, Raeth P (2012) What you see is what you get? – a comparison of theoretical lenses to study technology in organizations. In: 33 international conference on information systems (ICIS 2012), Orlando

Mueller B, Raeth P, Faraj S, Kautz K, Robey D, Schultze U (2012) On the methodological and philosophical challenges of sociomaterial theorizing: an overview of competing conceptualizations. In: 33 international conference on information systems (ICIS 2012), Orlando

Müller J (1990) Arbeitsmethoden der Technikwissenschaften – Systematik, Heuristik, Kreativität. Springer, Berlin

Murphy GL (2001) The big book of concepts. MIT Press

Mutch A (2010) Technology, organization, and structure – a morphogenetic approach. Org Sci 21(2):507–520

Mutch A (2013) Sociomateriality – taking the wrong turning? Inf Org 23(1):28–40

North DC (1991) J Econ Perspec 5(1):97–112

Orlikowski WJ (2010) The sociomateriality of organisational life: considering technology in management research. Cambridge J Econ 34(1):125–141

Orlikowski WJ, Scott SV (2008) Sociomateriality: challenging the separation of technology, work and organization. Acad Manag Ann 2(1):433–474

Pickering A (1992) From science as knowledge to science as practice. Sci Prac Culture 4

Pickering A (1995) The mangle of practice: time, agency, and science. University of Chicago Press, Chicago

Plott C (1987) Dimensions of parallelism: some policy applications of experimental methods. In: Roth A (ed) Laboratory experimentation in economics: six points of view. Cambridge University Press, New York

Popper KR (1957) The poverty of historicism. Routledge, London

Popper K (1980) The logic of scientific discovery. Unwin Hyman, London

Riemer K, Johnston R (2012) Place-making: a phenomenological theory of technology appropriation. Paper presented at the 33rd international conference on information systems, Orlando

Robinson J (1962) Essays in the theory of economic growth. Macmillan, London

Rorty R (1999) Philosophy and social hope. Penguin Books, New York

Roth AE (2002) The economist as engineer: game theory, experimentation, and computation as tools for design economics. Econometrica 70(4):1341–1378

Safra JE, Yeshua I et al. (2003) Encyclopædia britannica. Merriam-Webster

Schauer C, Strecker S (2007) Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik. WIRTSCHAFTSINFORMATIK 49(2):136–147

Scheer A-W (1994) ARIS toolset: a software product is born. Inf Syst 19(8):607–624

Schultze U (2011) The avatar as sociomaterial entanglement: a performative perspective on identity, agency and world-making in virtual worlds. In: Galletta DF, Liang TP (eds) 32 international conference on information systems (ICIS 2011), Shanghai

Schwab A, Abrahamson E, Starbuck WH, Fidler F (2011) Perspective-researchers should make thoughtful assessments instead of null-hypothesis significance tests. Org Sci 22(4):1105–1120

Scott SV, Orlikowski WJ (2013) Sociomateriality – taking the wrong turning? A response to Mutch. Inf Org 23(2):77–80

Scott SV, Orlikowski WJ (2014) Entanglements in practice: performing anonymity through social media. MIS Q 38(3):873–893

Seiffert H, Radnitzky G (1992) (eds) Handlexikon zur Wissenschaftstheorie. Deutscher Taschenbuch Verlag, München

Semper G (1851) Die vier Elemente der Baukunst. Braunschweig

Steininger K, Riedl R, Roithmayr F, Mertens P (2009) Fads and trends in business and information systems engineering and information systems research – a comparative literature analysis. Bus Inf Syst Eng 1(6):411–428

Straub DW (2009) Editor's comments: why top journals accept your paper. MIS Q 33(3):iii–ix

Strong DM, Johnson SA, Tulu B, Trudel J, Volkoff O, Pelletier LR, Bar-On I, Garber L (2014) A theory of organization-EHR affordance actualization. J Assoc Inf Syst 15(2):53–85

Strong DM, Volkoff O (2010) Understanding organization – enterprise system fit: a path to theorizing the information technology artifact. MIS Q 34(4):731–756

Sugden R (2000) Credible worlds: the status of theoretical models in economics. J Econ Methodol 7(1):1–31

Suppe F (1989) The semantic conception of theories and scientific realism. University of Illinois Press, Urbana

Sutton RI, Staw BM (1995) What theory is not. Adm Sci Q 40(3):371–384

Thalheim B (2007) The conceptual framework to user-oriented content management. In: Information modelling and knowledge bases, vol. XVIII of frontiers in artificial intelligence and applications. IOS Press, Amsterdam

Thalheim B (2010) Towards a theory of conceptual modelling. J Univers Comput Sci 16(20):3102–3137. http://www.jucs.org/jucs_16_20/towards_a_theory_of. Accessed 22 May 2016

Thalheim B (2011) The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling. In: Embley DW, Thalheim B (eds) The handbook of conceptual modeling: its usage and its challenges. Springer, Berlin, pp 547–580

Thalheim B (2012a) The art of conceptual modelling. In: Information modelling and knowledge bases XXII, vol. 237 of frontiers in artificial intelligence and applications. IOS Press, Amsterdam, pp 149–168

Thalheim B (2012b) The science and art of conceptual modelling. In: Hameurlain A et al (eds) TLDKS VI, LNCS 7600. Springer, Heidelberg, pp 76–105

Thalheim B (2014) The conceptual model – an adequate and dependable artifact enhanced by concepts. In: Information modelling and knowledge bases, vol. XXV of frontiers in artificial intelligence and applications, 260. IOS Press, Amsterdam, pp 241–254

Thalheim B, Nissen I (2015a) (eds) Wissenschaft und Kunst der Modellierung. De Gruyter, Ontos Verlag, Berlin

Thalheim B, Nissen I (2015b) Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung, Fallstudien zum Modellbegriff. De Gryuter, Boston

Thiel C (2004) Theorie. In Mittelstrass J (ed) Enzyklopädie: Philosophie und Wissenschaftstheorie, vol. IV. Metzler, Stuttgart, pp 260–270

Turban E, Sharda R, Delen D (2014) Business intelligence and analytics: systems for decision support. Pearson, London

Volkoff O, Strong DM, Elmes MB (2007) Technological embeddedness and organizational change. Org Sci 18(5):832–848

Wagner EL, Newell S, Piccoli G (2010) Understanding project survival in an ES environment: a sociomaterial practice perspective. J Assoc Inf Syst 11(5):276–297

Wand Y, Weber R (1988) An ontological analysis of some fundamental information systems concepts. Paper presented at the international conference on information systems (ICIS), Minneapolis, Minnesota

Weber R (2012) Evaluating and developing theories in the information systems discipline. JAIS 13(1):1–30

Wegener A (1966) The origin of continents and oceans. Courier Corporation

Weick KE (1989) Theory construction as disciplined imagination. Acad Manag Rev 14(4):516–531

Weick K (2005) Definition of "theory". In: Nicholson N, Audia P, Pillutla MM (eds) The Blackwell encyclopedia of management, vol. 11. Organizational behavior. Blackwell, Oxford

White RT (1994) Commentary: conceptual and conceptional change. Learn Instr 4:117–121

Williams R, Edge D (1996) The social shaping of technology. Res Policy 25(6):865–899

Winther RG (2015) The structure of scientific theories. In: Zalta EN (ed) The Stanford encyclopedia of philosophy

Woodward J (1958) Management and technology, 1st edn. Her Majesty's Stationery Office, London

Zelewski S (1995) Das Konzept technologischer Theorietransformationen – eine Analyse aus produktionswirtschaftlicher Perspektive. Universität Leipzig, Institut für Produktionswirtschaft und industrielle Informationswirtschaft

A Conceptual Model for Services

Bernhard Thalheim[1](✉) and Ajantha Dahanayake[2]

[1] Department of Computer Science, Christian Albrechts University Kiel,
24098 Kiel, Germany
`thalheim@is.informatik.uni-kiel.de`
[2] Department of Computer Information Science, Prince Sultan University, Riyadh,
Kingdom of Saudi Arabia
`adahanayake@pscw.psu.edu.sa`

Abstract. Models are a mainstay of every scientific and engineering discipline. Models are typically more accessible to study than the systems. Models are instruments that are effectively functioning within a scenario. The effectiveness is based on an associated set of methods and satisfies requirements of usage of the model. A typical usage of a model is explanation, informed selection, and appropriation of an opportunity. This usage is declared through information and directions for usage or more specifically through an informative model in the case of a service model.

1 Services and the Conception of a Service

Today, the service has gained recognition as the more realistic concept for dealing with complexities of cross-disciplinary systems engineering extending its validity beyond the classical information systems design and development realm [4]. In this respect the service concept combines and integrates the value created in different design contexts such as person-to-person encounters, technology enabled self-service, computational services, multi-channel, multi-device, location-based and context-aware, and smart services [13]. Therefore, the service concept reveals the intrinsic design challenges of the information required to perform a service, and emphasizes the design choices that allocate the responsibility to provide this information between the service provider and service consumer.

1.1 Some Well-Known Service Notions

The service is being defined using different abstraction models with varying applications representing multitude of definitions of the service concept [7]. The increasing interests in services have introduced service concept's abstraction into levels such as; business services, web services, software-as-a-service (SaaS), platform-as-a-services, and infrastructure-as-a-service [2]. Service architectures are proposed as means to methodically structure systems [1,5,16].

There are number of service notations available in the in the literature, and research has looked into the service mainly from two perspectives, (a) from the

© Springer International Publishing Switzerland 2015
M.A. Jeusfeld and K. Karlapalem (Eds.): ER 2015 Workshops, LNCS 9382, pp. 51–61, 2015.
DOI: 10.1007/978-3-319-25747-1_5

low-level technological point of view and (b) from the higher abstract business point of view. These two categories of service descriptions have derived number of service notations. Some of those main stream service notations are:

The *REA (Resource-Event-Agent) ontology* [8,11] uses as core concepts resources, economic event, and agent. The *RSS (Resource-Service-Systems) model* [12] is an adaptation of REA ontology stressing that REA is a conceptual model of economic exchange and uses a Service-Dominant Logic (SDL) [21]. The *model of the three perspectives of services* uses abstraction, restriction, and co-creation. It concentrates on the use and offering of resources [2]. The perspectives addressed by this model are: service as a means for abstraction; service as means for providing restricted access to resources; and service as a means for co-creation of value. The logics behind is the Goods Dominant Logic (GDL) model [22]. *Web service description languages* concentrate on Service-Orientated Architectures (SOAs) for web service domain. Software systems are decomposed into independent collaborating units [14]. Named services interact with one another through message exchanges. The *seven contexts of service design* [6,9,13] combine person-to-person encounters, technology-enhanced encounters, self-service, computational services, multi-channel, multi-device, and location-based and context-aware services description.

1.2 The Explanation, Selection, and Appropriation

Explanation, understanding and informed selection of a tool is one of the main usage scenarios for a software models. People want to solve some problems. Services provide solutions to these problems and require a context, e.g. skills of people, an infrastructure, a specific way of work, a specific background, and a specific kind of collaboration. In order to select the right service, a model of the service is used as an *instrument for explanation and quick shallow understanding* which service might be a good candidate, what are the strengths and weaknesses of the service under consideration, which service meets the needs, and what are the opportunities and risks while deploying such a service.

The best and simplest instrument in such usage scenario is the *instruction leaflet* or more generally as a specification of the information and directions on the basis of the *informative model.* We shall show in the sequel that this model of a service extends the cargo dimension [10] to the general notion of the informative model. Such models of a service enable people in directed, purposeful, rewarding, realistic, and trackable deployment of a service within a given usage scenario, i.e. use according to the qualities of the model [4]. After informed selection of a service, it might be used in the creation of new work order based on the assimilation of the service into the given context, i.e. appropriation of the service.

1.3 Developing a Service Model Based on the W*H Frame

Systems are typically characterised by a combination of large information content with the need of different stakeholders to understand at least some system aspects. People need models to assist them in understanding the context of their own work and the requirements on it. We concentrate in this paper on the support provided by models to understand how a system works, how it can be used or should not be used, and what would be the benefit of such a model. We illustrate this utilisation of models for services.

We develop a novel service model based on the W*H specification frame [4]. The W*H model [4] provides a high-level and conceptual reflection and reflects on the variety of aspects that separates concerns such as service as a product, service as an offer, service request, service delivery, service application, service record, service log or archive and also service exception, which allows and supports a general characterization of services by their ends, their stakeholders, their application domain, their purpose and their context.

2 The Notion of a Model

The theory of models is the body of knowledge that concerns with the fundamental nature, function, development and utilisation of models in science and engineering, e.g. in Computer Science. In its most general sense, a model is a proxy and is used to represent some system for a well-defined purpose. Changes in the structure and behaviour of a model are easier to implement, to isolate, to understand and to communicate to others. In this section we review the notion of the model that has been developed in [18–20].

2.1 Artifacts that Are Models

A model *is a well-formed, adequate, and dependable artifact that represents origins.* Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

The model should be well-formed according to some well-formedness criterion. As an instrument or more specifically an artefact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form. A model is used in a *context* such as discipline, a time, an infrastructure, and an application.

Models function as an instrument in some usage scenarios and a given usage spectrum. Their function in these scenarios is a combination of functions such as explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualisation, and description-prescription functions. The model functions effectively in some of the scenarios and less effectively in others. The function determines the *purpose* and the *objective* (or goal) of the model. Functioning of models is supported by methods. Such methods support tasks such as defining, constructing, exploring, communicating, understanding, replacing, substituting, documenting, negotiating, replacing, optimizing, validating, verifying, testing, reporting, and accounting. A model is *effective* if it can be deployed according to its objectives.

Models have several *essential properties* that qualify an artifact as a model. An well-formed artifact is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose.*

Well-formedness enables an artifact to be *justified* by an *empirical corroboration* according to its objectives, by rational coherence and conformity explicitly stated through formulas, by falsifiability, and by stability and plasticity.

The artifact is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics [17] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed artifact is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.

2.2 Artifacts as Instruments in Some Usage Scenario

Models will be used, i.e. there is some usage scenario, some reason for its use, some goal and purpose for its usage and deployment, and finally some function that the model has to play in a given usage scenario. A typical usage scenario is problem solving. We first describe a problem, then specify the requirements for its solutions, focus on a context, describe the community of practices and more specifically the skills needed for the collaborative solution of the problem, and scope on those origins that must be considered. Next we develop a model and use this model as an instrument in the problem solving process. This instrument provides a utility for the solution of the problem. The solution developed within the model setting is then used for derivation of a solution for the given problem in the origin setting.

A similar use of models is given for models of services. Service models might be used for the development of a service system. They might be used for assessment of services, for optimisation and variation of services, for validation-verification-testing, for investigation, and for documentation-visualization. In this paper we concentrate on the *explanation, informed selection, and appropriation* use of a service model. It must provide a high level description of the service itself. This usage is typical for a process of determining whether a service is of high utility in an application. Such usage is based on specific usage pattern or more specifically on a special model that is the *usage model of an instrument as a model.*

2.3 Conceptional Modelling: Modelling Enhanced by Concepts

An information systems 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. *Conceptional modelling*[1] aims to create an abstract representation of the situation under investigation, or more precisely, the way users think about it. *Conceptual models* enhance models with concepts that are commonly shared within a community or at least within the community of practice in a given usage scenario. Concepts specify our knowledge what things are there and what properties things have. Their 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 [15], e.g., lexical or ontological, logical, or reflective.

2.4 Adequacy and Dependability of Informative Models

Models are used in *explanation, informed selection, and appropriation* scenarios. We call such models *informative models.* Their main aim of is to inform the user according to his/her information demand and according to the profile and portfolio. The instrument steers and directs its users which are typically proactive. It supplies information that is desired or needed. Users may examine and check the content provided. Typical methods of such instruments are communication, orientation, combination, survey, and feedback methods.

Users have to get informed what is the issue that can be solved with the instrument, what are the main ingredients of the instrument and how they are used, what is the main background behind this instrument, and why they should use this instrument. They need a quick shallow understanding how simple, how meaningful, how adequate, how realistic, and how trackable is the instrument ($SMART$). They must be enabled to select the most appropriate instrument, i.e. they should know the strengthes, weaknesses, opportunities, and threats of the given instrument ($SWOT$).

The SWOT and SMART evaluation is the basis for adequateness and dependability of informative models. The informative model must be analogous in structure and function to its origins. It is far simpler than the origin and thus more focussed. Its purpose is to explain the origin in such a way than a user can choose this instrument because of its properties since all demanded properties are satisfied. The selection and appropriation of an instrument by the user depends on the explanatory statement on the profile and the portfolio of the given instrument, on coherence to the typical norms and standards accepted by the community of practice, on a statement on applicability and added value of the instrument, and the relative stability of the description given. The instrument usage becomes then justified. Furthermore, the instrument must suffice the demands of such scenarios. The quality in use depends on understandability and parsimony of description, worthiness and eligibility of presented origins, and the added value

---

[1] The words ‘conceptual’ and ‘conceptional’ are often considered to be synonyms. The word ‘conceptual’ is linked to concepts and conceptions. ‘Conceptual’ means that a thing - e.g. an instrument or artifact - is characterised by concepts or conceptions. The word ‘conceptional’ associates a thing as being or of the nature of a notion or concept. Conceptional modelling is modelling with associations to concepts. A conceptual model incorporates concepts into the model.

it has for the given utilisation scenarios. The external quality is mainly based on its required exactness and validation. The internal quality must support these qualities. The quality evaluation and the quality safeguard is an explicit statement of these qualities according to the usage scenarios, to the context, to the origins that are represented, and to community of practice.

2.5 The Cargo of a Model

The cargo of any instrument is typically a very general instrument insert like the package insert in pharmacy or an enclosed label. It describes the instrument, the main functions, the forbidden usages, the specific values of the instrument, and the context for the usage model. Following [10,20] we describe the cargo by a description of the *mission* of the instrument in the usage scenarios, the *determination* of the instrument, an *abstract declaration of the meaning* of the instrument, and a narrative explanation of the *identity* of the instrument.

The mission of a model consists of functions (and anti-functions or forbidden ones) that the model has in different usage scenarios, the purposes of the usages of the model, and a description of the potential and of the capacity of the model. The determination contains the basic ideas, features, particularities, and the usage model of the given instrument. The meaning contains the main semantic and pragmatic statements about the model and describes the value of the instrument according to its functions in the usage scenarios, and the importance within the given settings. Each instrument has its identity, i.e. the actual or obvious identity, the communicated identity, the identity accepted in the community of practice, the ideal identity as a promise, and the desired identity in the eyes of the users of the instrument.

2.6 The Informative Model

The *informative model* consists of the cargo, the description of its adequacy and dependability, and the SMART and SWOT statements. It informs a potential users through bringing facts to somebody's attention, provides these facts in an appropriate form according their information demand, guides them by steering and directing, and leads them by changing the information stage and level. Based on the informative model, the user selects the origin for usage with full informed consent or refuses to use it. It is similar to an instruction leaflet provided with instruments we use. The informative model is semantically characterized by: objectivity; functional information; official information; explanation; association to something in future; different representational media and presenters; degree of extraction from open to hidden; variety of styles such as short content description, long pertinent explanation, or long event-based description.

In the case of a service model, the informative model must state positively and in an understandable form what is the service, must describe what is the reward of a service, and must allow to reason about the rewards of the service, i.e. put the functions and purposes in a wider context (*PURE*). Informative models of a service are based on a presentation that is easy-to-survey and to understand, that is given in the right formatting and form, that supports elaboration and surveying, that avoids learning efforts for their users, that provides the inner content semantics and its inner associations, that might be based on icons and pictographs, and that presents the annotation and meta-information including also ownership and usability.

We shall now explore in the sequel what are the ingredients of such informative instruments in the case of a service model.

3 Service Specifications

3.1 Scenarios and Functions of Service Specifications

To capture the scenarios and functions of service specification we introduce *W*H model* in Fig. 1 that is a novel conceptual model for service modelling.

Service	Service Name			
Concept	Ends	*Wherefore?*		
		Purpose	*Why?*	
			Where to?	
			For When?	
			For Which reason?	
Content	Supporting means	*Wherewith?*		
		Application Domain	Application are	*Wherein?*
			Application case	*Wherefrom?*
			Problem	*For What*
			Organizational unit	*Where*
			Triggering Event	*Whence*
			IT	*What* *How*
Annotation	Source	*Where of?*		
		Party	Supplier	*By whom?*
			Consumer	*To whom?*
			Producer	*Whichever?*
		Activity	Input	*What in?*
			Output	*What out?*
Added Value	Surplus Value	*Worthiness?*		
		Context	Systems Context	*Where at?*
			Story Context	*Where about?*
			Coexistence Context	*Wither?*
			Time Context	*When?*

Fig. 1. The W*H Specification Frame for the Conceptual Model of a Service

The W*H model in Fig. 1 fulfills the conceptual definition of the service concept composing the need to serve the following purposes:

- The composition of the W*H model consisting of *content space*, *concept space*, *annotation space*, and *add value space* as orthogonal dimensions that captures the fundamental elements for developing services.
- It reflects number of aspects neglected in other service models, such as the handling of the service as a collection of offering, a proper annotation facility, a model to describe the service concept, and the specification of added value. It handles those requirements at the same time.
- It helps capturing and organizing the discrete functions contained in (business) applications comprised of underlying business process or workflows into interoperable, (standards-based) services.
- The model accommodates the services to be abstracted from implementations representing natural fundamental building blocks that can synchronize the functional requirements and IT implementations perspective.
- It considers by definition that the services to be combined, evolved and/or reused quickly to meet business needs.
- Finally, it represents an abstraction level independent of underlying technology.

In addition, the W*H model in Fig. 1 also serves the following purposes:

- The inquiry through simple and structured questions according to the primary dimension on wherefore, whereof, wherewith, and worthiness further leading to secondary and additional questions along the concept, annotation, content, add value or surplus value space that covers usefulness, usage, and usability requirements in totality.
- The powerful inquiring questions are a product of the conceptual underpinning of W*H grounded within the conceptional modelling tradition in the Concept-Content-Annotation triptych extended with the Added Value dimension and further integration and extension with the inquiry system of Hermagoras of Temnos frames.
- The W*H model is comprise of 24 questions in total that cover the complete spectrum of questions addressing the service description; (W5 + W4 + W10H +W4) and H stands for how.
- The models compactness helps to validate domain knowledge during solution modelling discussions with the stakeholders with high demanding work schedules.
- The comprehensibility of the W*H model became the main contributor to the understanding of the domain's services and requirements.
- The model contributes as the primary input model leading to the IT-service systems projection on solution modelling.
- It contribute as the primary input model leading to the IT-service systems projection on the evaluations criteria of systems functioning on its trustworthiness, flexibility to change, and efficient manageability and maintainability.

3.2 Dimensions of Service Specification

The Content Dimension: Services as a Collection of Offerings. The service defines the what, how, and who on what basis of service innovation, design, and development, and helps mediate between customer or consumer needs and an organizations strategic intent. When extended above the generalized business and technological abstraction levels, the content of the service concept composes the need to serve the following purposes:

- Fundamental elements for developing applications;
- Organizing the discrete functions contained in (business) applications comprised of underlying business process or workflows into inter operable, (standards-based) services;
- Services abstracted from implementations representing natural fundamental building blocks that can synchronize the functional requirements and IT implementations perspective;
- Services to be combined, evolved and/or reused quickly to meet business needs; Represent an abstraction level independent of underlying technology.

The abstraction of the notion of a service system within an organizations strategic intent emphasized by those purposes given above allow us to define the content description of services as a collection of offers that are given by companies, by vendors, by people and by automatic software tools [3]. Thus the content of a service system is a collection of service offerings.

The service offering reflects the supporting means in terms of with what means the service's content is represented in the application domain. It corresponds to identification and specification of the problem within an application area. The problem is a specific application case that resides with an organizational unit. Those problems are subject to events that produce triggers needing attention. Those triggering events have an enormous importance for service descriptions. They couple to the solution at hand that is associated with how and what of a required IT solution.

The Annotation Dimension. According to [14], annotation with respect to arbitrary ontologies implies general purpose reasoning supported by the system. Their reasoning approaches suffer from high computational complexities. As a solution for dealing with high worst-case complexities the solution recommends a small size input data. Unfortunately, it is contradicting the impressibility of ontologies and define content as complex structured macro data. It is therefore, necessary to concentrate on the conceptualisation of content for a given context considering annotations with respect to organizations intentions, motivations, profiles and tasks, thus we need at the same time sophisticated annotation facilities far beyond ontologies. Annotation thus must link the stakeholders or parties involved and activities; the sources to the content and concept.

The Concept Dimension. *Conceptional modelling* aims at creation of an abstract representation of the situation under investigation, or more precisely, the way users think about it. Conceptual models enhance models with concepts that are commonly shared within a community or at least between the stakeholders involved in the modelling process.

According to the general definition of concept as given in [19], *Concepts* specify our knowledge what things are there and what properties things have.

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, e.g., lexical or ontological, logical, or reflective.

Conceptualisation aims at collection of concepts that are assumed to exist in some area of interest and the relationships that hold them together. It is thus an abstract, simplified view or description of the world that we wish to represent. 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.

The definition of the ends or purpose of the service is represented by the concept dimension. It is the curial part that governs the service's characterization. The purpose defines in which cases a service has a usefulness, usage, and usability. They define the potential and the capability of the service.

The Added Value Dimension. The added value of a service to a business user or stakeholder is in the definition of surplus value during the service execution. It defines the context in which the service systems exists, the story line associated within the context, which systems must coexist under which context definitions prevailing to time. Surplus value defines the worthiness of the service in terms of time and labor that provide the Return of Investment (ROI).

4 Conclusion

There are many other usage models for services. This paper elaborated the *explanation, informed selection, and appropriation* usage model for a service. Other usage models of an instrument as a model are, for instance, optimization-variation, validation-verification-testing, understanding, extension and evolution, reflection-optimization, exploration, documentation-visualization, integration, hypothetical investigation, and description-prescription usage models. We introduced in this paper a general notion of the model and showed what makes description or specification a service to be become a model of the service.

References

1. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: a method for developing service-oriented solutions. IBM Syst. J. **47**(3), 377–396 (2008)
2. Bergholtz, M., Andersson, B., Johannesson, P.: Abstraction, restriction, and co-creation: three perspectives on services. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010. LNCS, vol. 6413, pp. 107–116. Springer, Heidelberg (2010)
3. Dahanayake, A.: CAME: an environment for flexible information modeling. Ph.D. dissertation, Delft University of Technology, The Netherlands (1997)

4. Dahanayake, A., Thalheim, B.: Co-design of web information systems. In: Thalheim, B., Schewe, K.D., Prinz, A., Buchberger, B. (eds.) Correct Software in Web Applications and Web Services. Texts & Monographs in Symbolic Computation, pp. 145–176. Springer, Wien (2015)
5. Erl, T.: SOA: Principles of Service Design. Prentice-Hall, Englewood Cliffs (2007)
6. Glushko, R.J.: Seven Contexts for Service System Design. In: Maglio, P.P., Kieliszewski, C.A., Spohrer, J.C. (eds.) Handbook of Service Science. Service Science: Research and Innovations in the Service Economy, pp. 219–249. Springer Science+Business Media LLC, New York (2010). doi:10.1007/978-1-4419-1628-0_11
7. Goldstein, S.M., Johnston, R., Duffy, J.A., Rao, J.: The service concept: the missing link in service design research? J. Oper. Manage. **20**, 121–134 (2002)
8. Hurby, P.: Model-Driven Design of Software Applications with Business Patterns. Springer, Heidelberg (2006)
9. Maglio, P., Srinivasan, S., Kreulen, J., Spohrer, J.: Service Systems, Service Scientists, SSME, and Innovation. Commun. ACM **49**(7), 81–85 (2006)
10. Mahr, B.: Zum Verhältnis von Bild und Modell. In: Visuelle Modelle, pp. 17–40. Wilhelm Fink Verlag, Mnchen (2008)
11. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment. Acc. Rev. **57**, 554–578 (1982)
12. Poels, G.: The resource-service-system model for service science. In: Trujillo, J., Dobbie, G., Kangassalo, H., Hartmann, S., Kirchberg, M., Rossi, M., Reinhartz-Berger, I., Zimányi, E., Frasincar, F. (eds.) ER 2010. LNCS, vol. 6413, pp. 117–126. Springer, Heidelberg (2010)
13. Spohrer, J., Maglio, P.P., Bailey, J., Gruhl, D.: Steps towards a science of service systems. IEEE Comput. **40**, 71–77 (2007)
14. Schewe, K.-D., Thalheim, B.: Development of collaboration frameworks for web information systems. In: IJCAI 2007 (20th International Joint Conference on Artificial Intelligence, Section EMC 2007 (Evolutionary models of collaboration)), pp. 27–32, Hyderabad (2007)
15. 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)
16. Stojanovic, Z., Dahanayake, A.: Service - Oriented Software Systems Engineering: Challenges and Practices. Idea Group Publishing, Hershey (2004)
17. Thalheim, B.: Towards a theory of conceptual modelling. J. Univ. Comput. Sci. **16**(20), 3102–3137 (2010). http://www.jucs.org/jucs_16_20/towards_a_theory_of
18. Thalheim, B.: The conceptual model ≡ an adequate and dependable artifact enhanced by concepts. In: Information Modelling and Knowledge Bases, vol. XXV, Frontiers in Artificial Intelligence and Applications, vol. 260, pp. 241–254. IOS Press (2014)
19. Thalheim, B.: Models, to model, and modelling - towards a theory of conceptual models and modelling - towards a notion of the model. In: Collection of Recent Papers (2014). http://www.is.informatik.uni-kiel.de/~thalheim/indexkollektionen.htm
20. Thalheim, B., Nissen, I. (eds.): Wissenschaft und Kunst der Modellierung. De Gruyter, Ontos Verlag, Berlin (2015)
21. Vargo, S.L., Lusch, R.F.: Evolving to a new dominant logic for marketing. J. Mark. **68**, 1–17 (2004)
22. Vargo, S.L., Maglio, P.P., Akaka, M.A.: On value and value co-creation: a service systems and service logic perspective. Eur. Manage. J. **26**, 145–152 (2008)

Wherefore Models are Used and Accepted? The Model Functions as a Quality Instrument in Utilisation Scenarios

Marina Tropmann-Frick*, Bernhard Thalheim*

*Christian Albrechts University Kiel, Department of Computer Science, D-24098 Kiel, Germany
mtr | thalheim@is.informatik.uni-kiel.de

Abstract. Science and technology widely uses models in a variety of utilisation scenarios. Models function as a representation of origins in some of these utilization scenario, e.g. they function for explanation, for optimization-variation, for validation-verification-testing, for reflection-optimization, for exploration, for hypothetical investigation, for documentation-visualization, and finally for description-prescription as a mediator between a reality and an augmented reality that developers of a system intend to build. The functions of a model determine the *purposes*, the *goals*, and the *kind* of the model. The model effect as an instrument in these utilisation scenarios. Its qualities determine whether a model is acceptable, whether it is of high utility, why and for which reason a model is successfully used, and what are the convincing properties. These qualities can be derived from the utilisation scenario in which a model functions as an instrument.

1 The Mission of Models in Computer Science and Engineering

Computer science and engineering widely uses models. At the same time, the notion of the model is still not commonly agreed. There are many different application cases for models. Models are representations[1] of a collection of origins or originals. Origins can be material goods, systems, software, reality, augmented reality, imaginations of a person, etc. Following H. Stachowiak Stachowiak (1973, 1992), a model is often defined in a phenomenalistic way based on three properties:

(1) *Mapping* property: the model has an origin and can be based on a mapping from the origin to the instrument.

---

[1] Computer science and engineering uses the word 'artefact' for a representation. An artefact has beyond the meaning *"any object made by human beings, especially with a view to subsequent use"* also other meanings such as:
- any mass-produced, usually inexpensive object reflecting contemporary society or popular culture;
- a substance or structure not naturally present in the matter being observed but formed by artificial means;
- a spurious observation or result arising from preparatory or investigative procedures;
- a structure seen in tissue after death, fixation, staining, etc.

(2) *Truncation* (*reduction*) property: the model lacks some of the ascriptions made to the origin.
(3) *Pragmatic* property: the model use is only justified for particular model users, the tools of investigation, and the period of time.

We observe however that these properties do not qualify a representation as a model. The mapping and truncation properties are far too strict and need further investigation. We might use representations that are not images of mappings such as a Turing machine, a system architecture, or development strategies. Furthermore, we might use representations that are not reducts of origins such as (conceptual) information system models for the variety of viewpoints users of databases might have. So, we need a better definition of the notion of a model.

Models are developed by a community of practice for utilisation by a community of practice and in a context. The utilisation depends on the intentions of users and their context. So, we observe that the utilisation of models determines (a) the kind of model, (b) the governing purposes or goals of utilisation of the model, (c) the properties of a model, (d) the amplification a model provides with extensions, (e) the idealisation by scoping the model to the ideal state of affairs, (f) the divergence by deliberately diverging from reality in order to simplify salient properties of interest, and (g) the added value of a model. The seven additional statements are combined in the *mission* a model has. The mission clarifies how the model functions well within its intended scenarios of usage according to its capacity and potential. The mission must be coherent with the context, the determination or specific basis of conduct or utilisation of the model, and must be acceptable for the users or – more concrete – the community of practice. Therefore, the mission clarifies the functions (and anti-functions or forbidden ones), purposes and goals of the utilisation, the potential and the capacity of the model.

The mission is combined in the *cargo* Mahr (2008); Thalheim (2015); Thalheim and Nissen (2015b) with the *determination* of the representation (basic ideas, features, particularities, and the utilisation), an *abstract declaration of the meaning* (main semantic and pragmatic statements about the model; description of value of the representation according to its functions in the utilisation scenarios; its importance within the given settings), and a narrative explanation of the *identity* within the five kinds of identity: actual, communicated, accepted, ideal, desired identity.

The Storyline of the Paper. Questions that must be answered with any model are: Why, for what cause or for which reason, on what account a model is utilised? What is the intention underlying this utilisation? Why a model is acceptable within a certain scenario? What are the characteristics of a model that convince and persuade users to utilise this model?

In this paper, we start with reminding the conception of the model. Models are utilised as an instrument in some scenarios. These scenarios determine the functions the model has to play. Functions determine the purposes and goals we aim at. At the same time, any model has also limitations that restrict its utilisation. Whether an instrument can function as a model depends on its adequacy and dependability. Adequacy is well-considered in many publications. Dependability needs however clarification and deeper investigation. It combines the justification of utilisation of the given instrument and the sufficiency of the instrument in the given scenarios. Sufficiency can be based on quality characteristics and on evaluation procedures for these characteristics. We thus may derive maturity statements of a given model. We finally apply this quality evaluation to model used for description and prescription scenarios, e.g. information system models.

2 The Conception of the Model

A **model** *is a well-formed, adequate, and dependable instrument that represents origins.* Embley and Thalheim (2011); Thalheim (2014a,b)

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

The model should be well-formed according to some well-formedness criterion. An well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*.

Well-formedness enables an instrument to be *justified* by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through formulas, by falsifiability, and by stability and plasticity.

The instrument is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics Thalheim (2010) such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed instrument is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.

Figure 1 depicts dependability and adequacy properties of a representation that is used as an instrument in some utilisation scenarios. We explore therefore next models in their utilisation scenarios. Adequacy and justification has already been defined and considered in detail in Embley and Thalheim (2011); Thalheim (2010, 2014a); Thalheim and Dahanayake (2015); Thalheim and Tropmann-Frick (2015, 2016).

The model has a *background* consisting of an *undisputable grounding* from one side (e.g. paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities) and of a *disputable and adjustable basis* from other side (e.g. assumptions, concepts, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense). The background its often given only in an implicit form.

Models *function* as *instruments* or tools. Typically, instruments come in a variety of forms and fulfill many different functions. Instruments are partially independent or autonomous of the thing they operate on. Models are however special instruments. They are used with a specific intention within a utilisation scenario. The quality of a model becomes apparent in the context of this scenario.

Lightweight models typically cut off background and context. They assume per default some utilisation scenario and reduce the functions of the model to the main function. The purpose is then driven by this function. Often the community of practice is set to some standard community that uses a specific kind of justification. In this case, the sufficiency criteria are often related to well-formedness criteria Cherfi et al. (2002, 2007), e.g. syntactic ones. We notice that the mapping, truncation and pragmatic properties become simpler. We may extend this kind of scoping to *generic models* Thalheim et al. (2014) that are particular or idealised models for a specific community of practice with a specific background, within a specific context, and for representation of a specific world of origins under consideration. Generic models can be calibrated to specific models through a process of data or situation calibration, refinement, concretisation, context enhancement, or instantiation.

3 Models and Utilisation Scenarios

Models function as an instrument in some usage scenarios and a given usage spectrum. Their function in these scenarios is a combination of functions (explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualisation, description-prescription). The model functions effectively in some of the scenarios and less effectively in others. The function determines the *purpose* and the *objective* (or goal) of the model. Functioning of models is supported by methods. Such methods support tasks such as defining, constructing, exploring, communicating, understanding, replacing, substituting, documenting, negotiating, replacing, optimizing, validating, verifying, testing, reporting, and accounting. A model is *effective* if it can be deployed according to its objectives.

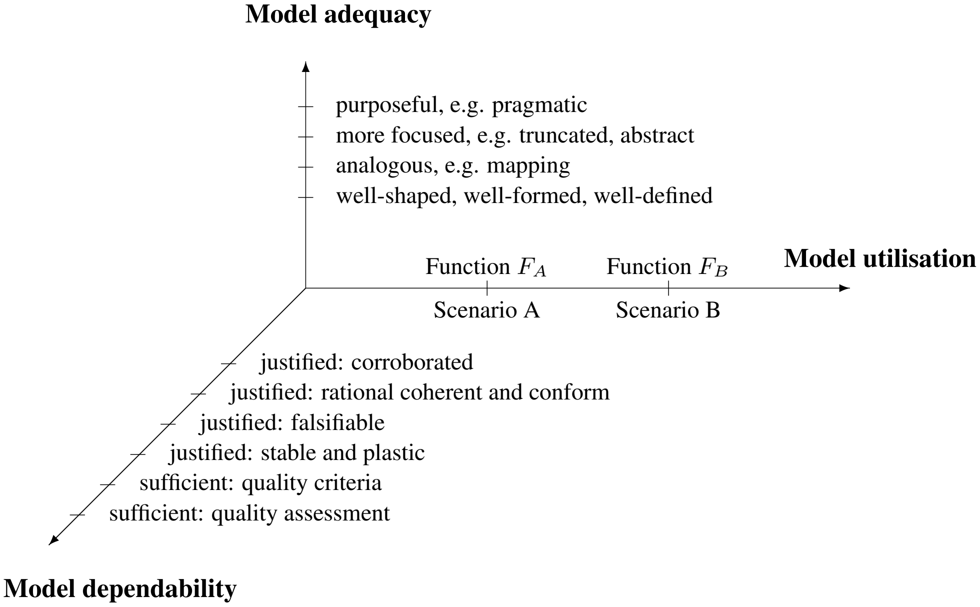


FIG. 1 – *The model as an instrument*
(1) that is used in utilisation scenarios in certain functions,
(2) that is adequate for representation of origins, and
(3) that is dependable for its utilisation.

Figure 1 present model configuration as a three-dimension characterisation based on utilisation and related adequacy and dependability properties. The specific form of characterisation varies a lot in dependence of the kind of model. Its mathematical rigidity can be rather Fuzzy or based on certain calculi (deductive, inductive, abductive, argumentation, etc.).

4 Functions and Kinds of Models

In general, a model is a representation of origins ('what') and in some cases a model for new origins Mahr (2015). It is used by users that form together with the modelers the community of practice ('who') within their context. The users have their specific 'stereotype' or pattern of using the model, i.e. the model has a function within these utilisation scenario ('where', 'whereat'). We can thus consider the origins, the community of practice, the utilisation, and the context as four governing directives for a model.

A model is thus an instrument that functions within utilisation scenarios ('Gebrauchsspiel' (deployment story), 'Sprachspiel' (language game, Wittgenstein (1958))) in different roles with different rigidity, modality and confidence ('how'). Each science and engineering discipline uses specific scenarios. The role and function of a model may vary. Typical functions of a model are: explanation, optimization-variation, validation-verification-testing, reflection-optimization, exploration, hypothetical investigation, documentation-visualization, and description-prescription. The last function uses models as a mediator between a reality and an augmented reality that developers of a system intend to build

The different functions of models are supported by different kinds of models. Models are in general used as *perception* models (reflection of one party's current understanding of world; for understanding the application domain), *situation* models (reflection of a given state of affairs), *conceptual* models (based on formal concepts and conceptions), *experimentation* models (as a guideline and basis for experimentation), *formal* models (based some formalism within a well-based formal language), *mathematical* models (in the language of mathematics), *computational* models (based on some (semi-)algorithm), *physical* models (as physical artifact), *visualisation* models (for representation using some visualisation), *representation* models (for representation of some other notion), *diagrammatic* models (using a specific language, e.g. UML), *exploration* models (for property discovery), and *heuristic* models (based on some Fuzzyness, probability, plausibility, correlation), etc. The large variety of notions for a model (e.g. see Thalheim and Nissen (2015a) for models used in science and engineering or Embley and Thalheim (2011); Thalheim (2014a,b) for conceptual models for database structuring) mainly reflects these different kinds.

5 Capacity, Potential, and Maturity of a Model

Capacity is a strategic measure whereas the potential is a tactical one. The potential can be used to derive the added value of a utilisation of a model within a given scenario. The potential allows to reason on the significance of a model within a given context, within a given community of practice, for a given set of origins, and within the intended profile.

The capacity relates an instrument to utilisation scenarios or the usage spectra. We answer the questions whether the instrument functions well and beneficial in those scenarios, whether it is well-developed for the given goals and purposes, whether it can be properly, more focused, comfortably, simpler and intelligible applied in those scenarios instead of the origins, and whether the instrument can be adapted to changes in the utilisation. The answers to these questions determine the main content or cargo, the comprehensiveness, and the authority or general value of a model. Another important aspect is the solution-faithfulness of the instrument. The capacity is an essential element of the model cargo, especially of the main content of the model.

Similar to SPICE assessments ISO/IEC (2006); Fiedler et al. (2009); Jaakkola et al. (2005), we may rate *maturation of a model* and a modelling approach to:

(0) A model is defined in an ad-hoc manner.

(1) A model is mainly defined in an informal form.

(2) Model development is systematic and managed and the model is of high quality.

(3) Models are based on standards and are well-understood.

(4) Models may be optimised, adapted and integrated with other models.

(5) Modelling follows a continuously improvable, supporting evolution and migration style.

The maturity ladder and modelling experience we have observed for three decades direct us to the hypothesis that most Computer Science modelling approaches have not yet reached level (2).

We use now the collection on the Kiel modelling approach Thalheim and Nissen (2015a) for an exploration of quality characteristics of models.

6 Sufficiency of Models: Quality Characteristics

Models have a function within their utilisation scenario. There are different utilisation scenarios such as construction, exploration, and explanation. According to Thalheim (2010), we may distinguish between

- *internal quality* characteristics (accuracy, suitability, interoperability, coherence, stability, generality, robustness, flexibility, self-contained, independence, minimality, language quality, compositionality, uniformity, changeability, documentation),
- *(development or) external quality* characteristics (correctness, pervasiveness, analysability, changeability, stability, testability, privacy of the models, ubiquity, expressiveness, generalisability, existence of refinement and abstraction hierarchies, traceability, adaptability, maturity, fault tolerance, recoverability, reliability compliance, configuratability, resource utilisation, scalability, testability, maintainability, stability, portability, reusability, replaceability), and
- *quality of use* characteristics (understandability, learnability, usefulness, comprehensibility, parsimony, operability, attractiveness, appropriatedness, availability, efficacy, efficiency, functionality, utility, usability, use, dependability, performance, fitness, productivity, safety, trust, satisfaction).

These quality characteristics are static. We may however also consider dynamic ones such as executability, refinement quality, scope restriction, effect preservation, context explicity, and completion tracking.

We may use the refined categorisation for internal characteristics in Fieber et al. (2008):

- The *inner quality* of a model is given by well-formedness characteristics (representation, syntactic well-formedness, semantic well-formedness, and pragmatic well-formedness,

modularity, controlled redundancy, clarity, style conventions, according to the rules), by syntactic characteristics (precision, syntactic simplicity, syntactic adequacy, level of detail), by semantic characteristics (universality, semantic simplicity, semantic adequacy, consistency, accuracy, degree of formalisation), and by pragmatic characteristics (conceptual integrity and uniformity, conformity, variety of representations, consistency with people or organisations).
- The *outer quality* relates the model to other models or to their origins: cohesion, correctness, completeness, traceability, changeability, validity, generality.

Another categorisation is given by Prat and Cherfi (2003); Cherfi et al. (2002); Akoka et al. (2007); Prat et al. (2014): The system dimension evaluates the goal, the environment, the structure, the activity and the evolution. It is supported by certain evaluation criteria. Software engineering uses metrics for evaluation. Another evaluation procedure is the one in Jaakkola and Thalheim (2010) where semantic and pragmatic calculi are used. The large variety of quality characteristics makes evaluation complex.

We prefer thus the approach used in Cherfi et al. (2007); Jaakkola and Thalheim (2010). A **quality evaluation stereotype** is a selection of quality characteristics that are driven by

- the functions of a model in utilisation scenarios that are considered, the resulting purposes and goals, and quality characteristics that are essential for the profile sufficiency,
- the kind of the model, and
- its capacity and potential.

Let us now develop one quality evaluation stereotype for the construction of systems.

7 The Description-Prescription Scenario for Structural Models of Information Systems

The construction scenario is one of the central modelling scenario for information and software system development. The model functions as a mediating specification for all viewpoints business users might have, as a blueprint for software development, and as an assessment artifact for the system developed accordingly.

Let us consider now quality characteristics for models that are used for development of information systems. We noticed already Thalheim and Tropmann-Frick (2015) that models are used (1) for communication and negotiation, (2) for conceptualisation, and for (3) realisation of a system. These models are different. Their primary quality characteristics vary as well.

We follow the design science approach Dahanayake and Thalheim (2011); Prat et al. (2014) and separate the construction process into three stages: relevance stage, modelling stage, and realisation stage. These stages may also be considered as the description phase followed by the prescription phase, i.e. first a system is described by a model and second the model is used as a prescription or blueprint for the realisation. This approach uses stepwise development of different artifacts according to Mahr (2009): first, the scope is set to a set of origins O; second, the relevant and necessary properties $\Phi(O)$ of O are elicited; third, these properties are mapped to objectives $\Psi(M)$ of the model M to be developed; fourth, the model is developed; fifth, we extract essential, relevant and necessary properties $\Phi(M)$ of M and map those to objectives $\Psi(Y)$ for the system Y under development. Finally, we may assess the system based on the properties $\Phi(Y)$ the system has.

We observe that support for modelling results in a wide variety of reasoning activities. For instance, reasoning about properties of a model is also based on an explicit consideration of the notion of an *analogy* between the model and the application domain origins or the model and its reflection in theories and constructions. Reasoning on objectives of realisations includes detection of requirements a system must satisfy. Figure 2 displays the different ways of working during an information systems development.

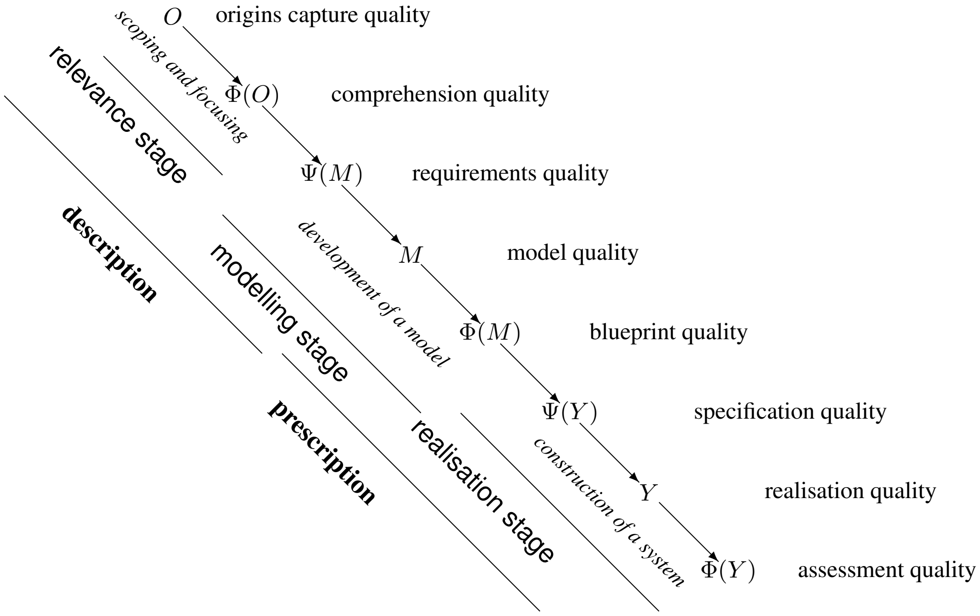


FIG. 2 – *Two of the three design science stages of information system modelling: description followed by prescription and corresponding kinds of quality characterisations*

The scenario uses models in three different functions during information and software systems development. These function are well-supported if O, $\Phi(O)$, $\Psi(M)$, M, $\Phi(M)$, $\Psi(Y)$, Y, and $\Phi(Y)$ obey the following properties:

Origins capture quality: Primary characteristics are: suitability, clarity, stability, plasticity, usability, and refinement & abstraction quality. We can also consider secondary characteristics such as: generality, controlled redundancy, usefullness, use, efficacy, and generality.

Comprehension quality: Primary characteristics are: adequacy, justification, generality, consistency, correctness, appropriatedness, fitness, and trust. We can also consider secondary characteristics such as: coherence, robustness, universality, and flexibility.

Requirements quality: Primary characteristics are: completeness, utility, accuracy, compositionality, understandability, and validity. We can also consider secondary characteristics such as: language quality, precision, generalisability, variety of representations, and clarity.

Model quality: Primary characteristics are: syntactic and semantic well-formedness, minimality, completeness, correctness, and usefulness. We can also consider secondary characteristics such as: modularity, uniformity, learnability, parsimony, simplicity, and degree of formalisation.

Blueprint quality: Primary characteristics are: accuracy, refinement & abstraction quality, correctness, and conformity. We can also consider secondary characteristics such as: coherence, robustness, and flexibility.

Specification quality: Primary characteristics are: adaptibility, completeness, and understandability. We can also consider secondary characteristics such as: well-formedness, syntactic characteristics, stability, flexibility, self-contained, uniformity, changeability and adaptability, and configuratability.

Realisation quality: Primary characteristics are: changeability, maintainability, dependability, configuratability, resource utilisation, and efficiency. We can also consider secondary characteristics such as: documentation, traceability, productivity, fault tolerance, reliability compliance, and scalability.

Assessment quality: Primary characteristics are: analysability, and testability. We can also consider secondary characteristics such as: configuratability, efficiacy, and trust.

The lists of quality characteristics are not complete. They show however that different characteristics are important during description and prescription. We observe that most of the quality characteristics must be semantically or syntactically defined Jaakkola and Thalheim (2010) and can only partially represented through metrics.

These quality characteristics are still too manyfold and become impractical. The lists become far shorter if we consider models in dependence on their main utilisation scenario. For instance, the description scenario uses models for communication and negotiation goals. The prescription scenario uses models according to the image, design and action, blueprint and realisation goals. The lists can also be calibrated according to the methodology that is used for information system development. The lists become condensed if the stages or their corresponding phases already use models, e.g. if high quality perception models and high quality situation models are already known at the phases of the relevance stage. A similar shaping can be developed for conceptualisation scenarios for information systems.

Let us now consider structure-driven information system development based on extended entity-relationship modelling language Thalheim (2000) that uses a prototyping methodology and a given concept space $\mathcal{C}$ in a well-known application area for straightforward description directly followed by prescription. This development uses a specific quality evaluation stereotype

$${}^{descr+prescr}_{\texttt{prototype}}\mathfrak{S}^{\mathrm{structureIS}}_{eER,\mathcal{C}}$$

which adornment is given by the scenarios, the methodology, the outcome, and the background of the model:

Driving quality concerns are the following ones:

- Functions of the model are quick communication (and negotiation) and prototypical system realisation. Resulting purposes are thus design and action, reasoning on consolidation, derivation of problematic elements, and realisation of a running system.

The goal is the development of at least one realisation as a proof-of-concept. The prototype may later be revised or used for proper development.

- The model is a mediating, guiding, and inspiriting instrument. It is the basis for prototypical realisation, for refinement of the approach taken, and for experimentation with the realisation. The model itself is informal or semi-formal. It incorporates visualisation features. In our case, it is tightly bundled with its diagrammatic sub-model in the extended entity-relationship modelling language.
- The capacity of the model describes the general properties of a model, e.g. its completeness, its non-functional and functional properties, its integrateability or interoperability, its advantages and disadvantages considering missing part, the development of appropriate functionality, provided reflections for user viewpoint, and its realisability of proper high-quality system. The potential characterises the performance of a model, e.g. the model capability, its fruitfulness, its restrictions and boundaries of the realisation, and corresponding justification considerations. The maturity of prototype-oriented models is typically on level 1 only.

Primary quality characteristics are controlled conceptual completeness, controlled conceptual correctness, syntactic correctness, syntactic completeness, flexibility, analysability, comprehensibility, potential operability, potential functionality, changeability, and application awareness through meaningful representation of the real world.

Secondary quality characteristics are conceptual minimality, coherence, traceability, configuratability, syntactic and semantic well-formedness, semantic adequacy, usefulness, efficacy, and use.

We observe that both primary and secondary quality characteristics are used at one stage but irrelevant almost all other stages. So, the list of quality characteristics becomes manageable for quality evaluation of O, $\Phi(O)$, $\Psi(M)$, M, $\Phi(M)$, $\Psi(Y)$, Y, and $\Phi(Y)$. For instance, assessment quality is given via analysability, potential functionality, changeability, configuratability, and efficacy. These five quality characteristics are partially derived from the quality characteristics for $\Psi(Y)$ and Y.

8 Principles of Model Development and Utilisation

We summarise our observations to principles of quality model design for description-prescription scenarios similar to Chen et al. (1999); Thalheim (2010):

Community principle: The model must support the entire community of practice and match to their understanding, their focus and scope, and their reasoning abilities in a non-disinterpretable form. The model can be used efficiently and comfortable and with a minimum of fatigue.

Scenario principle: The model accommodates a wide range of its application within its utilisation scenarios based on the specific utilisations of their users.

Adequacy principle: The model must be as focussed as only possible and must be understandable regardless of experience, knowledge, and skills of users that accept the same grounding and tolerate the basis. Unnecessary complexity is avoided. It is consistent with expectations and experience. It accommodates a wider range of bases.

Justification principle: All elements of the model are justified by a corroboration that relates them to origins, by coherence and conformity criteria, by an explicit statement on scope and focus, and by stability considerations against the potential set of origins.

Effectivity principle: The model delivers necessary elements effectively to their users, regardless of the user's skills and of ambient conditions. A model supports different representation models The model provides a clear line of sight of its elements for any user. It accommodates effective utilisation for its profile and provides variation features for similar functions.

Robustness principle: The model minimises hazards and the adverse consequences of accidental and unintended utilisations that are not supported by the profile of the model.

9 Concluding: The Model Functions as a Quality Instrument in Utilisation Scenarios

This paper is a contribution to a general theory of models and modelling in Computer Science. Typical deficiencies of modelling in Computer Science are: ad-hoc modelling, modelling in the small, limited reuse of models, models are not understood as some kind of programs, and rigid separation into sub-disciplines without development of a common understanding and culture. Models are considered to be the third dimension of science Thalheim and Nissen (2015a). Modelling is one of the four central paradigms of Computer Science beside structures (in the small and large), evolution or transformation (in the small and large), and collaboration (based on communication, cooperation, and coordination).

Models function in utilisation scenarios as instruments. As such models should be of high or at least sufficient quality. Many quality characteristics are known and partially of importance and relevance. Typically, qualities form a parameter space. Quality characteristics can be categorised into main or primary parameters and secondary, tertiary etc. ones in dependence on the function that a model has in a given utilisation scenario.

References

Akoka, J., L. Berti-Equille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdouí-Thion, Z. Kedad, S. Nugier, V. Peralta, and S. Sisaid-Cherfi (2007). A framework for quality evaluation in data integration systems. In *ICEIS (3)*, pp. 170–175.

Chen, P., J. Akoka, H. Kangassalo, and B. Thalheim (Eds.) (1999). *Conceptual Modeling, Current Issues and Future Directions, Selected Papers from the Symposium on Conceptual Modeling 1997*, Volume 1565 of *Lecture Notes in Computer Science*. Springer.

Cherfi, S. S.-S., J. Akoka, and I. Comyn-Wattiau (2002). Conceptual modeling quality - from EER to UML schemas evaluation. In *ER*, Volume 2503 of *Lecture Notes in Computer Science*, pp. 414–428. Springer.

Cherfi, S. S.-S., J. Akoka, and I. Comyn-Wattiau (2007). Perceived vs. measured quality of conceptual schemas: An experimental comparison. In *ER (Tutorials, Posters, Panels & Industrial Contributions)*, Volume 83 of *CRPIT*, pp. 185–190. Australian Computer Society.

Dahanayake, A. and B. Thalheim (2011). Enriching conceptual modelling practices through design science. In *BMMDS/EMMSAD*, Volume 81 of *Lecture Notes in Business Information Processing*, pp. 497–510. Springer.

Embley, D. and B. Thalheim (Eds.) (2011). *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*. Springer.

Fieber, F., M. Huhn, and B. Rumpe (2008). Modellqualität als Indikator für Softwarequalität: eine Taxonomie. *Informatik Spektrum 31*(5), 408–424.

Fiedler, G., H. Jaakkola, T. Mäkinen, B. Thalheim, and T. Varkoi (2009). Co-design of web information systems supported by SPICE. In *Information Modelling and Knowledge Bases*, Volume XX, Amsterdam, pp. 123–138. IOS Press.

ISO/IEC (2003-2006). Information technology - process assesment. parts 1-5. IS 15504.

Jaakkola, H., T. Mäkinen, B. Thalheim, and T. Varkoi (2005). Evolving the database co-design framework by SPICE. In *Proc. EJC'05, Informaton Modelling and Knowledge Bases Vol. XVII, Series Frontiers in Arificial Intelligence,*, Tallinn. IOS Press.

Jaakkola, H. and B. Thalheim (2010). A framework for high quality software design and development: A systematic approach. *IET Software*, 105–118.

Mahr, B. (2008). Cargo. Zum Verhältnis vo n Bild und Modell. In I. Reichle, S. Siegel, and A. Spelten (Eds.), *Die Wirklichkeit visueller Modelle*, pp. 17–40. München: Wilhelm Fink Verlag.

Mahr, B. (2009). Information science and the logic of models. *Software and System Modeling 8*(3), 365–383.

Mahr, B. (2015). Modelle und ihre Befragbarkeit - Grundlagen einer allgemeinen Modelltheorie. *Erwägen-Wissen-Ethik (EWE) Vol. 26, Issue 3*, 329–342.

Prat, N. and S. S.-S. Cherfi (2003). Multidimensional schemas quality assessment. In *CAiSE Workshops*, Volume 75 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Prat, N., I. Comyn-Wattiau, and J. Akoka (2014). Artifact evaluation in information systems design-science research - a holistic view. In *18th Pacific Asia Conference on Information Systems, PACIS 2014*, pp. 23.

Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Springer.

Stachowiak, H. (1992). Modell. In H. Seiffert and G. Radnitzky (Eds.), *Handlexikon zur Wissenschaftstheorie*, pp. 219–222. München: Deutscher Taschenbuch Verlag GmbH & Co. KG.

Thalheim, B. (2000). *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin.

Thalheim, B. (2010). Towards a theory of conceptual modelling. *Journal of Universal Computer Science 16*(20), 3102–3137. `http://www.jucs.org/jucs_16_20/towards_a_theory_of`.

Thalheim, B. (2014a). The conceptual model ≡ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, Volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pp. 241–254. IOS Press.

Thalheim, B. (2014b). Models, to model, and modelling - Towards a theory of conceptual models and modelling - Towards a notion of the model. Collection of recent papers, http://www.is.informatik.uni-kiel.de/∼thalheim/indexkollektionen.htm.

Thalheim, B. (2015). Das Modell des Modelles. *Erwägen-Wissen-Ethik EWE-Heft, Heft 3, 2015, 26. Jg.*, 407–409.

Thalheim, B. and A. Dahanayake (2015). A conceptual model for services. In *Invited Keynote, CMS 2015, ER 2015 workshop*, LNCS 9382, Berlin, pp. 51–61. Springer.

Thalheim, B. and I. Nissen (Eds.) (2015a). *Wissenschaft und Kunst der Modellierung*. Berlin: De Gruyter, Ontos Verlag.

Thalheim, B. and I. Nissen (2015b). *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, Chapter Ein neuer Modellbegriff, pp. 491–548. Boston: De Gryuter.

Thalheim, B. and M. Tropmann-Frick (2015). The conception of the conceptual database model. In *ER 2015*, LNCS 9381, Berlin, pp. 603–611. Springer.

Thalheim, B. and M. Tropmann-Frick (2016). Models and their capability. In C. Beierle, G. Brewka, and M. Thimm (Eds.), *Computational Models of Rationality*, Volume 29 of *College Publications Series*, pp. 34–56. College Publications.

Thalheim, B., M. Tropmann-Frick, and T. Ziebermayr (2014). Application of generic workflows for disaster management. In *Information Modelling and Knowledge Bases*, Volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pp. 64–81. IOS Press.

Wittgenstein, L. (1958). *Philosophical Investigations*. Oxford: Basil Blackwell.

Résumé

Models as adequate and dependable representations of origins. They function in utilisation scenarios and are thus instruments. Their dependability can be given on the basis of a justification and evaluation for some main quality characteristics which are selected according to the functions a model has in a given utilisation scenario. The quality characterisation can be defined as a manyfold of quality parameters in which some are main and others are not relevant or distinctive. Which quality characteristics is considered to be primary is determined by the utilisation scenarios and the functions the model has.

Model-Based Engineering for Database System Development

Bernhard Thalheim

Christian Albrechts University Kiel, Department of Computer Science
Olshausenstr. 40, D-24098 Kiel, Germany

thalheim@is.informatik.uni-kiel.de

Dedication and tribute to Antoni Olivé

Abstract. A model functions in a utilisation scenario as an instrument. It is well-formed, adequate and dependable. It represents or deputes origins. This conception of the the model is a very general one.
Computer engineering uses models for description of development intentions and for prescription of the system to be build. It typically uses a number of models depending on the layer of abstraction, the scope, the context, the community of practice, and the artefacts to be represented. Model-based development is one of key success factors for development of database systems. This paper thus develops foundations for model-based engineering. Database system development is used as the illustration example for this investigation.

1 Models in Computer Science and Computer Engineering

Models are a kernel element of Computer Science and Computer Engineering (CS&CE). They are used sometimes without any definition or with an intuitive understanding. We know, however, a large variety of model notions (e.g. the 46 notions in [46]). A general theory, technology, art, science, and culture of modelling remain to be one of the research lacunas.

1.1 The Model

A **model** *is a well-formed, adequate, and dependable instrument that represents origins.* [11, 44, 45]

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

The model should be well-formed according to some well-formedness criterion. As an instrument or more specifically an artifact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form.

Conceptual Modeling Perspectives, 137--153, Cham, 2017, Springer

1.2 Multi-Model Modelling

Most sciences use coexisting models as a coherent holistic representation of their understanding, their perception, and their theories. For instance, medical research [8] typically considers medical models as experimentum, practicale, ratio, speculativum, and theoreticalis. These models are developed with different scale, precision, variability, vision, veracity, views, viewpoints, volume, and variation.

Each of the models has some functions in utilisation scenarios, for instance, communication, negotiation, construction, and representation and depiction functions. Depending on these functions, the model may be considered to be adequate and dependable. If we use several models then coherence of these models becomes an issue. We may explicitly represent coherence of models through model suites [7, 43]. We may also layer models based on their abstraction and scale, e.g. [17]. UML [32] uses ensembles of models that are loosely coupled.

A *model suite* consists of a set of models, an explicit association or collaboration schema among the models, controllers that maintain consistency or coherence of the model suite, application schemata for explicit maintenance and evolution of the model suite, and tracers for the establishment of the coherence.

A specific model suite is used for co-design of information systems that is based on models for structuring, for functionality, for interactivity, and for distribution [39]. This model suite uses the structure model as the lead model for functionality specification. Views are based on both models. They are one kernel element for interactivity specification. Distribution models are additionally based on collaboration models, e.g. [38]. Co-design includes coherence and maintenance of coexistence and co-evolution. Models might be complementary or completing or reversing or opposing each other, e.g. static and dynamic models in the HERM and BPMN languages. At the same time, a model suite integrates models and thus forms a new and more complex model which may convey totally different meanings. Models share than purposes, responsibilities, and meanings. A model suite may consist of two or more models (bi-models or di(ptych)-models, triptych-models etc.). The association among models in a model suite is based on association styles and patterns such as master-slave, proxy, or publish-subscribe. Since models can exclusively serve some purpose the remaining models may be latent or inactive or of non-interest as long as the given purpose is of interest.

A specific model suite consists of two models which share most of their background, context, community of practice, their application scenario, and thus also function within these scenarios. These models coexist together, are interdependent, and are correlated to each other. We call such models *co-model.* Co-models form a diptyph[1].

They can be coalesced into one model with two different sub-models or they may depend from each other (see, for instance, the Königsberg bridge models in [28] with the topographical, topological and graph-theoretic models). Origins are often also models and thus form together with their model a co-model. The

---

[1] A diptych is work made of of two parts. So, we might call co-models also *di-models* or *diptych models.*

orgin M_1 thus conditions its model M_2, i.e. M_2/M_1 is a *conditional*. Modern CS&CE is full of examples of such co-models, e.g. [2, 9, 10, 13, 16, 25, 26, 31, 32, 34, 36, 41]. A model in a co-model also often inherits adequacy and dependability of the other model. Sometimes, they follow however also different backgrounds. For instance, eER-based conceptual modelling uses a global-as-design paradigm. BPMN-based conceptual models are based on a local-as-design approach with an orientation of actors with their roles.

1.3 Science and Engineering

Science and engineering are two rather different activities. According to the Encyclopedia Britannica [35], science is (1) the state of knowing, (2a) a department of systematized knowledge as an object of study, (2b) something (as a sport or technique) that may be studied or learned like systematized knowledge, (3a) knowledge or a system of knowledge covering general truths or the operation of general laws especially as obtained and tested through scientific method, (3b) such knowledge or such a system of knowledge concerned with the physical world and its phenomena alike in natural sciences, and (4) a system or method reconciling practical ends with scientific laws.

Engineering is nowadays performed in a systematic and well-understood form [1]. It also well supported in software engineering, e.g. CMM or SPICE [18]. Engineering is the art of building with completely different success criteria (see [37]: “Scientists look at things that are and ask ‘why’; engineers dream of things that never were and ask ‘why not’.” (Theodore von Karman) “Engineers use materials, whose properties they do not properly understand, to form them into shapes, whose geometries they cannot properly analyse, to resist forces they cannot properly assess, in such a way that the public at large has no reason to suspect the extent of their ignorance.” (John Ure 1998)).

S. Oudrhiri [33] considers four elements of matured engineering: “(a) the technological know-how, (b) a set of established practices, (c) a scientific approach for defining the underlying principles of these practices, and (d) an economical model to explain the implications of such practices in terms of value delivered (effectiveness) and resources consumed (efficiency)”. Engineering is inherently concerned with failures of construction, with incompleteness both in specification and in coverage of the application domain, with compromises for all quality dimensions, and with problems of technologies currently at hand. [48] distinguishes eight stages of engineering: inquire, investigate, vision, analyse, qualify, plan, apply, and report.

1.4 Co-Models and Model Suites in CS&CE

CS&CE often uses direct associations of models, i.e. a model is based on another model. Modelling is then concerned with two models at the same time. For instance, completed database structure modelling starts with a situation model that is represented by a perception model. This perception model is the basis for the business model which is again the basis for the conceptual model. The conceptual model is mapped to a logical model according to the platform for realisation of the database system. The logical model is then mapped to the physical model. This pairwise modelling is based on a dichotomy of the models.

This dichotomy is used for closing the gap between the user world (where the information system is a social system) and the IT world (where the information system is a technical system). Figure 1 displays the mediating functions of typical information system models. Classical development methodologies are often based on consideration of two models. For instance, structure modelling might start with a business data model that is intentionally based on the perception model within the user world. This business data model is refined to a conceptual structure model in a conceptual modelling language such as ER [42]. The conceptual structure model is enhanced, transformed or compiled to a logical data model. If we follow a systematic approach then the logical data model is refined, enhanced and transformed to a physical data model.

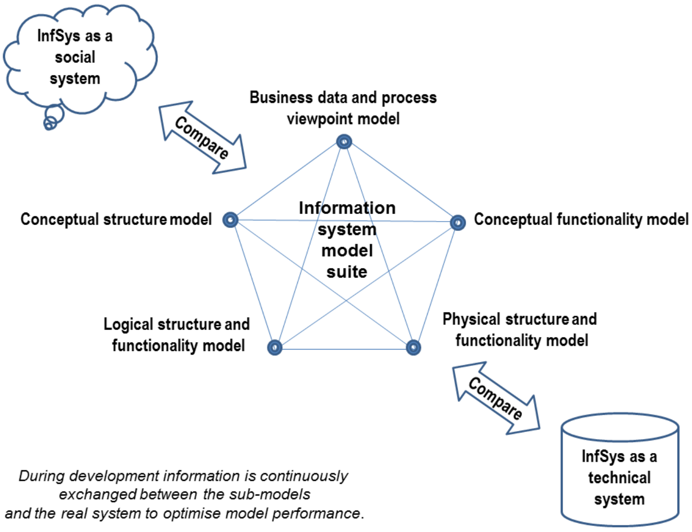


Fig. 1. Model suites: The five main models that comprise a complete model of a database system

1.5 Model-Based Engineering as Specific Model-Based Reasoning

Model-based reasoning is reasoning with the aid of models, reasoning about models in their own right, and reasoning that is model-determined [27, 30]. Models have then three different functions depending on these reasoning scenarios [6, 46]: models are instruments for reasoning which implies their prior construction and the reasoning necessary for their construction; models as targets of reasoning;

models as a unique subject of reasoning and its preliminary. Abduction has been considered the main vehicle of model-based reasoning. In CS&CE, reasoning is also based on explicit consideration of adequacy and dependability of models within the description/prescription scenario. From one side, models are used as a representation of some thought or better some mental models (e.g. *perception models*) which are representing the (augmented) reality (i.e. the perceived *situation model* and the objectives for system construction). From the other side models are used as blueprints for realisation of intentions by software systems. In the last case, models are also *documentation models* for the software system, at least at the first completion of the system.

Model-based engineering has been considered for a long time as '*greenfield*' development starting from scratch with a new development. Engineering is however nowadays often starting with legacy systems that must be modernised, extended, tuned, improved etc. This kind of '*brownfield*' development may be based on models for the legacy systems and migration strategies [22]. Again, we observe a co-model approach with a legacy model for revision, redevelopment, modernisation and migration and a target model for development of the new modernised and extended system. So, the legacy model (or legacy models) is associated with a sub-model of the target system.

1.6 The Objectives of the Paper

Model-based engineering attracts a lot of research, e.g. [3, 21, 23, 40]. Model-driven software development (MDSD) distinguishes enterprise, platform independent, platform specific, and code models. MDSD on the basis of model suites and with a direct consideration of model properties has not yet been investigated. So, we start with a case study in Section 2. This case study is used for derivation of principles in Section 4. Finally, Section 5 discusses the role of conceptual models in model-based engineering of database system development.

Due to space limitations, the paper cannot discuss in detail techniques that are necessary for systematic model-based engineering. Many techniques are already developed for specific modelling languages, for specific application domains, and for specific development approaches. A systematic generalisation and harmonisation of these techniques is still a research task. We illustrate the approach based on entity-relationship modelling (ERM) languages, on data-intensive applications and ERM-based development. The paper aims thus in a methodological background for model-based engineering. We restrict the paper to co-models and their specific style for model-based engineering.

2 A Case Study for Structure-Representing Co-Models

Let us consider two cases of co-models. It is often claimed that the ER modelling language can be used at the business and the conceptual layer in a similar form. If we look a bit more into the details then we discover essential differences that must be taken into consideration. For instance, we might have models that cannot be mapped to models at the lower layer or models that cannot be represented at the higher layer. At the same time, we might have many choices for lower layer models (Figure 2). Moreover, data models at one layer might not be entirely represented by data models at the other layer. For instance, cardinality constraints might not be representable by classical relational constraints. We must either enhance the relational language or represent constraints by procedural features of the relational database platforms.

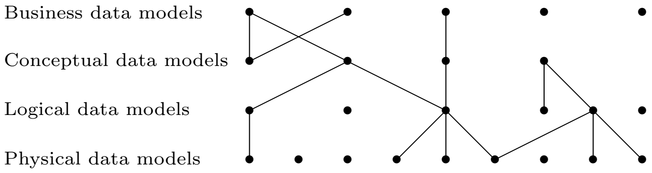


Fig. 2. Association of models in multi-layer modelling

2.1 Co-Models: Business Data Models and Conceptual Models

Business data models reflect the way how business users consider their data. Each business user considers only specific data within a specific viewpoint. A business application provides some kind of collaboration or exchange mechanism for these data.

The origins that are reflected in business data models are the situation model of a given application area and a collection of perception models that reflect specific viewpoints of business users. The understanding of data by business users is based on the way of work at business. So, data models represent their rather specific understanding of the application domain. These data models follow a local-as-design representation style. Conceptual models follow however a global-as-design approach [47], i.e. the model consists (i) of a global schema that harmonises and integrates the variety of viewpoints and (ii) of generalised (external) views that are derivable from the form the global schema and represent the local viewpoints. The two kinds of models - the business data models and the conceptual model - are tightly associated by an explicit infomorphism (i.e. generalised di-homomorphisms, see below). Adequateness and dependability of the conceptual model is derived from this association. Additionally, well-formedness of the conceptual model is based on the language, e.g. an extended entity-relationship (eER) modelling language, e.g. HERM [42].

Business (layer) data models and conceptual (layer) data models are a typical example of a *vertical model suite* since the first one is typically more abstract and the second one can be considered to be a refinement of the first one. The binding among these models is often implicit. We may however enhance the two models by a mapping that maps the first model to the second one. This mapping combines and harmonises the different views that are used at the business user layer.

2.2 Co-Models: Conceptual Models and Logical Models

Logical models are based on the same underlying semantics for the modelling language, e.g. set semantics. Physical data models typically use multi-set semantics (also called bag semantics) for (object-)relational database management systems. Logical models may follow object-relational approaches or purely relational approaches. eER conceptual models have an implicit semantics beside the explicit semantics. For instance, relationship types obey an inclusion and an existence constraint that restricts existence of relationship objects by existence of their referred component objects – in most cases entity objects.

Conceptual views are represented by a collection of object-relational views. We have a number of potential associations between conceptual and logical models. Which one is appropriate depends on choices for structuring, for reorganisation or optimisation or normalisation, for treatment of constraints, for handling of missing values, for controlled redundancy, for treatment of hierarchies, for naming, etc. Additionally, specific platform-oriented features are integrated into the logical model. The transformation follows rules and uses specific decisions.

So, the conceptual and the logical models are co-models that follow a refinement approach [49] (1) by injecting specific styles, tactics, embeddings, and language pattern to the logical model [1] and (2) by rules for transformation, extension, enhancement, and specialisation applicable to the logical model [12]. So, a conceptual model is typically associated to many logical models depending on the style of chosen refinement. We may consider an abstract description of the refinement approach as *pragmas* which are already given together with the conceptual model. The refinement may also result in an *information loss*. For instance, the view schemata defined for the conceptual model are mapped to a collection of relational views. The interrelation among the relational views is however not maintained in an explicit form.

Conceptual (layer) data models and logical (layer) data models also an example of a *vertical model suite* with a straightforward mapping from the conceptual layer to the logical layer.

2.3 Co-Models: Conceptual Co-Design of Structuring and Functionality

Database design and development typically is based on two models for structuring and functionality. The structure model is the ‘lead’ model for functionality since it defines the signature of the basic terms. The structure model imposes however also restrictions to the functions due to the integrity constraint enforcement and maintenance. Functionality is specified as a set of create-retrieve-update-delete functions. The data modification functions can be extended for preservation of integrity. The retrieval functions are defined based on a number of retrieval pattern and as algebraic expressions, e.g. HERM+ [42]. So, the lead model is some kind of 'order' model and the functionality model is partially 'enslaved' [15].

Structure models and functionality models form a *horizontal model suite.* Their association is based on an infomaorphism (see the similar vertical case in Section 4.1). All elements of the models are associated in a bipartite graph. The edges in the graph may be enhanced by existence dependencies, e.g. an operation or query uses the structural notions which are defined in the structural model. The control of such dependencies may be defined in a form similar to referential integrity.

2.4 Lessons Learned for Model-Based Engineering

A modelling language has its own obstinacy. It injects its background, its limitations and its treatment of semiotics into the model. Therefore, model-based engineering must explicitly represent these language specifics. Whenever models are used within a model suite, the association of models is language-biased and language-limited. Next, models are also driven by the directives, i.e. the artifacts to be represented, the profile of the model that is intended, the community of practice that might accept the model, and the context into which the model is set. Furthermore, the capacity and potential of the model itself restricts applicability. From the other side, we may restrict engineering to some kind of 'best' effective and efficient model. Finally, the classical approach to arbitrarily enhance a lower layer model limits the usefulness of the higher-layer model.

We may now consider either co-models at the same layer of abstraction ("*horizontal co-models*") or at different layers of abstraction ("*vertical co-models*"). Database structure development is typically based both on vertical co-models that are on adjoining layers and on horizontal co-models in the co-design case.

3 The First Principle of Modelling

3.1 Logoi of Modelling

Modelling results in a model as a *surface structure* and is in reality combined with a *deep structure* that is based on the background and the directives of the model. The deep structure of a model is represented by the *modelling logos*[2] [5, 24] that is the rationale or first principle behind modelling.

The model has its *background* $\mathfrak{B}$ consisting of an undisputable grounding from one side (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities) and of a disputable and adjustable basis from other side (assumptions, concepts, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense) which represent

---

[2] In the Faust poem by J.W. Goethe, Faust reasons in the study room scene on the meaning of the word 'logos' $\lambda\acute{o}\gamma o\varsigma$. This word has at least 6 meanings where Faust used only four of them: word, *concept, judgement,* mind, power, deed, and *reason.*

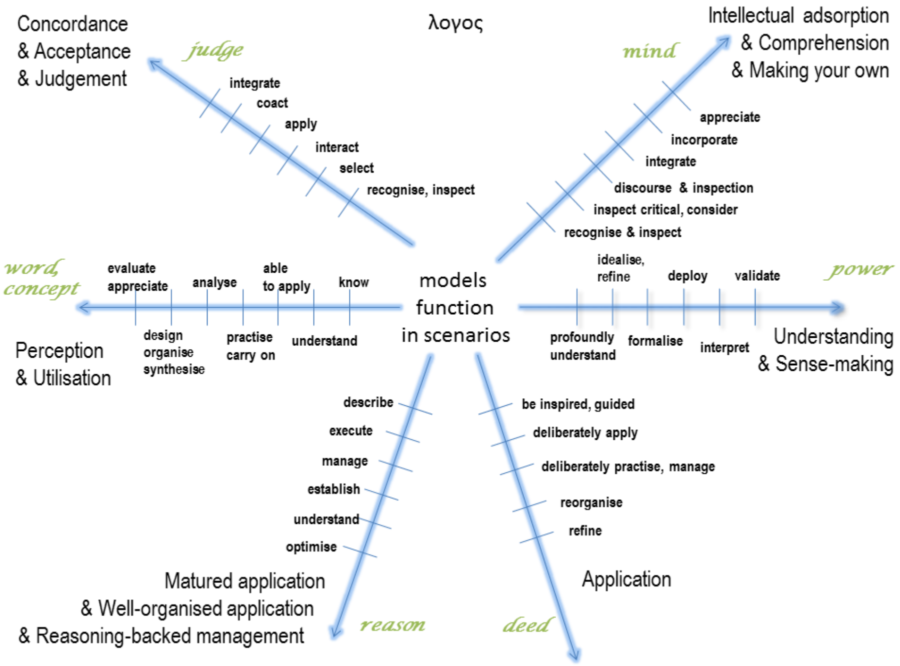


Fig. 3. Utilisation scenarios for models and stages of their deployment

the nature of things themselves. The background provides the deep structure of the model by explanations, analysis and manifestation. It is governed by its inner directives (origins/artefact to represented $\mathfrak{O}$, profile of the models (goal, purpose, function)) $\mathfrak{P}$ and the outer directives (community of practice, and the context) $\mathfrak{D}$. It is based on a language $\mathfrak{L}$ with its general notion, capacity and potential. Model development is based on actions $\mathfrak{A}$ and modelling and utilisation methods with their rational choice, i.e. the rationality expressed in the model as code, in interpretation and in action.

The ***modelling logos***[3] consists of the background, the outer directives, the language, and actions. The modelling logos is expected to understand before model development and utilisation. The logos thus determines the modelling notions of trueness, verifiability, rationality, and correctness. Parameters for models themselves are the inner directives. We claim that models cannot be understood without understanding the modelling logos.

3.2 Scenarios and Resulting Functions of Models

These different meanings of the Greek word logos are used in different utilisation scenarios. Concept and conceptions are the basis for the perception and utilisation scenario. A *conceptual model* is a model that incorporates concepts and conceptions. Models might be accepted in a community of practice based on judgements of members of a community of practice [19]. Models may be acceptable for this community and be thus intellectually absorbed. Models then gain an expressive power and make sense within an application. Models can also be used and applied in a development process. This application may also use methods of matured development. The last one is based on model-based reasoning which can be guided by maturity approaches, e.g. CMM and SPICE. So, we observe a number of scenarios which are depicted in Figure 3.

Models *function* as ***instruments*** in these scenarios at various stages of maturity. For instance, the application scenario may use models as an inspiration for further development. This stage is often observed for UML-backed programming. Instead, models may be deliberately applied or managed. They may be used as co-models and thus co-evolve together with the realisation, i.e. they become reorganised during utilisation. This reorganisation may also based on systematic approaches and thus be based on a refinement strategy.

4 Engineering for Vertical Co-Models

4.1 Database Development with Vertical Co-Models

Vertical co-models are widely used in CS&CE. The methodologies developed so far do however not consider the nature of multi-models. The case studies in Section 2 showed the influence of the *background-ladeness* of models. It is not

---

[3] The logos combines specification of the language, the knowledge behind, the reality under consideration, and the actions. [24]

easy to switch from a local-as-design paradigm to a global-as-design paradigms. Models are also *directives-laden*, especially with the outer directives community-of-practice and context. It is simpler if data are of the same granularity, scale and scope. For this reason conceptual models use an approach to represent data at their lowest scale and smallest granularity. Scientific databases (and also industrial databases, e.g. [20]) often start with raw data and consider them as the basis of all derived, purged, combined, and analysed data. They fail whenever size of databases matters.

The association between co-models can be based on the notion of the *infomorphism*. We extend the notion in [22] for models as follows. Two models M_1, M_2 are E_1, E_2-*infomorph* though two transformations E_1, E_2 with $E_1(M_1) = M_2$ and $E_2(M_2) = M_1$ if any model object o defined on M_i can be mapped via E_i to objects defined on M_j for $i, j \in \{1, 2\}$, $i \neq j$.

We notice that this notion allows to associate models with different granularity, models that incorporate views defined on top of a global schema, model suites within the local-as-design style that have a latent association model underneath, and co-evolution of models within a model suite. It can also be extended to model refinement similarly to [49]. We may use the infomorphism also for justification of one model by another model similar to the associations discussed in Subsections 2.1 and 2.2.

4.2 Model-Based Engineering with Co-Models

Model-based engineering is turning an idea into a reality on the basis of models. Models are used as the tacit knowledge for engineering through conception, feasibility, design, manufacture and construction. They reduce complexity while at the same time providing means for sustainable development and for coping with the interdependencies between systems - technical ones as well as social ones, at different layers at the same time.

Engineering of information systems still needs a lot of research, theories, skills and practices. System development becomes nowadays based on iterative development. The time of one-way models is over. Models are becoming reused, reconfigured, continuously evolving and integrated. So, the five plus two models in Figure 1 must co-evolve. Modern CS& CE is not anymore concentrated on a singleton development but has to look outwards, to handle the ‘big picture’, to think and to reflect during practising, to manage complexity and risks at the same time in an economic form.

The details of sub-systems are beyond common sense. We must rely on instruments as an abstract source of understanding and managing. One central instrument are *models* for the system world, for a system, for sub-systems, for embedded systems, and for collaboration of systems. Models allow us to understand what we want, what we think to know and to manage, how we make achieve what we want, what actually to do, and finally what we think might be the consequences. Since engineering is also a business activity, engineering activities must be affordable and financially predictable. Models provide a practical commonsense view that helps us to manage professionally and at acceptable risks. So, *model-based engineering* is one of the main issues of modern CS&CE. It goes far beyond model-driven development and model-driven architectures.

Therefore, we need *first-class models* and a technology to *handle models in a holistic manner*. One approach to master development is *layering*, i.e. coherently deploying various models of social systems and various models for technical systems. We develop this approach on the basis of business/conceptual and conceptual/logical co-models. In a similar form co-design of structuring and functionality may be managed and mastered.

So far we considered the modelling logo as a description logo. We may also consider the other model suite logos such as *control, application, organisation, economics, and evolution logos* for controllers, application, and tracers within model suites. Let us now sketch the controller and application ingredients for model-based engineering with co-models.
The *model suite association style* is based on general schemata for supporting programs (sub-model pattern for release, sharing, and access including scheduling of access), style of association (peer-to-peer , component, push-event, etc.), and on coordination activities describing the interplay among models. The control might be based on lazy or eager control styles.
The *association pattern* among models can be based on wrapping, componentisation, interception, extension or model models. The application processing can be active, proactive, synchronising or obligation-oriented. Synchronisation may use a variety of pattern. Whether association is based on parallel execution depends on the style of the association.
The *model suite architecture* describes inner association among models or submodels and is given by a general network with pairwise or n-ary bindings among these models.
The *model suite exchange* is based on constraints, their enforcement and the handling mechanisms for associations among models and sub-models. They might include also obligations for maintenance of changes within a model suite.
The main issue behind this approach is to deeply understand how these models can coexist, co-evolve, influence and restrict each other, and support or hinder the other. So, we first develop an insight into the deployment and especially the modelling logo of such model suites for a co-model example.

5 Conceptual Models as Mediators Within a Model Suite

The conceptual model is often used as a medium and mediator [29]. “Models function not just as a means but also as a means of representation” [14] with a deep background such as starting points and questions, knowledge, theories, actual hypotheses, tacit knowledge in tools, goals and objectives, tools, data generation, data on hand, data processing, and data interpretation [4]. Mediating models are retrospective and prospective at the same time and ravish. Beside mediation, other and different models can also be developed for documentation, communication, negotiation, orientation, inspiration, etc.

5.1 The Dichotomy of Description and Prescription for eER Models

The main function of eER models is its utilisation during database structure construction. The model consists of a schema, a number of views, and the realisation style [39, 47]. It is descriptive and prescriptive. The descriptive part reflects the business user models and thus uses an explicit association by views. The prescriptive part can be based on realisation templates. Adequacy is given due to the association to the business models, due to the objectives of description and prescription, due to the explicit restriction to the model focus, and due to the realisation context. Dependability is based on the association to the business user model, on the objectives of co-design, and on the capacity and potential of logical modelling languages that we intend to use. So, the model reflects two rather different origins, the business model and the logical model.

5.2 Some Modelling Logos of ER Modelling

Modelling logos of (extended) entity-relationship modelling languages are hidden within the language and not explicitly discussed in the ER literature. They are partially reflected in literature that introduce other languages. They should however be known whenever ER modelling is performed.

The background is reflected by (for details see [42]): In the *Global-As-Design* approach, the schema reflects all viewpoints. Local viewpoints are derivable and somehow reflectable. *Explicit existence existence* postulates that any object must exist before there can be a reference to it, i.e. rigid separation of creation and use. The model assumes a *closed-world view* and *unique names.* It is based on a well understood *name space* or glossary or ontology. *Salami-slice* representation uses homogenous, decomposed types (potentially with complex attributes) with incremental type construction. *Functionality representation* is deferred without consideration of the performance impact to the schema. *Separation into syntax and semantics* allow to define semantics on top of the syntax. Explicit semantics is based on constraints. *Paradigms, postulates, assumptions* of database technology and database support are assumed due to the three main quality criteria (performance, performance, performance). *Basic data types* are hidden with some mapping facilities to DBMS typing systems. *Visualisation* is represented by one holistic diagram that displays the entire syntax and semantics.

Outer directives are (for details see [42]): The *context* is entirely determined by DBMS technology of the last decades and heavily restricted by the platform and the systems that should be used. Data must become identifiable. The *population* is *finite* what causes problems with cyclic constraints, e.g. locally defined cardinality constraints are then global constraints. The community of practice consists mainly of DBMS professionals, modellers and may be business deciders. The first two groups are used to and biased by the paradigms, postulates, assumptions, etc. of DB technology.

The potential and capacity of the ER modelling language is restricted by the flatness of the schema definition. Schema construction may be guided by style guides and well-formedness characteristics. *Construction* of schemata is entirely *hierarchical* (or incremental or inductive) and follows approaches known for (hierarchical) first-order predicate logics. Construction is restricted to *3 or 4* or more *constructors* (entity, attribute, relationship types; additionally cluster types). Schema semantics is *canonically* defined. *Hidden set semantics* is used with *implicit* pointer semantics for relationship and cluster types. *Generalisation and specialisation* of all kinds are reflected through specific subtype or grouping (clustering) constructs. The manifold of specialisations is separated. Semantics is *static*. All schema elements are *completely defined*. Explicit semantics is defined through constraints which might however require *treatment beyond* (canonical) first order predicate logics. *Viewpoints* are defined through views on top of the schema definition via algebraic expressions. *Derived attributes* are defined via algebraic expressions. *Algebra* is restricted to terms that can be constructed for the algebra operations. Expressions may be generically defined with structures as parameters, e.g. insert(type) as generic operation.

Classical development methods are based on the kind of ER schema and view construction. They include methods for stepwise incremental construction, extension, decomposition, design, validation, and evaluation (see [42]). We may use a number of methodologies, such as top-down, bottom-up, modula, inside-out, and mixed. Classical utilisation actions and resulting methods are mapping and transformation methods (see [42]). Methods for integration, calibration, verification, control, reconfiguration, migration, and evolution are still under investigation.

The profile is restricted to the system construction function for mediating models.

6 Concluding: Models and Model-Based Engineering

Model-driven engineering and development has become an area of intensive research. Roles, limitations, background and directives of the model have however not been taken into consideration. In the past, panels often discussed which modelling approach and which modelling language is most appropriate. We realise now the models and also modelling languages have their own obstinacy. So, model-based engineering is background-laden and directives-laden.

Model-based engineering is based on the modelling know-how, on modelling practices, on modelling theory, and on modelling economics. We discussed the ingredients for model-based engineering for the case of co-models and of mediating models. This approach can be generalised to full co-design of structuring, functionality, interactivity and distribution. So far, the approach uses model suites. How this approach can be extended to any kind of model collections is an open research problem.

The paper has been restricted to the general programme of model-based engineering. The explicit and detailed description is the topic of two forthcoming papers. Model-based engineering uses a number of practices similar to SPICE or CMM approaches [18].

We may now combine our investigation in Figure 4. We distinguish the six dimensions: community of practice, background/knowlegde/context, application scenario and stories of model utilisation, situation/state/data, dynamics/evolution/change/operations, and models as representations and instruments. Models are used in a variety of functions. For instance, models of situations/states/data are often used for structuring, description, prescription, hypothetic investigation, and analysis. Models are used by members of the community of practice for communication, reflection, understanding, and negotiation. So, we observe that the function (or simpler the purpose or the goal) of the model is determined by the concrete way how a model is used. Model-based engineering is thus engineering supported by models that are used according to the function that a model might play in the engineering process.

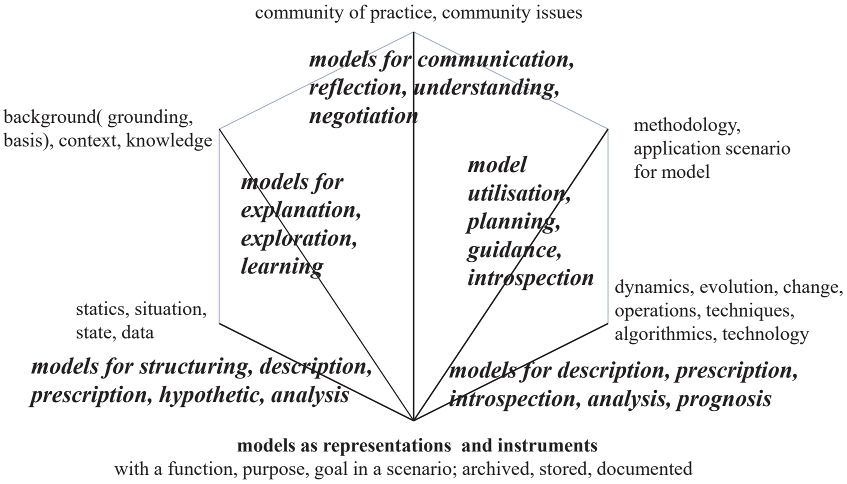


Fig. 4. Models and the five concerns in model-based engineering

References

1. B. AlBdaiwi, R. Noack, and B. Thalheim. Pattern-based conceptual data modelling. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 1–20. IOS Press, 2014.
2. N. Aquino, J. Vanderdonckt, J. I. Panach, and O. Pastor. Conceptual modelling of interaction. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 10, pages 335–358. Springer, Berlin, 2011.
3. D. Bjørner. *Domain engineering*, volume 4 of *COE Research Monographs.* Japan Advanced Institute of Science and Technolgy Press, Ishikawa, 2009.

4. C. Blättler. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Das Modell als Medium. Wissenschaftsphilosophische Überlegungen, pages 107–137. De Gruyter, Boston, 2015.
5. E. Brann. *The logos of Heraclitus.* Paul Dry Books, 2011.
6. J.E. Brenner. The logical process of model-based reasoning. In L. Magnani, W. Carnielli, and C. Pizzi, editors, *Model-based reasoning in science and technology*, pages 333–358. Springer, Heidelberg, 2010.
7. A. Dahanayake and B. Thalheim. Co-evolution of (information) system models. In *EMMSAD 2010*, volume 50 of *LNBIP*, pages 314–326. Springer, 2010.
8. W. Doerr and H. Schipperges, editors. *Modelle der Pathologischen Physiology.* Springer, 1987.
9. D. Dori. Object-process methodology for structure-behavior codesign. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 7, pages 209–258. Springer, Berlin, 2011.
10. A. Düsterhöft and K.-D. Schewe. Conceptual modelling of application stories. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 11, pages 359–380. Springer, Berlin, 2011.
11. D. Embley and B. Thalheim, editors. *The Handbook of Conceptual Modeling: Its Usage and Its Challenges.* Springer, 2011.
12. D. W. Embley and W.Y. Mok. Mapping conceptual models to database schemas. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 1, pages 123–164. Springer, Berlin, 2010.
13. E. D. Falkenberg, W. Hesse, and A Olive, editors. *Information system concepts: towards a consolidation of views.* Chapmann & Hall, 1995. Proceedings IFIP WG8.1 Conferences ISCO3.
14. I. Hacking. *Representing and Intervening. Introductory topics in the philosophy of natural science.* Cambrdge Press, 1983.
15. H. Haken, A. Wunderlin, and S. Yigitbasi. An introduction to synergetics. *Open Systems and Information Dynamics*, 3:97–130, 1995.
16. W. Hesse. Modelle - Janusköpfe der Software-Entwicklung - oder: Mit Janus von der A- zur S-Klasse. In *Modellierung 2006*, volume 82 of *LNI*, pages 99–113. GI, 2006.
17. P. J. Hunter, W. W. Li, A. D. McCulloch, and D. Noble. Multiscale modeling: Physiome project standards, tools, and databases. *IEEE Computer*, 39(11):48–54, 2006.
18. ISO/IEC. Information technology - process assessment - part 2: Performing an assessment. IS 15504-2:2003, 2003.
19. R. Kaschek. *Konzeptionelle Modellierung.* PhD thesis, University Klagenfurt, 2003. Habilitationsschrift.
20. Y. Kiyoki and B. Thalheim. Analysis-driven data collection, integration and preparation for visualisation. In *Information Modelling and Knowledge Bases*, volume XXIV, pages 142–160. IOS Press, 2013.
21. A. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture - Practice and Promise.* Addison Wesley, 2006.
22. M. Klettke and B. Thalheim. Evolution and migration of information systems. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 12, pages 381–420. Springer, Berlin, 2011.
23. J. Krogstie. *Model-based development and evolution of information systems.* Springer, 2012.
24. A.V. Lebedev. *The Logos Heraklits - A recomnstruction of thoughts and words; full commented texts of fragments (in Russian).* Nauka, 2014.

25. P. C. Lockemann and H. C. Mayr. *Computer-based information systems.* Springer, Berlin, 1978. In German.
26. L. Maciaszek. *Requirements analysis and design.* Addison-Wesley, Harlow, Essex, 2001.
27. L. Magnani, W. Carnielli, and C. Pizzi, editors. *Model-Based Reasoning in Science and Technology: Abduction, Logic, and Computational Discovery.* Springer, 2010.
28. B. Mahr. *Mathesis & Graphé: Leonhard Euler und die Entfaltung der Wissenssysteme*, chapter Denken in Modellen, pages 85–100. Akademie-Verlag, 2010.
29. M.S. Morgan and M. Morrison, editors. *Models as mediators.* Cambridge Press, 1999.
30. N. J. Nersessian. *Creating Scientific Concepts.* MIT Press, 2008.
31. A. Olivé. Conceptual schema-centric development: A grand challenge for information systems research. In *Proc. CAiSE*, pages 1–15, 2005.
32. A. Olivé. *Conceptual modeling of information systems.* Springer, Berlin, 2007.
33. R. Oudrhiri. Software engineering economics: A framework for process improvement. In I. Comyn-Wattiau, C. du Mouza, and N. Prat, editors, *Ingenierie Management des Systemes D'Information*, pages 265–282. Cépaduès, 2016.
34. K. Pohl. *Process centred requirements engineering.* J. Wiley and Sons Ltd., 1996.
35. J.E. Safra, I. Yeshua, and et. al. *Encyclopædia Britannica.* Merriam-Webster, 2003.
36. I. Ramos Salavert, O. Pastor López, J. Cuevas, and J. Devesa. Objects as observable processes. In A. Olivé, editor, *Proc. 4th Int. Workshop on the Deductive Approach to Information Systems and Databases, DAISD'93*, pages 51–72, Lloret de Mar, 1993.
37. A. Samuel and J. Weir. *Introduction to Engineering: Modelling, Synthesis and Problem Solving Strategies.* Elsevier, Amsterdam, 2000.
38. K.-D. Schewe and B. Thalheim. Development of collaboration frameworks for web information systems. In *IJCAI'07 (20th Int. Joint Conf on Artificial Intelligence, Section EMC'07 (Evolutionary models of collaboration)*, pages 27–32, Hyderabad, 2007.
39. K.-D. Schewe and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter Co-Design of Web Information Systems, pages 293–332. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.
40. T. Stahl and M. Völter. *Model-driven software architectures.* dPunkt, Heidelberg, 2005. (in German).
41. W. Steinmüller. *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik.* Wissenschaftliche Buchgesellschaft, Darmstadt, 1993.
42. B. Thalheim. *Entity-relationship modeling – Foundations of database technology.* Springer, Berlin, 2000.
43. B. Thalheim. *The Conceptual Framework to Multi-Layered Database Modelling based on Model Suites*, volume 206 of *Frontiers in Artificial Intelligence and Applications*, pages 116–134. IOS Press, 2010.
44. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
45. B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In *Encyclopedia of Database Systems.* 2017.
46. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung.* De Gruyter, Boston, 2015.

47. B. Thalheim and M. Tropmann-Frick. The conception of the conceptual database model. In *ER 2015*, LNCS 9381, pages 603–611, Berlin, 2015. Springer.
48. M. Tropmann and B. Thalheim. Performance forecasting for performance critical huge databases. In *Information Modelling and Knowledge Bases*, volume XXII, pages 206–225. IOS Press, 2011.
49. Q. Wang and B. Thalheim. Data migration: A theoretical perspective. *DKE*, 87:260–278, 2013.

Conceptual Modeling Perspectives, 137--153, Cham, 2017, Springer

General and Specific Model Notions

Bernhard Thalheim

Christian Albrechts University Kiel, Department of Computer Science
Olshausenstr. 40, D-24098 Kiel, Germany
thalheim@is.informatik.uni-kiel.de

Abstract. Models are a universal and widely used instrument in Computer Science and Computer Engineering. There is a large variety of notions of models. A model functions in a utilisation scenario as an instrument. It is well-formed, adequate and dependable. It represents or deputes origins. This conception of the model is a very general one. Based on the notion of a stereotype as a starting point we show that specific or particular model notions are specialisations of the general notion.

1 Models in Computer Engineering and Computer Science

Models are principle instruments in modern computer engineering (CE), in teaching any kind of computer technology, and also modern computer science (CS). They are built, applied, revised and manufactured in many CE&CS subdisciplines in a large variety of application cases with different purposes and context for different communities of practice.

1.1 The Omnipresence of Models in CE&CS

The wide deployment of models is supported by an expansive scientific literature on model usages. There are many different model notions, e.g. [30] discussed more than 50 different definitions of models used in CE&CS programs. All subdisciplines in CE&CS use models such as phenomenological models, computational models, developmental models, explanatory models, didactic models, imaginary models, mathematical models, substitute models, iconic or diagrammatic models, formal models, and analogue models. There is no branch in CE&CS that does not widely use models as instruments.

It is now well understood that models are something different from theories. They are often intuitive, visualisable, and ideally capture the essence of an understanding within some community of practice and some context. At the same time, they are limited in scope, context and the applicability.

We realised also that models become an research issue on their own. Models are expressions, descriptions, icons, statements, etc. from one side and desiderata, representations, deputies, instruments, designs, products etc. from the other side. They might suggest something that we might later be able to explain or to construct. Models also help us to explain a system, help us to deal with more realistic situations, and tell us which intuition and understand is a good one. *How we handle such variety of deployments, understandings, and approaches?*

LNCS 10509, 13--27, Cham, 2017, Springer

1.2 The General Notion of the Model

There is however a general notion of a model and of a conception of the model:

> *A* **model** *is a well-formed, adequate, and dependable instrument that represents origins.* [6, 24–26]

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

The model should be well-formed according to some well-formedness criterion. As an instrument or more specifically an artifact a model comes with its *background*, e.g. paradigms, assumptions, postulates, language, thought community, etc. The background its often given only in an implicit form.

The background is often implicit and hidden. *Is there any approach to consider the background in a simpler form?*

An well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*.

So far, the adequateness notion is far too fuzzy and too wide. *Can be develop a simpler notion of adequateness that still covers the approaches we are used in our subdiscipline?*

Well-formedness enables an instrument to be *justified* by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability, and by stability and plasticity within a collection of origins.

The instrument is *sufficient* by its *quality* characterisation for internal quality, external quality and quality in use or through quality characteristics [20] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

A well-formed instrument is called *dependable* if it is sufficient and is justified for some of the justification properties and some of the sufficiency characteristics.

Again, dependability is a wide field. *Do we need this broad coverage for models? Or is there any specific treatment of dependability for subdisciplines or specific deployment scenarios?*

If there are many specific and particular notions of the model: *Can we relate different notions of models with each other? Can be define interfaces among models? Is there any standard notion for a sub-discipline? Are specific or particular notions derivable from the general notion of the model?*

And finally, this notion is a very general one. *How does the general notion match with other understandings and approaches to modelling in CE&CS?*
Or more generally for sciences based on Occam's razor principle: *Are there specific or particular notions of the model within specific constellations that sufficiently represent all relevant aspects requested and nothing more?*

1.3 Generality versus Specificity

The general notion of a model covers all aspects of adequateness, dependability, well-formedness, scenario, functions and purposes, backgrounds (grounding and basis), and outer directives (context and community of practice). It covers all known so far notions in agriculture, archeology, arts, biology, chemistry, computer science, economics, electrotechnics, environmental sciences, farming, geosciences, historical sciences, languages, mathematics, medicine, ocean sciences, pedagogical science, philosophy, physics, political sciences, sociology, and sports. The models used in these disciplines are instruments used in certain scenarios.

Sciences distinguish between general, particular and specific things. Particular things are specific for general things and general for specific things. The same abstraction may be used for modelling. We may start with a general model. So far, nobody knows what is such general model for most utilisation scenarios. Models *function* as *instruments* or tools. Typically, instruments come in a variety of forms and fulfill many different functions. Instruments are partially independent or autonomous of the thing they operate on. Models are however special instruments. They are used with a specific intention within a utilisation scenario. The quality of a model becomes apparent in the context of this scenario.

It might thus be better to start with generic models. A ***generic model*** [3, 15] is a model which broadly satisfies the purpose and broadly functions in the given utilisation scenario. It is later tailored to suit the particular purpose and function. It generally represents origins under interest, provides means to establish adequacy and dependability of the model, and establishes focus and scope of the model. Generic models should satisfy at least five properties: (1) they must be accurate; (2) the quality of the generic model allows that it is used consciously; (3) they should be descriptive, not evaluative; (4) they should be flexible so that they can be modified from time to time; (5) they can be used as a first "best guess".

Generic models might also be an abstraction of other models that are used as an inspiration for development of the new model and that are based on the experience of the modeller. Generic models can be calibrated to specific models through a process of data or situation calibration, refinement, concretisation, context enhancement, or instantiation.

Generic models [29] are typically specialised to more specific ones in a development process. Generic models are widely used under different names or development approaches such as inverse modelling, model-driven architectures and development, universal applications, data mining and analysis, pattern-based development, reference models, inductive learning, and model forensics. All these approaches develop models by stepwise refinement of the root or initial model, by selection and integration of model variations, and by mutation and recombination of the model where the the root model is a generic model with parameters (also structures and operations as parameters as well as the architecture).

Instead, we also may start with general models. Typically, we prefer however ***particular*** or idealised ***models*** as a starting point for a specific community of practice with a specific background, within a specific context, and for representation of a specific world of origins under consideration. Generic models can be calibrated to specific models through a process of data or situation calibration, refinement, concretisation, context enhancement, or instantiation. *Lightweight models* [28] typically cut off background and context. They assume per default some utilisation scenario and reduce the functions of the model to the main function. The purpose is then driven by this function. Often the community of practice is set to some standard community that uses a specific kind of justification.

Therefore, we face the problem: *What is the best starting point for development of a model?* This paper answers this question by introducing stereotypes of particular models in Sections 3 and 4. For this we use the separation of abstraction into stereotypes, pattern, and templates [1].

1.4 The Storyline and Objectives of This Paper

Since the model notion is too broad we might ask ourselves whether more specific notions can be used in subdisciplines of CE&CS. We might also consider whether some of the proposed notions are simpler and better to use. We might start with the main properties of models (mapping or analogy, truncation or abstraction or focus, pragmatic, amplification, distortion, idealisation, carrier, added value, purpose [12, 16, 18, 21]) and specialise them. We might also discuss the variety of notions [23, 30] and compare them with the general one. The main question is however whether these different notions are sufficient within their environment, i.e. which specific notion of the model is sufficient for which utilisation, for which community, within which context, under which general conditions and within which understanding.

Models are used as perception models, mental models, situation models, experimentation models, formal model, mathematical models, conceptual models, computational models, inspiration models, physical models, visualisation models, representation models, diagrammatic models, exploration models, heuristic models, etc. Although this categorisation provides an entry point for a discussion of model properties, the phenomenon of being a model can be properly investigated. Each category is too broad and combines too many different aspects at the same time.

We thus first discuss notions which are commonly accepted and discover that these notions are laden by background, community, context, and utilisation scenarios. This ladenness can be represented by definitional frames for the model notion. These frames may now be used for defining stereotypes of model notions.

2 Specialising and Refining the Model Notion

2.1 Stereotypes for Models and Particular Notions of Models

Modelling stereotypes describe the general modelling situation. Generic models are typically a general modelling solution in a certain utilisation scenario, context, background, and community of practice. For instance, a *structure stereotype* describes data structuring environment within a certain modelling situation. The corresponding generic models can be refined and used during model development. They can be considered to be classes or collections of potential models.

2.2 Two Model Notions and their Specific Approaches

Let us consider two of the 49 model notions we collected [27] for CE&CS. We will show that these notions are applicable but are heavily biased and thus paradigmatically use a lot of latent semantics behind.

Models for Model-Based Development. The Scandinavian and Dutch schools of (conceptual) modelling have developed a sophisticated approach to modelling since the late 60ies. One result is the famous FRISCO report [7]. More recently, J. Krogstie [11] states:
"Model*: A model is an abstraction externalised in a professional language. A model is assumed to be simpler than, resemble, and have the same structure and way of functioning as the phenomena it represents.*
Phenomenon*. A phenomenon is something as it appears in the mind of a person. The world is perceived by persons to consist of phenomena. ...*
Property*. A property is an aspect of a phenomena that can be described and given a value. A phenomenon will have a set of potentially relevant properties. ...*
Constitutive rule*. A deontic rule that applies to phenomena that exist only because a rule exist. ...*
Professional language*. A professional language is a language used by set of perosns working in certain kind of area or in a scienctrific discipline. Usually such a language is not learned before the person has been active in the area for a while.*
Language model*. The model of a language. Within conceptual modelling, this is often termed 'meta-model', which is only a proper term when looking upon it from the point of view of repository-management for a modelling tool where the instantiation of the model is another model in the same or a different modelling language.*
Conceptual model*. A model of a domain made in a formal language or semi-formal language with a limited vocabulary.*
System*. A system is a set of correlated phenomena, which is itself a phenomenon. ...*
System model*. A model of a system.*"

Analysing these notions and more specifically the notion of the model, we realise that there must exist an origin that we can call *matured perception model.* At the same time, the modelling approach is entirely biased by its discipline, its school of thought, it context, and - as a partially explicit component - its community of thought. At the same time, we consider only phenomena in a set-based fashion and not within a conception/conception network. So, the modelling approach is using a rather restricted world view.

This restricted world view is however entirely sufficient since the model is used in one very specific utilisation scenario: system construction. We observe the ***import of latent*** paradigmatic (computing-oriented, function-backed, economic, ...) ***models*** with predefined meaning, specific context and background concepts (space, time, settlement, environment, ...) within this scenario. The main function of the model is that of a ***mediator*** that describes the (augmented and perceived) model and that prescribes a system to be investigated or perceived. The adequateness property uses homomorphisms.

This approach is typical for model-based (software) development [4, 10, 11, 17] within the specific consideration of specific platform-independent models such as conceptual models and of platform-dependent models as refinements of the generic ones. This approach uses latent hidden generic models as community knowledge. Beside the community dependence, the development biases are also latent in this model notion.

Model Notions with Justification. Extending and revisiting the model notion with its mapping, truncation and pragmatic properties by H. Stachowiak [16], R. Kaschek [9] introduces a ***model*** as a material or virtual *artifact* (1) that is called a model within a community of practice (2) based on a judgement (3) of appropriateness for representation of other artifacts (things in reality, systems, ...) and serving a *purpose* (4) within this community.
Already [9] discussed the forgetful development of software products. Classically we observe that (i) developers base their design decisions on a “partial reality”, i.e. on a number of observed properties within a part of the application, (ii) developers are developing the information system within a certain context, (iii) developers reuse their experience gained in former projects and solutions known for their reference models, and (iv) developers use a number of theories with a certain exactness and rigidity.
The design decisions made during the design process are deeply influenced by these four hidden factors. In some approaches revisions made during the information systems development are recorded. However, since the background knowledge is not recorded the documentation of the information systems development is fragmentary.

The justification of models [9] is here explicit. It should however be combined with a statement of quality that has been achieved so far. The quality criteria are implicit. The model notion [9] is based on the community of practice behind the model. Forgetful development is one of the specific properties. The community of practice drives the context of the model and of modelling. At the same time, appropriateness is more general than (homomorphic) mapping and truncation.

2.3 The Background as the Hidden Component of Models

The two cases show that the model notion is often laden by its specific background. The background consists of undisputable elements (grounding: paradigms, postulates, restrictions, theories, culture, foundations, commonsense) and disputable one (basis: concepts, foundations, language as carrier, assumptions, thought community, thought style, conventions, practices). ***Background laden*** models are already using the grounding and the basis without making it explicit.

2.4 The Particular Notion of a Conceptual Model

Conceptual models are nothing else as models that incorporate concepts and conceptions which are denoted by names in a given name space. A concept space [1] consists of concepts [13] as basic elements, constructors for inductive construction of complex elements called conceptions, a number of relations among elements that satisfy a number of axioms, and functions defined on elements.

The general Sapir-Whorf hypothesis [33, 36] states the principles of language determinism (the language governs thinking) and language relativity (coded distinctions made in one language might not be expressible in another language). The weak form refers to the dependence of perception, remembering and simplicity on language. We may transfer this hypothesis to *concept-ladenness* of languages. Some languages might have richer concepts and conceptions than others[2]. Therefore, concepts and conceptions that are expressed in certain language heavily influence semiotics of models since the basis of models is also concerned with concepts and conceptions to be used and thus related to the the (discipline's context).

They use a specific background: a concept space that clarifies the meaning of the elements of the model. The concept space is often application dependent and based on the understanding of notions in the application area. The linguistic meaning of designators and annotations is an inherent but hidden element of the

So, we notice: the conceptual model is ***concept space laden***.

2.5 The Ladenness of Model Notions

In a similar way we observe also other kinds of ladenness:

***Context-ladenness*:** The application domain and disciplinary context is often already given due to the introduction of the model. It is often enhanced by focus and scope depending on the concrete deployment of the model. The time and space issues are typically implicit.

***Community-ladenness*:** A community of practice tries to be efficient. Such kind of efficiency includes an agreement of the way how thing are considered, i.e. a "school of thought" and commonly accepted practices, conventions, and assumptions.

***Development- and utilisation-ladenness*:** Models must function effectively within the utilisation scenarios. For this reason, a number of biases are inherited by the the model notion due to the orientation and function of the model. Utilisation also determines most of the quality characteristics, the assessment of the model, and the tolerance that might be applied.

---

[1] We follow R.T. White [24, 35] and distinguish between concepts, conceptual, conceptional, and conceptions.

[2] Think for instance about the finer notions for whole in Aborigine language: yarla, pirti, pirnki, kartalpa, yulpilpa, mutara, nyarrkalpa, pulpa, makarnpa, and katarta.

2.6 Lessons Learning: Towards a General Approach to Modelling

We observe that modelling mainly consists of three macro-steps, two intentional and implicit and one extensional and explicit:
(I) *Setting the definitional frame* with priming, language, and actor setting: Priming defines the undisputable decisions (called grounding), the concept space, and the context. Actors within a community of practice act in certain roles while fulfilling a task. They are biased by their disputable but somehow accepted background or basis.
(II) *Choice of a model stereotype* consisting of accepting the definitional frame, of agenda setting, and of initialisation pattern: Agenda setting restrict potential utilisation scenarios of models. It thus results in a clarification of the model functions and thus also purpose and goal. Initialisation may be based on generic models or modelling experience, e.g. on the basis of reference models.
(III) *Model development and deployment* is the classical macro-step and well investigated for many modelling problems.
The two first intentional macro-steps are hardly often explicitly mentioned. We often use already existing models (generic, reference, perception, situation, documentation, etc.) as a starting point without making a reference to it.

3 Definitional Frames for Model Notions

Definitional frames are often somehow agreed practice and commonsense within a context and within a community of practice. They are somehow implicit. Without knowing and managing them we might however come-up with models that drive us to spurious results or pitfalls. This paradox is well known for natural sciences or economics. Disputes in the past on whether semantical modelling, object-role modelling, relational modelling etc. are based on a misunderstanding of the definitional frames that have been used.

3.1 Priming and Orientation

The model is mostly developed within some *context* of a discipline, an application area, and an environment such as an infrastructure. Context may also incorporate certain foci and scopes for the model. Context may also be concerned with time. The context is taken as granted and not questioned.

Models are instruments and therefore design for utilisation. That means they are also set into the existing world. This world is based on some fundament or *grounding*. The grounding consists of the commonly accepted and not disputed postulates, paradigms, restrictions, theories, culture, foundations, and commonsense. Models thus inherit this grounding and do not explicitly refer to this grounding.

Models represent origins. These origins bring in their own world view, their own concepts and conceptions. The *concept(ion) space* is therefore for models some referred background. It is used for assigning a meaning to the constructs of the model, for consideration of properties of the model, and for validation of the model. Therefore, models often use concepts either in an explicit form (becoming thus conceptual models) or in a reference form as abstract formal notations which provide potentially an explanation of the model and its elements (most often for formal or mathematical models). In the first case, the concept space is given and not disputed whereas in the second case the concept space is hidden but available upon demand.

The fourth component of priming is the *context* agreement. It integrates the application domain, the specific thoughts in this application and thus the disciplinary context, scope, focus, infrastructure and time. We answer the when, whereat, whereabout, wherein, where, for what, wherefrom, and whence questions, may be partially also the what question.

3.2 Actors

The community of practice is far more influential than typically assumed. Community members play their specific roles, have their task portfolio, responsibilities and obligations during a development process. They have however also their interests which are injected into the modelling decisions. They have their preferred method spectrum and neglect others [2]. So, they choose also the modelling language [32] with all the limitations and potential of the language.

A community of practice is typically not interested in revision of the grounding. The community agrees typically also on the basis, i.e. on assumptions, on the thought style and understanding, on practices, and on conventions within the setted definitional frame. That means the community of practice determines the background meaning of a model and adequateness and dependability. The community also has a hidden raw understanding what means that a model is well-defined, analogous, focused and purposeful. A similar raw agreement is already made on dependability, i.e. on justification and sufficiency. The corroboration, rational coherence, validation, stability and plasticity is somehow already generically set and taken as commonly agreed.

So, we need to question the influence of the social and professional community: whom (to whom, by whom), whichever. These questions answer to the presetting of the model. The message of the model is the same within the community.

3.3 Languages and Basics

Languages enable and restrict at the same time [33, 36]. They have their own obstinacy and thus restrict representability. From the other side, the provide rules for well-formedness, especially for syntactic ones. Professional languages additionally provide rules for semantic well-formedness. The community of practice also introduces its rules of pragmatic well-formedness. So, the language supports 'beauty' of models due to the inherent phonetics. We known from audience theory that representation determines later thinking, usage, and understanding of a model.

For instance, ER modelling supports well a global-as-design procedure on the basis of a global conceptual schema. If we follow the approach that syntactics also determines the operations and the algebra [19] then the different viewpoints which are required by the business user can be expressed via view collections defined on top of the conceptual schema.

Due to the carrier property, the language enables also to adjust practices, methodologies, pattern, typical routines, and commonsense. These elements of the basis complete the background. The language has a symbolic level which formes the culture of its users and provides a meaning. Professional languages use denotations and connotations. They provide a code that professionals learned to read.

So, the language answers the wherewith question. The imposed basics answer the question with what means. The language and the background form together some kind of 'gatekeeper' since we implicitly decide what to represent.

4 Stereotypes of Models in Utilisation Scenarios

Stereotypes of modelling have already been considered in discussions on methodologies, e.g. [5,8,14]. Typically, a methodology is bound to one stereotype and one kind of model within one utilisation scenario. We can however be more flexible. Stereotypes are governing, conditioning, steering and guiding the model development. They determine the model kind, the background and way of modelling activities. They persuade the activities of modelling. They provide a means for considering the economics of modelling.

4.1 Starting with Completing the Definitional Frame

The potential definitional frames are either selected on convenience or after consideration of appropriateness. Often one frame is taken for granted in most IT modelling approaches. The definitional frame sets up the acceptable background of a model. It is typically implicit.

4.2 Model Utilisation Scenario

Models are used as instruments in some utilisation scenarios. They have a number of functions in these scenarios. Based on an understanding of these functions we know what is the goal and purpose of such models. Therefore, we can now define the profile (goals, purposes, functions) that a model must fulfill. Due to the instrument property we also know which tasks are going to be solved with instruments. That means we know the task portfolio.

The profile and the portfolio create the 'spin' of the model since they convey a value judgement that might be immediately apparent and they create inherent bias by setting of the modelling task. The spin attempts to steer the way a model becomes useful to others.

4.3 Agenda Setting

Finally, we can define what is the agenda of the modelling tasks and of the model deployment. The agenda setting answers the why, for which reason, wherefore, worthiness, and whither away questions. This agenda can be formalised as a protocols setting and an orientation behind the model.

Based on the agenda, we sketch also adequateness and dependability. We can determine what means that a model is well-formed, which analogy or similarity is going the be used, which kind of focus allows to restrict the modelling task, and what means to be purposeful for a model.

At the same time, we have set up the main justification approaches. We already know explanatory statements and viability for the elements of the model based on the profile and background. We can sketch the arguments that support the model. We reflect norms and standards accepted by the community of practice, e.g. common practices for achieving inner coherence. The validation procedure is already set up for the model. We also may use which kind of robustness the model must have in order not to be over-fitted.

The model must not represent anything what might be representable. We know in this pre-setting which quality characteristics for quality in use, external and internal quality must be observed and which ones can be neglected. The quality characteristics are enhanced by evaluation procedures. So, we already define which discrimination is tolerated, which modality (necessity, contingency or possibility, relativity) can applied within the context, and which confidence of the evaluation is necessary.

Justification and sufficiency form our criteria for dependability. We can define for the model that is intended to build what means to be admissible, rigid, right, and fit.

4.4 Initial Model Setting

Models represent their origins. We might start from scratch, explore origins, discover essential and relevant elements, decompose them and explore then the modelling task. A first (nominal) model is the result of a composition or amalgamation step. Model formulation results then in development of a model. We might also base modelling on already existing models either for a given system or on the basis of referential models. We might also start with a generic model. In all these cases, we are already conditioned by the definitional frame. Additionally, we selected a modelling workflow or development strategy [19, 22].

The initial setting also inherits *latent models* that come with the grounding, the context, and the basis.

After setting the stereotype, we start with model development according to the chosen strategy within the agenda and the definitional frame. The typical questions answered in this step are: whereof, how, what, with which restrictions. Additional questions are concerned with adequateness and dependability of the model especially with quality characteristics.

4.5 A Test Case for the Approach

We might consider all notions in [27, 31]. Let us only consider the *construction scenario* for IT systems. The *stereotype* we shall use incorporates (1) the typical and also specific *IT grounding* with all its paradigms, postulates, theories, foundations, culture, commonsense, and restrictions, (2) the *mediator function* of models in the construction scenario, (3) the *IT community of practice* with its obligations, interests, tasks portfolio from one side, and the biases accepted in the community such as school of thought, practices, commonsense, and assumptions, and (4) the selection of the *languages* and *concept space* that might be used. It also provides a collection of *reference models* as their basis for opportunities. These reference models are *latent models*.

So, in this case, the modelling case is based on the needs and the functions a model might play in system construction. The context is given by current IT systems, current infrastructures and by system development foci and scopes. Therefore, IT grounded is not reconsidered. The choice of the concept space is determined by the notion of the system. The community of practice determines the language and the biases the community likes. The agenda is a mediating one. The model is used either for description of a development idea and for prescription of a forthcoming system or for documentation of an existing system. Initialisation might be based on generic models, on reference models or on already existing models.

Then we arrive with some model definition as [34]:
"A model is a simplified reproduction of a planned or real existing system with its processes on the basis of a notational and concrete concept space. According to the represented purpose-governed relevant properties, it deviates from its origin only due to the tolerance frame for the purpose."

5 Concluding: Stereotyping as the Spinning Principle

Models are one of the instrument in sciences, engineering and every life. They are not yet properly understood for their way of functioning, their impact, their potential, their capacity, and their anti-profile (not-supported utilisations). We do not want to overload the notion. Models should be used and understood. Therefore, we need a notion that is as simple as possible in the given scenario and given situation. At the same time, we should not loose the specific agreements we have made for models. Models must be effective, efficient, user-friendly, economic, and well-organised. Otherwise, nobody can properly use the conclusions and results that have been generated by the help of models. Sometimes, models may mis-orientate, condition, biase or persuade [23] users in their understanding and must be corrected after paradigmatic revision and synthesis.

So, we need a general specification of the model kind that allows from one side to reason on the potential, capacity, adequacy, and dependability of the given model and from the other side to be aware of the anti-profile and the cases in which the model is not promising, not adequate, may direct to wrong conclusions, and has its pitfalls.

This paper uses definitional frames and stereotypes for a holistic treatment of models. From one side, the model notion covers all what is necessary. From the other side, the specific agreements have to be explicitly given and must not be guessed. So providing the stereotype allows to understand the model, its quality characteristics, its capacity and its potential. It also allows to understand in which cases the model is not useful or more explicitly to know in which cases the model should not been used.

This paper does not claim that existing models or model notions are bad. We cannot handle here the large variety of modelling techniques. Model management is out of scope of this paper. Instead, we contribute to general model theory and harmonise notions of models by development of an approach that allows to derive specific notions of a model from the general one and thus to inherit investigations made for one model notion by other approaches to modelling.

References

1. B. AlBdaiwi, R. Noack, and B. Thalheim. Pattern-based conceptual data modelling. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 1–20. IOS Press, 2014.
2. R. Berghammer and B. Thalheim. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Methodenbasierte mathematische Modellierung mit Relationenalgebren, pages 67–106. De Gryuter, Boston, 2015.
3. A. Bienemann, K.-D. Schewe, and B. Thalheim. Towards a theory of genericity based on government and binding. In *Proc. ER'06, LNCS 4215*, pages 311–324. Springer, 2006.
4. D. Bjørner. *Domain engineering*, volume 4 of *COE Research Monographs*. Japan Advanced Institute of Science and Technolgy Press, Ishikawa, 2009.
5. M. H. Brackett. *Data resource quality*. Addison-Wesley, Boston, 2000.
6. D. Embley and B. Thalheim, editors. *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*. Springer, 2011.
7. E. D. Falkenberg, W. Hesse, P. Lindgren, B. E. Nilsson, O. J. L. Han, C. Rolland, R. K. Stamper, F. J. M. Van Assche, A. A. Verrijn-Stewart, and K. Voss. *A Framework of Information System Concepts, The FRISCO Report (Web Edition)*. IFIP, ifip@ifip.or.at, 1998.
8. C. C. Fleming and B. von Halle. *Handbook of relational database design*. Addison-Wesley, Reading, MA, 1989.
9. R. Kaschek. *Konzeptionelle Modellierung*. PhD thesis, University Klagenfurt, 2003. Habilitationsschrift.
10. A. Kleppe, J. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture - Practice and Promise*. Addison Wesley, 2006.
11. J. Krogstie. *Model-based development and evolution of information systems*. Springer, 2012.
12. B. Mahr. Information science and the logic of models. *Software and System Modeling*, 8(3):365–383, 2009.
13. G. L. Murphy. *The big book of concepts*. MIT Press, 2001.
14. G. Simsion. *Data modeling - Theory and practice*. Technics Publications, LLC, New Jersey, 2007.
15. G. Simsion and G.C. Witt. *Data modeling essentials*. Morgan Kaufmann, San Francisco, 2005.
16. H. Stachowiak. Modell. In Helmut Seiffert and Gerard Radnitzky, editors, *Handlexikon zur Wissenschaftstheorie*, pages 219–222. Deutscher Taschenbuch Verlag GmbH & Co. KG, München, 1992.
17. T. Stahl and M. Völter. *Model-driven software architectures*. dPunkt, Heidelberg, 2005. (in German).

18. W. Steinmüller. *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1993.
19. B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
20. B. Thalheim. 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`.
21. B. Thalheim. 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*, chapter 17, pages 547–580. Springer, Berlin, 2011.
22. B. Thalheim. The art of conceptual modelling. In *Information Modelling and Knowledge Bases XXII*, volume 237 of *Frontiers in Artificial Intelligence and Applications*, pages 149–168. IOS Press, 2012.
23. B. Thalheim. The conception of the model. In *BIS*, volume 157 of *Lecture Notes in Business Information Processing*, pages 113–124. Springer, 2013.
24. B. Thalheim. The conceptual model ≡ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
25. B. Thalheim. Models, to model, and modelling - Towards a theory of conceptual models and modelling - Towards a notion of the model. Collection of recent papers, http://www.is.informatik.uni-kiel.de/∼thalheim/indexkollektionen.htm, 2014.
26. B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In *Encyclopedia of Database Systems*. 2017.
27. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*. De Gruyter, Boston, 2015.
28. B. Thalheim and M. Tropmann-Frick. Wherefore models are used and accepted? The model functions as a quality instrument in utilisation scenarios. In I. Comyn-Wattiau, C. du Mouza, and N. Prat, editors, *Ingenierie Management des Systemes D'Information*, 2016.
29. B. Thalheim, M. Tropmann-Frick, and T. Ziebermayr. Application of generic workflows for disaster management. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 64–81. IOS Press, 2014.
30. M. Thomas. Modelle in der Fachsprache der Informatik. Untersuchung von Vorlesungsskripten aus der Kerninformatik. In *DDI*, volume 22 of *LNI*, pages 99–108. GI, 2002.
31. O. Thomas. Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation. Technical Report Heft 184, Institut für Wirtschaftsinformatik, DFKI, Saarbrücken, Mai 2005.
32. C. von Dresky, I. Gasser, C. P. Ortlieb, and S Günzel. *Mathematische Modellierung: Eine Einführung in zwölf Fallstudien*. Vieweg, 2009.
33. P. Wanner, editor. *The Cambridge Encyclopedia of Language*. Cambridge University Press, 1987.
34. S. Wenzel. Referenzmodell fr die Simulation in Produktion und Logistik. *ASIM Nachrichten*, 4(3):13–17, 2000.
35. R.T. White. Commentary: Conceptual and conceptional change. *Learning and instruction*, 4:117–121, 1994.
36. B.L. Whorf. *Lost generation theories of mind, language, and religion*. Popular Culture Association, University Microfilms International, Ann Arbor, Mich., 1980.

Normal Models and Their Modelling Matrix

Bernhard Thalheim

Department of Computer Science, Christian Albrechts University Kiel, D-24098 Kiel
thalheim@is.informatik.uni-kiel.de

Abstract. Models are one of the central instruments of modern Computer Science and Computer Engineering. The notion of the model is however not commonly agreed. There are - from one side - very general and universal notions and - from the other side - rather specific ones which are easy to use within some focus and scope and fail to be applicable in other sub-disciplines. Model development is typically based on an explicit and rather quick description of the 'surface' or normal model and on the mostly unconditional acceptance of a deep model. We discover that model development is based on stereotypes. The basis of a stereotype is the deep model which is tacit and latent knowledge in normal models. The deep model is the 'logos' of a normal model. The scenarios of model deployment and the functions the model plays in these scenarios are tacit and latent engineering in normal models.

Keywords: model, model notion, conceptual modelling, (modelling) matrix, deep model, normal model

1 Models, Models, Models: Everywhere but Different

1.1 101 Notions of the Model Concept

Computer science and computer engineering expressively use the conception of model for daily work. Modelling is one of their four central paradigms beside structures (in the small and large), evolution or transformation (in the small and large), and collaboration (based on communication, cooperation, and coordination). E.g. [69] selected 35 of notions which are commonly used in business informatics. As a very short list we may consider the following statements:

[3]: A model is a mathematical description of a business problem.
[4]: A model is the result of a construction process for which the selected part of the origin satisfies the purpose.
[7]: A model is the representation of an object system for the purpose of some subject. It is the result of a construction process by the modeller who addresses a representation of these objects for model user at a certain time and based on some language. A model consists of this construction, the origin, the time and a language.

[21]: A model can be simply considered to be a material or virtual artifact which is called model within a community of practice based on a judgement of appropriateness for representation of other artifacts (things in reality, systems, ...) and serving a purpose within this community.
[26]: A model is an abstraction externalised in a professional language. A model is assumed to be simpler than, resemble, and have the same structure and way of functioning as the phenomena it represents.
[33]: The model prescribes concepts as a particular kind of relation relating a subject and an entity.
[48]: A model is an object that has been developed and is used for solution of tasks which cannot be directly solved for the origin by a subject because of its structural and behavioural analogy to an origin.
[53]: Models are governed by the purpose, are mappings of an origin and reflect some of the properties observed or envisioned for the origin. They use languages as carrier.
[71]: A model is a simplified reproduction of a planned or real existing system with its processes on the basis of a notational and concrete concept space. According to the represented purpose-governed relevant properties, it deviates from its origin only due to the tolerance frame for the purpose.

The following general notion in [66] has been combined and generalised the understanding of the concept of a model in Archeology, Arts, Biology, Chemistry, Computer Science, Economics, Electrotechnics, Environmental Sciences, Farming and Agriculture, Geosciences, Historical Sciences, Humanities, Languages and Semiotics, Mathematics, Medicine, Ocean Sciences, Pedagogical Science, Philosophy, Physics, Political Sciences, Sociology, and Sport Science.

Definition 1 *[14, 47, 62, 64, 67] A* model *is an instrument that is adequate and dependable. It has a profile (goal or purpose or function), represents artifacts and is used for some deployment scenario. As an instrument, a model has its own background (e.g. foundation (paradigms, postulates, theories, disciplinary culture, etc.) and basis (concepts, language, assumptions, practice, etc.)). It should be well-defined or well-formed.*

Adequacy is based on satisfaction of the purpose, analogy to the artifacts it represents and the focus under which the model is used. Dependability is based on a justification for its usage as a model and on a quality certificate. Models can be evaluated by one of the evaluation frameworks. A model is functional if methods for its development and for its deployment are given. A model is effective if it can be deployed according to its portfolio, i.e. according to the tasks assigned to the model. Deployment is often using some deployment model, e.g. for explanation, exploration, construction, description and prescription.

1.2 Models as the Third Dimension of Science

Models have been considered to be somewhere in the middle between state of affairs (world, situations, data etc.) and theories (concepts and conceptions, statements, beliefs, etc.) since they may describe certain aspects of a situation and may represent parts of a theory. Figure 1 displays this understanding.

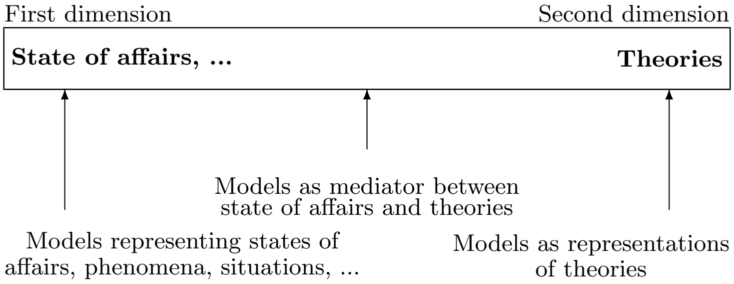


Fig. 1. Models between state of affairs and theories

“Models are partially independent of both theories and worlds.” [39] The understanding of a model to be a mediator between a world and a theory is however far too restricted.

Models should be considered to be the third dimension of science [8, 66, 68][1] as depicted in Figure. Disciplines have developed a different understanding of the notion of a model, of the function of models in scientific research and of the purpose of the model. Models are often considered to be artifacts. Models might also be mental models and thought concepts. Models are used in *utilisation scenarios* such as construction of systems, verification, optimization, explanation, and documentation. *In these scenarios* they *function* as *instruments*[2].

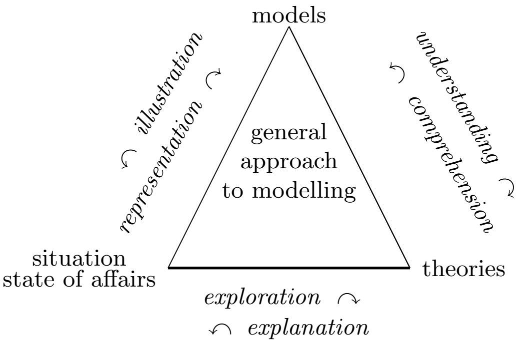


Fig. 2. Models are independent and are the third dimension of science

---

[1] The title of the book [11] has inspired this observation.

[2] An instrument is among others (1) a means whereby something is achieved, performed, or furthered; (2) one used by another as a means or aid or tool [45].

Given the utilisation scenarios, we may use models as perception models, mental models, situation models, experimentation models, formal model, mathematical models, conceptual models, computational models, inspiration models, physical models, visualisation models, representation models, diagrammatic models, exploration models, heuristic models, informative models, instructive models, etc. They are a means for some purpose (or better: function within a certain utilisation scenario), are often volatile after having been used, are useful inside and often useless outside the utilisation scenario. Models are different [8] in the four generations of science (empirical science,theory-oriented science, computational science, data science) [17].

1.3 Kuhn's Conception of Normal Science

T. Kuhn realised that the history of science consists of normal science punctuated by periods of revolution. He combined the underpinning of normal science by a notion of paradigms. His notion of paradigms substantially and circularly changes within his work. He did not get a satisfying definition for it[3]. He integrated normative and empirical disciplines and got a general picture on how science works.

Until his work, philosophers of science made some assumptions about how science works, because they were confident in the methods of science and its success. However failures and irrationalities were not explainable. Kuhn suggested that science can only be understood "warts and all". The main conception in his book [27] is the distinction between normal science and (revolutionary) evolving science. Normal science is strict and governed by what he called paradigm as an object of consensus within a community and context. Education in such sciences is governed by success stories, e.g. examples, and is not governed by rules or methods. So, it is far more dogmatic [9, 16, 27] . Dogmatism, brainwashing, and indoctrination have the advantage of simplicity, fruitfulness, parsimony, and understandability within certain community and are thus enablers of success. Normal sciences maintain confidence within this community. They condition the members of a scientific community.

The investigations Kuhn has made can also be observed for modern natural sciences. Modern physics is partially normal science. It uses *standard models.* For instance, astrophysics [19] is based on the Lyndon-Bell hypothesis that assumes the existence of a black hole in galaxies with accretion layers around the black hole with broad and narrow line regions it as the central energy source. The "inner" or "deep" model behind defines the space of possible models. The modelling approach develops the final model based on a composition approach starting with parametric model fragments. Each fragment is conditioned on

---

[3] Later [29], he revises this notion to a 'disciplinary matrix'. His understanding of the matrix combines a wider notion of commonsense within a community of practice for relatively unproblematic disciplinary communication and for relative common agreement. We shall turn to this notion and clarify what it means in this paper. Our clarification follows [30].

a background, i.e. a grounding and a basis, for instance, for approximations, perspective, granularity, and operating assumptions underlying the fragment. Mutually contradictory backgrounds are organised into background groups, and a set of coherence constraints is used to govern the use of the backgrounds [40]. Techniques such as constraint satisfaction, (causal, fitting, ...) approximations, conflict resolution graphs, and inverse modelling are then used for construction of an adequate and dependable model. Typical other deep models are the standard big bang model and the standard model of particle physics.

1.4 The Usefulness of a General Notion of the Model

The work [66] offered a rational reconstruction of the notion of a model. Definition 1 introduces an explicit and clear the full notion of the model, the reasons behind modelling, modelling decisions, and modelling practices. This work follows the positivist approach by [44, 43] and sees modelling as an a priori engineering discipline. Additionally, models have also an explanatory value. Models and modelling methods have their own obstinacy. They widen the understanding, theory, and engineering within an area and restrict at the same time.

We realised that models may be artifacts or mental things. Models are however going to be used in a certain way. This way can be stereotyped as a scenario. Models function within such scenarios. This functioning explains then what is the purpose and the goal of the model itself. Taking this turn, models are then *instruments* as an element of technology.

1.5 Outline of the Paper

Modelling does not start from scratch. Rather it uses previous experience, approaches developed so far, commonalities agreed within a community of practice, disciplinary and other context, and also a consensus within an application. Therefore, it is not 'greenfield' work. It might be 'brownfield' work for migration or modernisation project. In all cases, modelling is laden by its grounding, its basis, its specific application scenarios, its community of practice, and its context. This laddeness can be understood as the deep model or the 'logos' underneath the normal[4] model.

The main issue of this paper is the development of an understanding of the matrix of conceptual models. Matrices consist of deep models and modelling scenario setting. Normal models are governed by such modelling matrices.

---

[4] The word 'normal' has different meanings [1]. It seems that Computer Science and Logics prefer 'normal' as conforming or constituting a norm or standard or level or type or social norm. We prefer the meaning as being appropriately average or within certain limits or occurring naturally or being characterised by average development. Being 'normal' also means to be in accordance to accepted consensus or rules or laws.

2 Normal Modelling

Normal modelling is the day-to-day business of modelling in most areas and also in Computer Science and Computer Engineering. The matrix introduced in the sequel can be seen as a pervasive, disciplinary and well-accepted framework in which modellers perceive and develop their model. Normal modelling is concerned with model development and model deployment for the given application task and nothing else beyond that.

2.1 Modelling is Often Stereotyped

The modelling process is often stereotyped [65]. It reuses experience gained in the community of practice and especially by the modellers. It is additionally governed by puzzles and expectations and especially the origins with the selected concept space. This approach assures modellers that each model is adequate and dependable, and provides standards for evaluating its adequateness and dependability. It uses a definitional frame, is based on certain modelling situations - or better on certain *modelling scenarios* - that determine the *agenda* of the modelling process, and can be started from scratch or with an *initial model*, e.g. a generic model.

The *definitional frame* defines the setting of the modelling process, i.e. (1) its *priming and orientation* that is governed by the context (application domain or discipline, school of thought, time, space, granularity, scope) and the grounding (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), (2) the *actors* (which form its community of practice) with their roles, responsibilities, and obligations, and (3) the *language* (as a carrier) and *basics* (assumptions, concepts, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense). Another part of the definitional frame is defined as specific adequacy and dependability criteria which are applied to a model.

2.2 Modelling is Mainly Normal Modelling

We will realise in the sequel that such kind of stereotyped modelling uses two models behind the model. It defines a *macro-model* for the modelling process itself, i.e the way how the model and especially the surface model is going to be developed. It is also the basis for the *deep model* that directs the modelling process and the surface or normal model. The deep model can be understood as the common basis for a number of models. Education on conceptual modelling starts, for instance, directly with the deep model. In this case, the deep model has to be accepted and is thus hidden and latent.

Normal modelling is similar to normal science. It is based on some kind of consensus about how modelling should be done. It is governed by some implicit knowledge, by commonsense, and - more generally - by consensus behind that we call in the sequel '*matrix*'. It is thus puzzle-solving. Normal modelling is what modellers do most of the time. A typical puzzle-solving task is the development of a conceptual schema for a given application within a given business context for a given community of practice, and within a given system orientation. The next puzzle is then solved by the next conceptual schema. Textbooks and education mainly develops puzzle-solving skills.

In normal modelling, the modelling theories, modelling tools, modelling attitudes, and modelling assumptions comprise the modelling matrix. They are kept fixed, permitting the cumulative generation of puzzle-solutions The modelling matrix undergoes revision whenever the underlying technology, the context, or the application are changing. If the consensus on modelling is lacking then competing schools of thought possess differing procedures, theories, even practices. Normal modelling proceeds on the basis of perceived similarity to exemplars.

Education and also edification is governed more by examples than by rules or methods. E.g. the field of conceptual modelling is mainly taught on the basis of examples; even more: nowadays on the basis of toy examples. In daily practice, models should be used and understood. Therefore, we need a notion that is as simple as possible in the given scenario and given situation. At the same time, we should not loose the specific agreements we have made for models. Models must be effective, efficient, user-friendly, economic, and well-organised. Otherwise, nobody can properly use the conclusions and results that have been generated by the help of models. Sometimes, models may mis-orientate, condition, biase or persuade [60] users in their understanding and must be corrected after paradigmatic revision and synthesis.

The orientation on normal models has also its pitfalls. For instance, cardinality constraints [56] have mainly be developed for relational technology of the early 90ies or 80ies. At his time, the mapping of these constraints was a deep research issue. Nowadays, object-oriented database system technology allows a far more sophisticated handling of constraints. Maintenance can be deferred (eager or lazy integrity enforcement). Time management allows to handle more optimal timepoints for consistence maintenance. Consistency can be supported at the row level. Integrity constraints can be maintained at the application level. Integrity can be made through views. Finally, flexible strategies may be used, besides the no-action and rollback approach, e.g. on the basis of triggers or stored procedures. Therefore, we may generalise cardinality constraints to conditional cardinality constraints [54].

2.3 Normal Models are Governed by Their Modelling Matrix

A matrix is “something within or from which something else originates, develops, or takes from” [1]. The matrix is assumed to be correct for normal models. Normal modelling involves showing how systems and their models can be fitted into the elements the matrix provides. Most of this work is detail-oriented. So, the matrix governs the modelling process. A failure to solve a modelling task reflects on the modellers’ skills, and not on the legitimacy of the setting.

Normal modelling accepts one notion of a model as normal. It just happens in a broad set of presupposed, unquestioned assumptions that govern among other things the sort of models to be developed, how these models are investigated and deployed, and how these models are interpreted. If the matrix would be question then modelling becomes difficult if not too time consuming. The matrix guides and instructs. Normal modelling is perfectly good modelling as long as the tasks are solved by the models that have been developed. So, modellers can be 'blinded' by the success although they are close-minded. The consensus provides a good means for collaboration and minor modernisation. The modelling tasks are focussed on the task spectrum that is preferred at present. So, the matrix got its kind of faith and trust. This kind of 'brainwashing' or indoctrination is the basis for teaching.

2.4 Normal Modelling Develops its Boundaries

All matrices have their limit, restrictions and even pitfalls. It might happen that the model does not solve the task in a proper form or that the model is not sufficient or that models develop their obstinacy or that models result in anomalies. Problematic tasks are not counterexamples to normal modelling. In this case, the matrix must be revised since it is not adequate anymore.

The first resolution step is the introduction of new elements to the current matrix. E.g. the entity-relationship modelling language has been heavily extended by about 50 constructs in the 80ies and early 90ies [56] until it has been detected that these extensions will not become coherent. At the same time, work-arounds have been built for overcoming limitations. Currently object-relational technology is available. It seems that ER modelling might somehow suffice with model creation. However, classical theories are not sufficient anymore. The confidence into the ER approach to modelling weakens nowadays. With the advent of object-centred modelling and the supporting XML technology one might ask whether conceptual modelling can be based on the ER matrix. Since data collections might also evolve in their structuring, the class-oriented technique must be nowadays revised what has already been discovered within the research on conceptual modelling for 'big' data. Before changing the matrix we must, however, develop a proper understanding of it. We will turn to this problem in the next section.

The transformation of models become a bottleneck whenever the matrixes of the given model and of its transformation do not match properly. Such impedance mismatches have already widely been discussed for object-oriented programming in imperative environments. A similar observation is valid too for conceptual modelling based on sets and physical modelling based on multi-sets and references (see, for instance, SQL with specific referential integrity and multi-set handling).

3 A Case Study: Information Systems Modelling

3.1 Historical Matrices

The matrix is an essential component of the identity within a community of practice or within a scientific community. It identifies puzzles to be solved, governs expectation, assures modellers that each puzzle fulfills its purpose, and provides standards for evaluating.

According to [1], a matrix is "something within or from which something else originates, develops, or takes from". Normal models will be understood as models crammed with the modelling matrix. The existence of such modelling matrices makes modelling simpler and supports parsimony. It is a kind of complex laddeness of a model [65]. What modellers develop depends, in pertinent part, on what they already believe or expect. Developing is less passive, less receptive than many had thought. Modelling is dependent on the chosen modelling matrix.

Let us consider one example where we observe surprisingly many postulates, paradigms, theories, assumptions, accepted practices, bindings to a school of thought, context, and commonsense. The matrix is well-accepted but not explicitly explained in textbooks and research papers. The *entity-relationship modelling language* became popular in the late 70ies as a means for documenting logical relational schemata and for visualising the association among types[5]. The entity-relationship modelling language became now some kind of standard despite the unknown and not explicitly given matrix underlying this language. It uses a *Global-As-Design* approach where the schema reflects all viewpoints. Local viewpoints are derivable and somehow reflectable. The default semantics (and sometimes the only one to be considered) is *set semantics* for collections of objects for a type. The *reference semantics* for relationship types is hidden and not properly understandable during schema development but used in transformation. *Explicit existence existence* postulates that any object must exist before there can be a reference to it, i.e. rigid separation of creation and use. The model assumes a *closed-world view* and *unique names*. It is based on a well understood *name space* or glossary or ontology. *Salami-slice* representation uses homogenous, decomposed types (potentially with complex attributes) with incremental type construction. It is *type-centric*. According to tradition of logic-based computer science, *semantics follows syntax*, i.e. the definition of semantics can be given if the syntax is already defined that it uses. The user perspective is cut out, i.e. we base the model on *neglected pragmatics*. *Functionality representation* is deferred without consideration of the performance

---

[5] In his oral presentation of his keynote speech at ER96, C. Bachman [5] claimed that the new modelling languages has been introduced as a reaction of the inflexibility and due to the insufficiency of his network modelling language for representation of relational schemata. Due to the popularity of his modelling language it has not been possible for him to publish the new language. His claim has been the basis for the development of a new approach under his supervisorship [12] (reprinted in [14]).

impact to the schema. *Separation into syntax and semantics* allow to define semantics on top of the syntax. Explicit semantics is based on constraints. *Paradigms, postulates, assumptions* of database technology and database support are assumed due to the three main quality criteria (performance, performance, performance). *Basic data types* are hidden until mapping to facilities provided by DBMS typing systems or to logical models. *Visualisation* is represented by one holistic diagram that displays the entire syntax and semantics. One satyr or misbelief is representation of associations among types on the basis of binary types despite the valence of normal verbs in natural languages. In general, *binarisation* is possible by introduction of abstract artificial types and by relating the new type to each of the association component [6]. *The advent of data cubes has shown that an explicit* co-handling of views empowers database technology. *Star and snowflake schemata* introduced for data warehouses are nothing else than view in the basis of high-order relationship types [56].

3.2 Puzzle-Solving in Information Systems Modelling

In the sequel, we observe that the matrix forms the implicit and tacit knowledge behind modelling, i.e. the second component of the rigor cycle in design science research. The matrix generates a concensus about how modelling should be done. This consensus distinguishes modelling from other scientific or engineering endeavours.

Modern application with dynamic structuring of objects such as big data collections cannot be properly represented by the static structuring which is one of silver bullet assumptions of DBMS since it provides optimisation facilities that brought the victory of relational technology over network or hierarchical technology.

Puzzle-solving left open a good number of problems for future research [61]. One of the lacunas is the NULL marker problem [22, 46]. It becomes a bottleneck whenever aggregation functions are going to be applied [31]. The representation of NULL-polluted types by a collection of NULL-free subtypes is computationally infeasible. Schema-wide constraint maintenance is another big problem at present.

Education in this area has been built on success stories and proceeds on the basis of perceived similarity to success cases. At present, information system modelling still modelling in the small. Modelling in the large or modelling in the world must be based on different matrices; which ones is not clear yet. Puzzle-solving allows to transfer experience gained for one problem to another class of problems and to evaluate and appreciate solutions of other (e.g. reference or generic models). Design science research is oriented on cumulative addition

---

[6] RDF representation has chosen this way on the price of the maintenance and retrieval nightmare. A better binarisation has been used by MIMER resp. RAPID on the basis of sixth normal form storage (called nowadays one-column representation).

of new knowledge in terms of the application of the modelling and designing method.

3.3 Limitations and Pitfalls of Conceptual Modelling

Alike any language, the entity-relationship approach is not covering all issues. It is not cognitive complete since it represents only 2 of 6 cognitive categories[7] (container, link). Pitfalls of this approach are similar to the 88 pitfalls of object-oriented programming [70]. Salami-slice tactics is often not appropriate. Things in the application domain are however multi-facetted. For instance, a human is represented via a Person type that is separated from the Student type etc. The ER language is still based on the approach one-schema-one-diagram approach. Schemata are typically flat. Applications are however structured. For instance, [25] presented a three-dimensional structure of schemata with the application dimension, the volatile workflow data change recording dimension, and the metadata dimension. Additionally one might think of the user involvement dimension within a schema.

The ER modelling language has nowadays also been aging. Object-relational DBMS support features that should be representable at the conceptual level due to their utility. Modern technology provides user defined types, identification trees for components of relationship types and subtypes with overwriting by new surrogate types, flexible view-oriented handling of integrity constraints, indexing mechanism, maintenance of data blocks etc. These features are not representable in the classical ER modelling language but useful for conceptualisation.

The modelling process is far more dogmatic then understood in cookbooks or textbooks. It is somehow ritual or routine-based in education as well in practice. Models are mainly shallow models. They represent a part of the application. The specifics of constraints are not well applied. For instance, cardinality constraints specify the extremal minimal/maximal cases. Users however concentrate on the normal situation.

3.4 Views: The Overlooked Element of Conceptual Modelling

One of the limitations of the ER matrix is neglecting user views and viewpoints we consider now. The misunderstanding of view and viewpoints causes a small crisis in understanding database technology and modelling. The crisis has manifested in the development of data warehouses, data marts, star and snowflake schemata. The latter schemata or else than conceptual or business user views on the database. Currently the approach falls into a lot of difficulties and resulted in the development of a 'novel' technology[8].

---

[7] As shown in [59], the extended ER language HERM [56] covers 5 of 6 where the last sixth category center-periphery can be represented on the basis of HERM views.

[8] A similar trillion $ mistake is now the evolution of big data. The hype will shred a lot of resources until a new consensus occurs. In order to understand we remember

The three-layer architecture of information systems is a commonly accepted and widely taught conception. It is however neither true nor useful. It is only a starting point for understanding how a system might work. In reality, user viewpoints come with the application or business user level. They are represented by user schemata. Then these viewpoint schemata are integrated into the conceptual schema in the 'global-as-design' and 'local-as-view' approach. In order to represent the viewpoints we should use view schemata at the conceptual level what is however not consensus [67]. The logical level turns all the user view schemata to view definitions with the loss of the association between the relational views due to the limitations of relational technology. So far, this is the current state-of-the-art. It is nothing else than an anomaly since this schematology repeatedly resisted solution.

It could be improved with the approach developed in [20]. [67] defines a conceptual model to consist of a conceptual schema and of a collection of conceptual views that are associated (in most cases tightly by a mapping facility) to the conceptual schema. A conceptual schema is then mapped to a collection of logical views.

A database schema could not be anymore seen as an integrated, holistic schema with the same level of detail. Instead, we are able to represent a number of viewpoints at different abstraction level, with different foci and scopus, with different aging and currency, with supporting mechanisms depending on currency requirements, etc. So, the database structure model forms some kind of 'web' [23] instead of one schema with derived views. Viewpoints represent structures in whatever order is best for human comprehension and thus expressing it in a stream of consciousness order.

4 The Modelling Matrix

4.1 Deep Models and Scenarios Form the Modelling Matrix

T. Kuhn [29] widely used the notion of paradigm in a variety of forms and explanations. Essentially his notions can be understood as a disciplinary matrix [32], i.e. a symbolic generalisation, a meta-model, and collection of sample cases. Based on the observations on stereotyped modelling we may distinguish four initialisation phases:

(i) orientation on modelling scenarios and used macro-models for development with derivation of the function (and thus purpose and goal) a model has;

---

the discussions about the relational data model in panels at ER92, ER93, ER95, and ER95. The new consensus was relatively undeveloped. It was not able to represent all modelling situations the network or hierarchical data models could. Later it was realised that the different modelling styles could not been judged on a common scale. All three approaches have some shared habits and ways of seeing things. Proponents of these different approaches tended to talk past each other. Dogmatism and idiosyncrasy function in a complex social arrangement [28] such as conferences and journals.

(ii) acceptance of the grounding and of language and the general concept space;
(iii) setting of a deep model as the hidden, latent model or acceptance of such for some context and a community of practice;
(iv) acquisition of origins for modelling.

Definition 2 *The* deep model *consists of the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense) of modelling.*

The deep model thus uses a collection of undisputable elements of the backgreound as grounding and additionally a disputable and adjustable basis which is commonly accepted in the given context by the community of practice. It is typically used for many normal models but not explicitly stated whenever a normal model has been stated. The deep model is far more dogmatic than often understood. It is some kind of model 'logos' behind the normal model.

At the same time, the deep model is a rich source of knowledge [32, 34] that is already provided by the deep model, i.e. the deep model carries the knowledge and beliefs as well as the culture of the community of practice. It supports communication within the community of practice that accepts the deep model as common ground and has already agreed on the judgements made for the deep model. This common background also includes a common ontology. The deep model provides an identity within this community for the shared 'correct' opinion. The normal model becomes an epistemic instrument that is based on the common ground.

Definition 3 *The* modelling matrix *consists of the deep model and the modelling scenarios. The agenda is derived from the modelling scenario and the scenarios.*

So, the modelling scenario and the deep model serve as a part of the definitional frame within a model development process. They define also the capacity and potential of a model whenever it is utilised. The normal model can be deployed in a specific form as long as the scenarios and the deep model are not changed. For instance, database structure modelling on the basis of the entity-relationship approach has an ordinary interpretation for all developed schemata.

Different matrixes solve different problems. It might happen that a normal model with one matrix does not make sense if the matrix is changed. A typical case is co-modelling is modelling on the basis of the entity-relationship modelling language for structures and on the basis of BPMN diagrams for processes [68].

Models typically represent a number of origins. It is often the case that these origins use a common application-specific concept space, e.g. an application ontology with its lexicology and lexicography [59]. The application-specific concept space is annotated by a namespace.

A modelling matrix may be enhanced by generic or reference models. Generic models are abstractions of a set of models that represent similar solutions. They are later tailored to suit the particular purpose and function. A generic generally represents origins under interest, provides means to establish adequacy and dependability of the model, and establishes focus and scope of the model. A reference model is used as a blueprint for a fully fledged model and provides a general solution in an application area..

4.2 Adequacy and Dependability Governed by the Modelling Matrix

The modelling matrix allows to derive specific pattern for specification of adequacy and dependability of a model. The general notion of the model in Definition 1 defines adequacy based on an analogy property, a focus property, and purposefulness. Dependability is based on a justification and a quality certificate. Justification is given by an empirical corroboration according to modelling objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability, and by stability and plasticity within a collection of origins. Quality can be defined by characteristics that state the internal and the external quality as well as the quality in use. The certificate is the result of an evaluation of an accepted bundle of quality characteristics through some evaluation procedure.

It seems that the statement of adequacy and dependability is a heavy and sumptuous procedure. In reality it is far more simpler due to the existence of a modelling matrix. The entity-relationship modelling matrix uses, for instance, a homomorphism or infomorphism mapping property for analogy, a focus that is already determined by the situation or perception models or other origins, the purpose of full representation within the language setting, an empirical corroboration due to the mapping from the situation or perception model or other origins, the conformity that is already inherent in the modelling matrix, falsifiability via validation of the model against the origins, and stability against origins as a general class of situation and perception models or other origins. A similar definitional frame can be observed for many model kinds in Computer Science and Engineering.

As already discussed in Section 3.1, the matrix of entity-relationship modelling is quite comprehensive. We are explicitly explicitly using this matrix in dependence on the scenario. For instance, in system construction scenarios: closed-world schemata, Salami slice schemata, methods for simple transformation; adequate for direct incorporation; hierarchical schemata; separation of syntax and semantics; tools with well-defined semantics; viewpoint derivation; componentisation and modularisation; integrity constraint formulation support; methods for integration, variation. In communication scenarios we orient the matrix to: viewpoint and flavour representation; flexible usage (full logical independence); variable name space representation; methods for reasoning, understanding, presentation, exploration; methods for explanation, check, appraise, experience

This orientation governs the well-formedness criteria such as[9]:

- unambiguous esp. for transformation,
- easy to read,
- aspect-separated, e.g. by colouring different parts,
- naming styles, e.g. either singular or plural,
- higher normal forms,
- optional structure routed to the subtype,
- freeness of semantical cycles,
- distinguishability of attributes, e.g. unique name assumption,
- meaningful names, avoidance of auxiliary verbs, e.g. ‘has’, ‘is’, ...,
- non-empty classes, and
- flag avoidance.

Central characteristics for well-formed schema are: closed world and unique name assumptions; concept enhancement and well-defined name space; no sharpening or contrasting; well-founded logics; layering of functionality, views and interaction.

The adequacy of eER schemata is based on the following properties for origins $\mathfrak{A}$ and for the scenario $\mathfrak{S}$:

1. $\mathfrak{A}$-analogous: structural analogy (homomorphic, but not qualitative, functional) resulting in structural alignment; metaphysical, epistemological and heuristic adequacy
2. $\mathfrak{A}$-reduced (or $\mathfrak{A}$-focused): compactness, no repetition, high-level descriptive abstraction; conceptual minimal
3. $\mathfrak{S}$-purposeful: either for construction of another representation (thus with construction hints and tactics; with simple transformation; normalised, simple integrity enforcement) or for communication with the (business) user (thus with different viewpoints and flavours; simple viewpoints; cognitive complete).

The focus of eER Schemata is based on the following characteristics:

- Separation into kernel object types, dependent types, and properties: Kernel objects have their own relative existence independence.
- Kernel object types and typical/central types become entity types; properties may be complex and are typically mapped to (complex) attribute types; hierarchies are separated and then represented by generalisation/specialisation hierarchies; relationship types are either application association types, user-relating types, meta-associations, or workflow hocks. This is similar to good practices for E/R/A/C mappings.
- All derivable constructs are represented otherwise. Irrelevant, specific elements are avoided.
- The schema concentrates on important, relevant and typical elements.

---

[9] For details we refer to classical database design books [36, 6, 51, 55, 56]

- The schema must be as simple as possible, avoid unnecessary abstractions, provide a precise meaning for each type, reduce any complexity on Salami slice techniques, and should combine similar elements.
- Rigid incremental schema.

The purposefulness of eER Schemata is given be the following orientation in dependence on the purpose:

Orientation model: corporate overview as a context, data as a source and sink, “environment” model;

Communication model: external schemata depending on the context;

Conceptual model: things of significance, concepts, assertions; semantic model for the business language (“divergent”); architectural model (general categories, “convergent”, platform independent);

Realisation model: based on technology and platform; internal (logical) schemata (platform-specific: relational, XML, ...) (with technological twists), physical schema (storage, (vendor-specific); and

Documentation model of ground structures used in a given application system.

Scenarios typically combine a number of functions for the ER models. So we might use several schemata and especially view(ppoint) schemata as a model suite [57].

Dependability of eER schemata and models is defined by:

1. Justification is based on embedding the model into the understanding of the application domain, i.e. through an external corroboration. The internal corroboration is based on the language. The origins determine items of the model. So, no additional acquisition and elicitation is needed. The model conforms to standards accepted in CoP, e.g. Salami slice tactics; correctness; restriction to essential business items; approved, closed world schemata; partially evolution prone, partial flexibility; simple diagramming with overlay diagrams.
   The model should be cognitive complete based on an appropriate representation of things of interest in real world with some ordering (e.g. hierarchies (up-down, front-back)) and additionally based on other cognitive dimensions (container, part-whole, link, centre-periphery, source-path-goal) [59]. The model is a deputy of relationships of interest in the real word with some ordering additionally based on other cognitive dimensions. We might use additional characteristics of interest for both sides [24, 41, 13].
2. Sufficiency is defined by an evaluation form and by characteristics for internal and external quality and quality of use. Typical criteria are [56]: completeness, naturalness, minimality, system independence, flexibility, self-explanation, ease of reading and using of firm quality and evaluated. We mainly use quality in use characteristics without any error tolerance. Additionally, we assume (a) avoidance of redundancy (or at least restriction to necessary (controlled)) , (b) avoidance imposed implementation restrictions, (c) internal and external characteristics for the usage of the model

as blueprint without requirement for completeness of constraint sets, (d) natural keys, (d) avoidance of mega-attributes, and (e) complete confidence in all model components.

We notice, that most of the adequacy and dependability characteristics are assumed to be given with any eER schema or model. They are not mentioned but assumed. So, they are a part of the matrix.

A similar definitional frame can be defined for BPMN and other workflow diagrams.

Specific definitional frames are used for adequacy and dependability statements for models

· which provide specific extensions as an *amplification* which are not observed in the origins,
· which are *distortions* and are used for improving the origins (e.g. the physical world) or for inclusion of visions of better reality, e.g. for construction via transformation or in *Galilean models*, and
· which are *idealisations* through abstraction from origins by scoping the model to the ideal state of affairs.

Therefore, the modelling matrix allows to reduce and to simplify the statement whether a model is adequate and dependable. The reduction also stems from the definitional frame that is already used for the deep model.

4.3 Development of the Normal Model and the Matrix

Education and practice in modelling typically starts with acceptance of a matrix. Whether this matrix is adequate or not is not questioned. So, we can use the modelling matrix and define the specific model in dependence on a function or purpose. Let us now consider, revise and extend the model notions in [67].

Definition 4 *The* normal conceptual eER database structure model for communication and negotiation *comprises the database schema, reflects viewpoints and perspectives of different involved parties and their perception models. The matrix for communication scenario implicitly links to (namespaces or) concept fields of parties which are partially used. It defines adequacy and dependability based on the association of the perception models to viewpoints and of the viewpoints with the schema. A partial communication model does not use a schema and does not associate viewpoints to schema elements.*

As already observed in [67], normal models used for *communication and negotiation* follow additional principles: Viewpoints and specific semantics of users are explicitly given. The normal model is completely logically independent from the platform for realisation. The name space is rather flexible. The normal model is functioning and effective if methods for reasoning, understanding, presentation, exploration, explanation, validation, appraisal and experimenting are attached.

Definition 5 *The* normal conceptual eER database structure model for conceptualisation *consists of a collection of views for support of business users. The* deep model *is based on a mapping for schema elements that associates potential elements of the normal model to the common concept field and the perception models of business users. They may be extended by a skeleton that combine business user viewpoints or by a global schema which combines these viewpoints. The matrix uses a strict adequacy and dependability. It is based on a context-driven conceptualisation of the application domain.*

Conceptualisation is based on one or more concept or conception spaces of business users. Semantics is typically rather flexible. The normal model and the viewpoint rather reflect the normal cases and do not extend these cases to the extremal cases, e.g. for (cardinality) constraints.

The deep model and the normal model for description can be defined in a similar way. They are representations, refinements and amplifications [58, 63] of situation or reality models and therefore refinements and extensions of the communication model.

Definition 6 *The* normal conceptual database structure model for description *comprises the database schema, and a collection of views for support of business users. The model reflects a collection of a commonly accepted reality models that reflects perception or situation models with explicit association to views, and a shallow declaration of model adequacy and dependability. The deep model and the matrix are driven by the description scenario and completely bound to the understanding in the application area and to technology, methodology and theory which is commonly agreed within the community of practice.*

The descriptive normal model reflects the origins and abstracts from reality by scoping the model to the normally considered state of affairs. The deep model also provides an idealisation.

Prescriptive models that are used for system construction are filled with anticipation of the envisioned system. They deliberately diverge from reality in order to simplify salient properties of interest, transforming them into artifacts that are easier to work with.

Definition 7 *The* normal conceptual database structure model for prescription *comprises the database schema and a collection of views for both support of business users and system operating. It is based on a deep model that provides a number of a realisation templates according to the platform capabilities. The matrix declaration of model uses strict adequacy and dependability.*

The matrix also defines directives (or pragmas) [2] and transformation parameters [56] . The deep model also consists of general descriptions or templates for realisation style and tactics, for configuration parameters (coding, services, policies, handlers), for generic operations, for hints for realisation of the database, for performance expectations, for constraint enforcement policies, and for support features for the system realisation.

These notions of normal models, deep models, and matrices specialise general notions like those given in the introduction or the notion by W. Steinmüller

(*"A model is information: on something (content, meaning), created by someone (sender), for somebody (receiver), for some purpose (usage context).*)[53] or B. Mahr (*" A model is always at the same time a model 'of something' and a model 'for something'. Its function is to 'carry' some 'cargo' from its 'matrix' to its 'applicate'."*)[35] or F. Matthes and J. Schmidt (*"A relational database model on the basis of the approach by E.F. Codd describes semantics of declarations and statements within a database specification language and thus corresponds to an abstract model of a programming language with its static and dynamic semantics which can be specified through formal type and evaluation rules."*)[38].

Modelling is often 'brownfield' work. The model exists already and has been developed based on another matrix. Consider, for instance the schemata in [15, 18, 37, 50, 49, 52]. The schemata follow a certain matrix, e.g. in this case IDEF. Therefore, typical applications combine a number of matrices.

5 Conclusion

5.1 Model $\equiv$ Normal Model $\bowtie$ Matrix

A model can thus be understood as a normal model combined with a matrix and especially its deep model similar to the visible (or surface) and invisible parts of an iceberg. The matrix forms a relatively stable component for a larger collection of models and can be thus neglected for these models. The matrix is considered to be valid and does not thus need a justification. This observation led us to the conclusion that modelling is mainly normal modelling.

5.2 The Model Matrix as the Stabile Ground of Normal Models

We observed that models consist of a normal model and its matrix (or a number of its matrices). The matrix is neither questioned nor a matter of redefinition in a modelling process. It is taken for granted. A special case are 'brownfield' models which have a legacy matrix and a current matrix and which may consist of a model suite of mutual models for each of the matrices.

A matrix may evolve as well due to its limitations, revisions of dependability and adequacy required for an application, misconceptions, or missing elements. In our area, we observe changes of the deep model only for cases when technology entirely changes, e.g. the transfer from network or hierarchical modelling languages to the relational ones. The relational environment has changed however as well. So far, it is at its best an evolution step for matrices if at all. Database structure modelling has not changed for more than two decades although technology has changed a lot. Matrix evolution is also caused by changes in the scenarios.

Matrices are relatively stable. Normal models are under continuous change also due to rational and empirical evaluation or due to quality problems, e.g.

validity & completeness, reliability & coherence, and conformity & correspondence. Therefore, normal model evolution is mainly based on a stabile ground, i.e. a stabile matrix.

One advantage of such stabile grounds is the potential for accumulation and maturation of normal model development and utilisation. It enables knowledge elicitation and acquisition used in design science [72]. It is then part of the rigor cycle.

A simple form of matrix evolution is the combination of scenarios into a coherent set of scenarios. This combination or adduction allows to combine the matrices into holistic ones. The deep models are then typically model suites. A specific form of matrix evolution is consolidation of the matrix, for instance, by development of supporting theories and by maturing methodologies. In this case, the normal models can still be used in the same form.

5.3 Model Notions for Normal Models

We may now elaborate the notions in the introduction. It seems to be obvious how these notions match to our understanding of normal models and their matrices. So, let us consider two additional examples.

An Example for Database IT Practice. [42] considers the mediator/-communication scenario. The model is used for "the representation of some aspects" of the situation model, "enables clearer communication" about the situation model, and "serves as a blueprint to shape and to construct the proposed structures" in the situation model. So, a *normal* "data model is a device that

– helps the users or stakeholders understand clearly the database system that is being implemented based on the information requirements of an organization, and

– enables the database practitioners to implement the database system exactly conforming to the information requirements."

This notion of the model is determined by the given two scenario, by the deep model of database models, by the community of business users and database developers, by data engineering and DBMS as its context.

Models for Domain resp. Software Engineering. An application domain is a universe of discourse, an area of human activity or an area of science. Domain engineering is understood as modelling: "a careful description of the domain as it is, void of any reference to possibly desired new software, including requirements to new software". [10] "By a domain theory we understand a formal model of a domain such that properties of the model the domain can be stated and formally verified - claiming that these properties are properties of the domain being modelled." "A domain model is thus a description of a sufficient number of domain entities, domain functions, domain events and domain behaviours - so formulated and detailed that one is able to answer most relevant questions about the domain."

The deep model is partially explicitly given as a domain theory. The implicit part is, for instance, the notion of an application domain, the focus to description and the functions of the model, the underlying mathematical theory, the modelling language (entities, ...), and the way of associating. All this forms the matrix of the "domain model".

A similar observation can be made for classical software engineering, e.g. [26, 65].

5.4 Modelling from Art to Science

Modelling is still considered to be an art. It will become a science in future in the understanding of [8]. Moving paths are thus: from practices to principles, from skilled performance to fundamental recurrences, from action to explanation, from invention to discovery, from synthesis to analysis, and from construction to dissection. Modelling as science is organised to understand, exploit and cope with an application. It encompasses natural and artificial aspects of the application. It codifies the body of knowledge mainly n the basis of deep models and matrices. It will have a commitment to normal models for discovery and validation. Models will thus become reproducible. Modelling is enhanced by falsifiability, testing, validation and verification. Modelling as a science has the ability to make reliable predictions, some of them might be surprising. Modelling might also be based on other techniques than presented in this paper. All models in [66] have their matrix. So, modelling based on normal models with their matrix will still be one of the main forms of modelling culture.

References

1. Websters ninth new collegiate dictionary, 1991.
2. ISO/IEC JTC 1/SC 22. Information technology – Programming languages – C. ISO/IEC 9899:2011, Stage: 60.60, 2011.
3. D. Abts and W. Mülder. *Grundkurs Wirtschaftsinformatik : Eine kompakte und praxisorientierte Einführung.* Vieweg, 2004.
4. P. Alpar. *Computergestützte interaktive Methodenauswahl.* PhD thesis, Frankfurt Main Univ., 1980.
5. C. W. Bachman. Impact of object oriented thinking on ER modeling. LNCS 1157, pages 1–4. Springer, Berlin, 1996.
6. C. Batini, S. Ceri, and S. Navathe. *Conceptual database design (an entity-relationship approach).* Benjamin/Cummings, Redwood City, 1992.
7. J. Becker and R. Schütte. *Handelsinformationssysteme : Domänenorientierte Einführung in die Wirtschaftsinformatik.* Moderne Industrie, 2004.
8. M. Bichler, U. Frank, D. Avison, J. Malaurent, P. Fettke, D. Hovorka, J. Krämer, D. Schnurr, B. Müller, L. Suhl, and B Thalheim. Theories in business and information systems engineering. *Business & Information Systems Engineering*, pages 1–29, 2016.

9. A. Bird. *Thomas Kuhn.* Routledge, 2014.
10. D. Bjørner. *Domain engineering*, volume 4 of *COE Research Monographs.* Japan Advanced Institute of Science and Technolgy Press, Ishikawa, 2009.
11. S. Chadarevian and N. Hopwood, editors. *Models - The third dimension of science.* Stanford University Press, Stanford, California, 2004.
12. P. P. Chen. The entity-relationship model: Toward a unified view of data. *ACM TODS*, 1(1):9–36, 1976.
13. Ferdinand de Saussure. *Cours de Linguistique Générale.* Payot, 1995.
14. D. Embley and B. Thalheim, editors. *The Handbook of Conceptual Modeling: Its Usage and Its Challenges.* Springer, 2011.
15. M. Fowler. *Refactoring.* Addison-Wesley, Boston, 2005.
16. P. Godfrey-Smith. Models and fictions in science. *Philosophical studies*, 143(1):101–116, 2009.
17. J. Gray. eScience: A transformed scientific method. Technical report, Talk given Jan 11, 2007. Edited by T. Hey, S. Tansley, and K. Tolle. http://research.microsoft.com/en-us/um/people/gray/talks/NRC-CSTB_eScience.ppt, Microsoft Research Publications, 2007.
18. D. C. Hay. *Data model pattern: Conventions of thought.* Dorset House, New York, 1995.
19. T. Illenseer. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, chapter Astrophysikalische Modellbildung am Beispiel aktiver galaktischer Kerne, pages 251–280. De Gryuter, Boston, 2015.
20. H. Jaakkola and B. Thalheim. Architecture-driven modelling methodologies. In *Information Modelling and Knowledge Bases*, volume XXII, pages 97–116. IOS Press, 2011.
21. R. Kaschek. *Konzeptionelle Modellierung.* PhD thesis, University Klagenfurt, 2003. Habilitationsschrift.
22. H.-J. Klein. Null values in relational databases and sure information answers. In L. Bertossi et al., editor, *Semantics in Databases*, pages 102–121, Dagstuhl Castle, December 2002. Springer-Verlag. LNCS 2582.
23. Donald E. Knuth. *Literate Programming.* Number 27 in CSLI Lecture Notes. Center for the Study of Language and Information at Stanford/ California, 1992.
24. B. Kralemann and C. Lattmann. Models as icons: modeling models in the semiotic framework of peirce's theory of signs. *Synthese*, 190(16):3397–3420, 2013.
25. F. Kramer and B. Thalheim. Component-based development of a metadata data-dictionary. In *BIS*, volume 176 of *Lecture Notes in Business Information Processing*, pages 110–121. Springer, 2014.
26. J. Krogstie. *Model-based development and evolution of information systems.* Springer, 2012.
27. T. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, Chicago, Illinois, 1962.
28. T. Kuhn. Objectivity, value judgment, and theory choice. *Arguing About Science*, pages 74–86, 1977.
29. T. Kuhn. *Die Entstehung des Neuen.* Surkamp, Frankfurt a. M., 1978.
30. I. Lakatos. Falsification and the methodology of scientific research programmes. *Criticism and the Growth of KnowledgeÄ_Ä, ed. I. Lakatos and A. Musgrave (London: Cambridge University Press, 1974)*, 2014.
31. H.-J. Lenz and B. Thalheim. OLAP databases and aggregation functions. In *13th SSDBM 2001*, pages 91 – 100, 2001.

32. B. Mahr. *Das Wissen im Modell.* Technische Universität, Fakultät IV Berlin, 2004.
33. B. Mahr. Intentionality and modeling of conception. In *Judgements and Propositions. Logical, Linguistic, and Cognitive Issues.* Logos, 2010.
34. B. Mahr. *Mathesis & Graphé: Leonhard Euler und die Entfaltung der Wissenssysteme*, chapter Denken in Modellen, pages 85–100. Akademie-Verlag, 2010.
35. B. Mahr. Modelle und ihre Befragbarkeit - Grundlagen einer allgemeinen Modelltheorie. *Erwägen-Wissen-Ethik (EWE)*, Vol. 26, Issue 3:329–342, 2015.
36. H. Mannila and K.-J. Räihä. *The design of relational databases.* Addison-Wesley, Wokingham, England, 1992.
37. D. Marco and M. Jennings. *Universal meta data models.* Wiley Publ. Inc., 2004.
38. F. Matthes and J. W. Schmidt. *Datenbankhandbuch*, chapter Datenbankmodelle und Datenbanksprachen, pages 1–154. Springer Berlin, 1987.
39. M.S. Morgan and M. Morrison, editors. *Models as mediators.* Cambridge Press, 1999.
40. P.P. Nayak. *Automated modeling of physical systems.* Springer, LNCS 1003, Berlin, 1995.
41. C.S. Peirce. What is a sign? In Peirce Edition Project, editor, *The essential Peirce: selected philosophical writings*, volume 2, pages 4 – 10. Indiana University Press, Bloomington, Indiana, 1998.
42. P. Ponniah. *Data modeling fundamentla.* John Wiley & Sons, Hoboken, 2007.
43. K. R. Popper. *Logik der Forschung.* J.C.B. Mohr (Paul Siebeck), Tübingen, 10th. edition, 1994.
44. K. R. Popper. Subjektive oder objektive Erkenntnis? In David Miller, editor, *Karl Popper Lesebuch*, pages 40–59. J.C.B. Mohr (Paul Siebeck), Tübingen, 1994/1995.
45. J.E. Safra, I. Yeshua, and et. al. *Encyclopædia Britannica.* Merriam-Webster, 2003.
46. K.-D. Schewe and B. Thalheim. NULL value algebras and logics. In *Information Modelling and Knowledge Bases*, volume XXII, pages 354–367. IOS Press, 2011.
47. K.-D. Schewe and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter Co-Design of Web Information Systems, pages 293–332. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.
48. B. Scholz-Reiter. *Konzeption eines rechnergesttzten Werkzeugs zur Analyse und Modellierung integrierter Informations- und Kommunikationssysteme in Produktionsunternehmen.* PhD thesis, TU Berlin, 1990.
49. L. Silverston. *The data model resource book. Revised edition*, volume 2. Wiley, 2001.
50. L. Silverston, W. H. Inmon, and K. Graziano. *The data model resource book.* John Wiley & Sons, New York, 1997.
51. G. Simsion. *Data modeling essentials - Analysis, design and innovation.* Van Nonstrand Reinhold, New York, 1994.
52. G. Simsion and G.C. Witt. *Data modeling essentials.* Morgan Kaufmann, San Francisco, 2005.
53. W. Steinmüller. *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik.* Wissenschaftliche Buchgesellschaft, Darmstadt, 1993.
54. V. Storey and B. Thalheim. Conceptual modeling: Enhancement through semiotics. In *Proc. ER'17*, LNCS, accepted for publication, forthcoming, Cham, 2017. Springer.
55. T. J. Teorey. *Database modeling and design: The entity-relationship approach.* Morgan Kaufmann, San Mateo, 1989.

56. B. Thalheim. *Entity-relationship modeling – Foundations of database technology.* Springer, Berlin, 2000.
57. B. Thalheim. *The Conceptual Framework to Multi-Layered Database Modelling based on Model Suites*, volume 206 of *Frontiers in Artificial Intelligence and Applications*, pages 116–134. IOS Press, 2010.
58. B. Thalheim. 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.`
59. B. Thalheim. Syntax, semantics and pragmatics of conceptual modelling. In *NLDB*, volume 7337 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2012.
60. B. Thalheim. The conception of the model. In *BIS*, volume 157 of *Lecture Notes in Business Information Processing*, pages 113–124. Springer, 2013.
61. B. Thalheim. Open problems of information systems research and technology. In *Invited Keynote, BIR'2013. LNBIB 158*, pages 10–18. Springer, 2013.
62. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
63. B. Thalheim. Models, to model, and modelling - Towards a theory of conceptual models and modelling - Towards a notion of the model. Collection of recent papers, http://www.is.informatik.uni-kiel.de/∼thalheim/indexkollektionen.htm, 2014.
64. B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In *Encyclopedia of Database Systems.* 2017.
65. B. Thalheim. General and specific model notions. In *Proc. ADBIS'17*, LNCS, accepted for publication, forthcoming, Cham, 2017. Springer.
66. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung.* De Gruyter, Boston, 2015.
67. B. Thalheim and M. Tropmann-Frick. The conception of the conceptual database model. In *ER 2015*, LNCS 9381, pages 603–611, Berlin, 2015. Springer.
68. B. Thalheim and M. Tropmann-Frick. Models and their capability. In C. Beierle, G. Brewka, and M. Thimm, editors, *Computational Models of Rationality*, volume 29 of *College Publications Series*, pages 34–56. College Publications, 2016.
69. O. Thomas. Das Modellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation. Technical Report Heft 184, Institut für Wirtschaftsinformatik, DFKI, Saarbrücken, Mai 2005.
70. B. F. Webster. *Pitfalls of object-oriented development: a guide for the wary and entusiastic.* M&T books, New York, 1995.
71. S. Wenzel. Referenzmodell fr die Simulation in Produktion und Logistik. *ASIM Nachrichten*, 4(3):13–17, 2000.
72. R. Wieringa. *Design science methodology for information systems and software engineering.* Springer, Heidelberg, 2014.

Data Mining Design and Systematic Modelling

Yannic Kropp Bernhard Thalheim
Christian Albrechts University Kiel, Department of Computer Science, D-24098 Kiel, Germany
yk@is.informatik.uni-kiel.de thalheim@is.informatik.uni-kiel.de

Abstract. Data mining is currently a well-established technique and supported by many algorithms. It is dependent on the data on hand, on properties of the algorithms, on the technology developed so far, and on the expectations and limits to be applied. It must be thus matured, predictable, optimisable, evolving, adaptable and well-founded similar to mathematics and SPICE/CMM-based software engineering. Data mining must therefore be systematic if the results have to be fit to its purpose. One basis of this systematic approach is model management and model reasoning. We claim that systematic data mining is nothing else than systematic modelling. The main notion is the notion of the model in a variety of forms, abstraction and associations among models.

Keywords: data mining, modelling, models, framework, deep model, normal model, modelling matrix

1 Introduction

Data mining and analysis is nowadays well-understood from the algorithms side. There are thousands of algorithms that have been proposed. The number of success stories is overwhelming and has caused the big data hype. At the same time, brute-force application of algorithms is still the standard. Nowadays data analysis and data mining algorithms are still taken for granted. They transform data sets and hypotheses into conclusions. For instance, cluster algorithms check on given data sets and for a clustering requirements portfolio whether this portfolio can be supported and provide as a set of clusters in the positive case as an output. The Hopkins index is one of the criteria that allow to judge whether clusters exist within a data set. A systematic approach to data mining has already been proposed in [3, 17]. It is based on mathematics and mathematical statistics and thus able to handle errors, biases and configuration of data mining as well. Our experience in large data mining projects in archaeology, ecology, climate research, medical research etc. has however shown that ad-hoc and brute-force mining is still the main approach. The results are taken for granted and believed despite the modelling, understanding, flow of work and data handling pitfalls. So, the results often become dubious.

Data are the main source for information in data mining and analysis. Their quality properties have been neglected for a long time. At the same time, modern data management allows to handle these problems. In [16] we compare the critical findings or pitfalls of [21] with resolution techniques that can be applied to overcome the crucial pitfalls of data mining in environmental sciences reported there. The algorithms themselves are another source of pitfalls that are typically used for the solution of data mining and analysis tasks. It is neglected that an algorithm also has an application area, application restrictions, data requirements, results at certain granularity and precision. These problems must be systematically tackled if we want to rely on the results of mining and analysis. Otherwise analysis may become misleading, biased, or not possible. Therefore, we explicitly treat properties of mining and analysis. A similar observation can be made for data handling.

Data mining is often considered to be a separate sub-discipline of computer engineering and science. The statistics basis of data mining is well accepted. We typically start with a general (or better generic) model and use for refinement or improvement of the model the data that are on hand and that seem to be appropriate. This technique is known in sciences under several names such as inverse modelling, generic modelling, pattern-based reasoning, (inductive) learning, universal application, and systematic modelling.

Data mining is typically not only based on one model but rather on a model ensemble or model suite The association among models in a model suite is explicitly specified. These associations provide an explicit form via model suites. Reasoning techniques combine methods from logics (deductive, inductive, abductive, counter-inductive, etc.), from artificial intelligence (hypothetic, qualitative, concept-based, adductive, etc.), computational methods (algorithmics [6], topology, geometry, reduction, etc.), and cognition (problem representation and solving, causal reasoning, etc.).

These choices and handling approaches need a systematic underpinning. Techniques from artificial intelligence, statistics, and engineering are bundled within the CRISP framework (e.g. [3]). They can be enhanced by techniques that have originally been developed for modelling, for design science, business informatics, learning theory, action theory etc.

We combine and generalize the CRISP, heuristics, modelling theory, design science, business informatics, statistics, and learning approaches in this paper. First, we introduce our notion of the model. Next we show how data mining can be designed. We apply this investigation to systematic modelling and later to systematic data mining. It is our goal to develop a holistic and systematic framework for data mining and

Proceedings of the XIX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2017), Moscow, Russia, October 10-13, 2017

analysis. Many issues are left out of the scope of this paper such as a literature review, a formal introduction of the approach, and a detailed discussion of data mining application cases.

2 Models and Modelling

Models are principle instruments in mathematics, data analysis, modern computer engineering (CE), teaching any kind of computer technology, and also modern computer science (CS). They are built, applied, revised and manufactured in many CE&CS sub-disciplines in a large variety of application cases with different purposes and context for different communities of practice. It is now well understood that models are something different from theories. They are often intuitive, visualizable, and ideally capture the essence of an understanding within some community of practice and some context. At the same time, they are limited in scope, context and the applicability.

2.1 The Notion of the Model

There is however a general notion of a model and of a conception of the model:

A **model** *is a well-formed, adequate, and dependable instrument that represents origins* [9, 29, 30].

Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilization scenarios.

A well-formed instrument is *adequate* for a collection of origins if it is *analogous* to the origins to be represented according to some analogy criterion, it is more *focused* (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its *purpose*.

Well-formedness enables an instrument to be *justified* by an *empirical corroboration* according to its objectives, by *rational coherence* and *conformity* explicitly stated through conformity formulas or statements, by *falsifiability* or *validation*, and by *stability* and *plasticity* within a collection of origins.

The instrument is *sufficient* by its *quality* characterization for internal quality, external quality and quality in use or through quality characteristics [28] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some *assurance evaluation* (tolerance, modality, confidence, and restrictions).

2.2 Generic and Specific Models

The general notion of a model covers all aspects of adequateness, dependability, well-formedness, scenario, functions and purposes, backgrounds (grounding and basis), and outer directives (context and community of practice). It covers all known so far notions in agriculture, archaeology, arts, biology, chemistry, computer science, economics, electro-technics, environmental sciences, farming, geosciences, historical sciences, languages, mathematics, medicine, ocean sciences, pedagogical science, philosophy, physics, political sciences, sociology, and sports. The models used in these disciplines are instruments used in certain scenarios.

Sciences distinguish between general, particular and specific things. Particular things are specific for general things and general for specific things. The same abstraction may be used for modelling. We may start with a general model. So far, nobody knows how to define general models for most utilization scenarios. Models *function* as *instruments* or tools. Typically, instruments come in a variety of forms and fulfill many different functions. Instruments are partially independent or autonomous of the thing they operate on. Models are however special instruments. They are used with a specific intention within a utilization scenario. The quality of a model becomes apparent in the context of this scenario.

It might thus be better to start with generic models. A ***generic model*** [4, 26, 31, 32] is a model which broadly satisfies the purpose and broadly functions in the given utilization scenario. It is later tailored to suit the particular purpose and function. It generally represents origins of interest, provides means to establish adequacy and dependability of the model, and establishes focus and scope of the model. Generic models should satisfy at least five properties: (i) they must be accurate; (ii) the quality of generic models allows that they are used consciously; (iii) they should be descriptive, not evaluative; (iv) they should be flexible so that they can be modified from time to time; (v) they can be used as a first “best guess”.

2.3 Model Suites

Most disciplines integrate a variety of models or a *society of models*, e.g. [7, 14] Models used in CE&CS are mainly at the same level of abstraction. It is already well-known for threescore years that they form a *model ensemble* (e.g. [10, 23]) or horizontal *model suite* (e.g. [8, 27]). Developed models vary in their scopes, aspects, and facets they represent and their abstraction.

A ***model suite*** consists of a set of models {M_1,..., M_n}, of an association or collaboration schema among the models, of controllers that maintain consistency or coherence of the model suite, of application schemata for explicit maintenance and evolution of the model suite, and of tracers for the establishment of the coherence.

Multi-modelling [11, 19, 24] became a culture in CE&CS. Maintenance of coherence, co-evolution, and consistency among models has become a bottleneck in development. Moreover, different languages with different capabilities have become an obstacle similar to multi-language retrieval [20] and impedance mismatches. Models are often loosely coupled. Their dependence and relationship is often not explicitly expressed. This problem becomes more complex if models are used for different purposes such as construction of systems, verification, optimization, explanation, and documentation.

2.4 Stepwise Refinement of Models

Refinement of a model to a particular or special model provides mechanisms for model transformation along the adequacy, the justification and the sufficiency of a model. Refinement is based on *specialization* for better suitability of a model, on *removal* of unessential elements, on *combination* of models to provide a more holistic view, on *integration* that is based on binding of model components to other components and on *enhancement* that typically improves a model to become more adequate or dependable.

Control of correctness of refinement [33] for information systems takes into account (A) a focus on the refined structure and refined vocabulary, (B) a focus to information systems structures of interest, (C) abstract information systems computation segments, (D) a description of database segments of interest, and (E) an equivalence relation among those data of interest.

2.5 Deep Models and the Modelling Matrix

Model development is typically based on an explicit and rather quick description of the 'surface' or *normal model* and on the mostly unconditional acceptance of a *deep model*. The latter one directs the modelling process and the surface or normal model. Modelling itself is often understood as development and design of the normal model. The deep model is taken for granted and accepted for a number of normal models.

The deep model can be understood as the common basis for a number of models. It consists of the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense) of modelling. It uses a collection of undisputable elements of the background as grounding and additionally a disputable and adjustable basis which is commonly accepted in the given context by the community of practice. Education on modelling starts, for instance, directly with the deep model. In this case, the deep model has to be accepted and is thus hidden and latent.

A (modelling) matrix is something within or from which something else originates, develops, or takes from. The matrix is assumed to be correct for normal models. It consists of the deep model and the modelling scenarios. The modelling *agenda* is derived from the modelling scenario and the utilization scenarios. The modelling scenario and the deep model serve as a part of the *definitional frame* within a model development process. They define also the capacity and potential of a model whenever it is utilized.

Deep models and the modelling matrix also define some frame for adequacy and dependability. This frame is enhanced for specific normal models. It is then used for a statement in which cases a normal model represents the origins under consideration.

2.6 Deep Models and Matrices in Archaeology

Let us consider an application case. The CRC 1266[1]

> *"Scales of Transformation – Human*
> *Environmental Interaction in Prehistoric and*
> *Archaic Societies"*

investigates processes of transformation from 15,000 BCE to 1 BCE, including crisis and collapse, on different scales and dimensions, and as involving different types of groups, societies, and social formations. It is based on the matrix and a deep model as sketched in Figure 1. This matrix determines which normal models can still be considered and which not. The initial model for any normal model accepts this matrix.

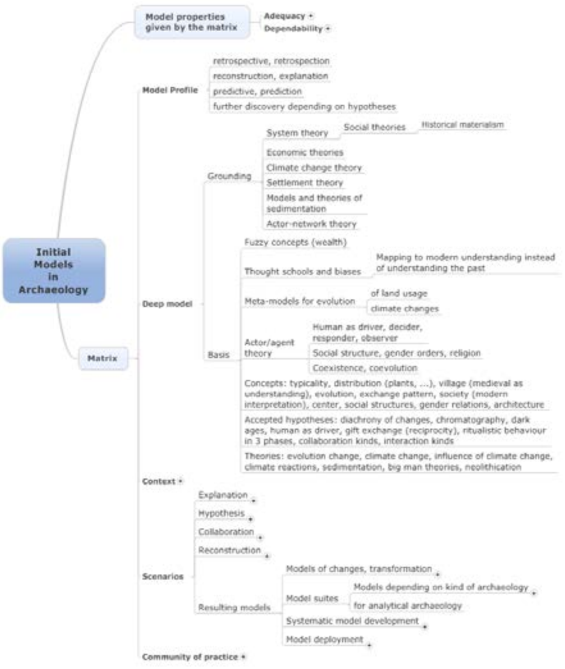


Figure 1 Modeling in archaeology with a matrix

We base our consideration on the matrix and the deep model on [19] and the discussions in the CRC. Whether the deep model or the model matrix is appropriate has already been discussed. The final version presented in this paper illustrates our understanding.

2.7 Stereotyping of a Data Mining Process

Typical modeling (and data mining) processes follow some kind of ritual or typical guideline, i.e. they are stereotyped. The *stereotype* of a modelling process is based on a general modelling situation. Most modelling methodologies are bound to one stereotype and one kind of model within one model utilization scenario.

---

[1] https://www.sfb1266.uni-kiel.de/en

Stereotypes are governing, conditioning, steering and guiding the model development. They determine the model kind, the background and way of modelling activities. They persuade the activities of modelling. They provide a means for considering the economics of modelling. Often, stereotypes use a definitional frame that primes and orients the processes and that considers the community of practice or actors within the model development and utilization processes, the deep model or the matrix with its specific language and model basis, and the agenda for model development. It might be enhanced by initial models which are derived from generic models in accordance to the matrix.

The model utilization scenario determines the function that a model might have and therefore also the goals and purposes of a model.

2.8 The *Agenda*

The agenda is something like a guideline for modeling activities and for model associations within a model suite. It improves the quality of model outcomes by spending some effort to decide what and how much reasoning to do as opposed to what activities to do. It balances resources between the data-level actions and the reasoning actions. E.g. [17] uses an agent approach with preparation agents, exploration agents, descriptive agents, and predictive agents. The agenda for a model suite uses thus decisions points that require agenda control according to performance and resource considerations. This understanding supports introspective monitoring about performance for the data mining process, coordinated control of the entire mining process, and coordinated refinement of the models. Such kind of control is already necessary due to the problem space, the limitations of resources, and the amount of uncertainty in knowledge, concepts, data, and the environment.

3 Data Mining Design

3.1 Conceptualization of Data Mining and Analysis

The data mining and analysis task must be enhanced by an explicit treatment of the languages used for concepts and hypotheses, and by an explicit description of knowledge that can be used. The algorithmic solution of the task is based on knowledge on algorithms that are used and on data that are available and that are required for the application of the algorithms. Typically, analysis algorithms are iterative and can run forever. We are interested only in convergent ones and thus need termination criteria. Therefore, conceptualization of the data mining and analysis task consists of a detailed description of *six main parameters* (e.g. for inductive learning [34]):

(a) The *data analysis algorithm*: Algorithm development is the main activity in data mining research. Each of these algorithms transfers data and some specific parameters of the algorithm to a result.

(b) The *concept space*: the concept space defines the concepts under consideration for analysis based on certain language and common understanding.

(c) The *data space*: The data space typically consists of a multi-layered data set of different granularity. Data sets may be enhanced by metadata that characterize the data sets and associate the data sets to other data sets.

(d) The *hypotheses space*: An algorithm is supposed to map evidence on the concepts to be supported or rejected into hypotheses about it.

(e) The *prior knowledge space*: Specifying the hypothesis space already provides some prior knowledge. In particular, the analysis task starts with the assumption that the target concept is representable in a certain way.

(f) The *acceptability and success criteria*: Criteria for successful analysis allow to derive termination criteria for the data analysis.

Each instantiation and refinement of the six parameters leads to specific data mining tasks.

The result of data mining and data analysis is described within the knowledge space. The data mining and analysis task may thus be considered to be a transformation of data sets, concept sets and hypothesis sets into chunks of knowledge through the application of algorithms.

Problem solving and modelling considers, however, typically six aspects [16]:

(1) *Application, problems, and users*: The domain consists of a model of the application, a specification of problems under consideration, of tasks that are issued, and of profiles of users.

(2) *Context*: The context of a problem is anything what could support the problem solution, e.g. the sciences' background, theories, knowledge, foundations, and concepts to be used for problem specification, problem background, and solutions.

(3) *Technology*: Technology is the enabler and defines the methodology. It provides [23] means for the flow of problem solving steps, the flow of activities, the distribution, the collaboration, and the exchange.

(4) *Techniques and methods*: Techniques and methods can be given as algorithms. Specific algorithms are data improvers and cleaners, data aggregators, data integrators, controllers, checkers, acceptance determiners, and termination algorithms.

(5) *Data*: Data have their own structuring, their quality and their life span. They are typically enhanced by metadata. Data management is a central element of most problem solving processes.

(6) *Solutions*: The solutions to problem solving can be formally given, illustrated by visual means, and presented by models. Models are typically only normal models. The deep model and the matrix is already provided by the context and accepted by the community of practice in dependence of the needs of this community for the given application scenario. Therefore, models may be the final result of a data mining and analysis process beside other means.

Comparing these six spaces with the six parameters we discover that only four spaces are considered so far in data mining. We miss the user and application space as well as the representation space. Figure 2 shows the difference.

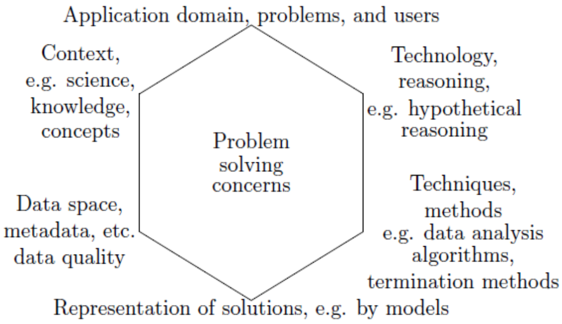


Figure 2 Parameters of Data Mining and the Problem Solving Aspects

3.2 Meta-models of Data Mining

An abstraction layer approach separates the application domain, the model domain and the data domain [17]. This separation is illustrated in Figure 3.

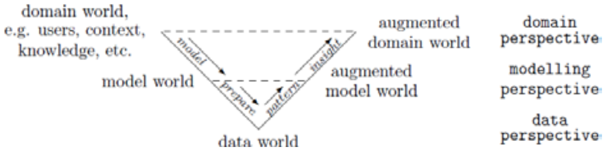


Figure 3 The V meta-model of Data Mining Design

The data mining design framework uses the inverse modeling approach. It starts with the consideration of the application domain and develops models as mediators between the data and the application domain worlds. In the sequel we are going to combine the three approaches of this section. The meta-model corresponds to other meta-models such as inductive modelling or hypothetical reasoning (hypotheses development, experimenting and testing, analysis of results, interim conclusions, reappraisal against real world).

4 Data Mining: A Systematic Model-Based Approach

Our approach presented so far allows to revise and to reformulate the model-oriented data mining process on the basis of well-defined engineering [15, 25] or alternatively on systematic mathematical problem solving [22]. Figure 4 displays this revision. We realize that the first two phases are typically implicitly assumed and not considered. We concentrate on the non-iterative form. Iterative processes can be handled in a similar form.

4.1 Setting the Deep Model and the Matrix

The problem to be tackled must be clearly stated in dependence on the utilization scenario, the tasks to be solved, the community of practice involved, and the given context. The result of this step is the deep model and its matrix. The first one is based on the background, the specific context parameter such as infrastructure and environment, and candidates for deep models.

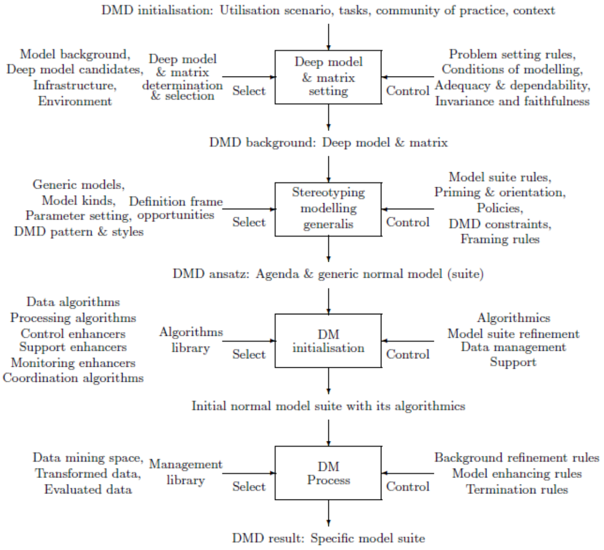


Figure 4 The Phases in Data Mining Design (Non-iterative form)

The data mining tasks can be now formulated based on the matrix and the deep model. We set up the context, the environment, the general goal of the problem and also criteria for *adequateness* and *dependability* of the solution, e.g. *invariance properties* for problem description and for the task setting and its mathematical formulation and *solution faithfulness properties* for later application of the solution in the given environment. What is exactly the problem, the expected benefit? What should a solution look like? What is known about the application?

Deep models already use a background consisting of an undisputable grounding and a selectable basis. The explicit statement of the background provides an understanding of the postulates, paradigms, assumptions, conceptions, practices, etc. Without the background, the results of the analysis cannot be properly understood. Models have their profile, i.e. goals, purposes and functions. These must be explicitly given. The parameters of a generic model can be either order or slave parameters [12], either primary or secondary or tertiary (also called genotypes or phenotypes or observables) [1, 5], and either ruling (or order) or driven parameters [12]. Data mining can be enhanced by knowledge management techniques.

Additionally, the concept space into which the data mining task is embedded must be specified. This concept space is enhanced during data analysis.

4.2 Stereotyping the Process

The general flow of data mining activities is typically implicitly assumed on the basis of stereotypes which form a set of tasks, e.g. tasks of prove in whatever system, transformation tasks, description tasks, and investigation tasks. Proofs can follow the classical deductive or inductive setting. Also, abductive, adductive, hypothetical and other reasoning techniques are applicable. Stereotypes typically use model suites as a collection of associated models, are already biased by priming and orientation, follow policies, data mining design constraints, and framing.

Data mining and analysis is rather stereotyped. For instance, mathematical culture has already developed a good number of stereotypes for problem formulation. It is based on a mathematical language for the formulation of analysis tasks, on selection and instantiation of the best fitting variable space and the space of opportunities provided by mathematics.

Data mining uses *generic models* which are the basis of normal models. Models are based on a separation of concern according the problem setting: dependence-indicating, dependence-describing, separation or partition spaces, pattern kinds, reasoning kinds, etc. This separation of concern governs the classical data mining algorithmic classes: association analysis, cluster analysis, data grouping with or without classification, classifiers and rules, dependences among parameters and data subsets, predictor analysis, synergetics, blind or informed or heuristic investigation of the search space, and pattern learning.

4.3 Initialization of the Normal Data Models

Data mining algorithms have their capacity and potential [2]. Potential and capacity can be based on SWOT (strengths, weaknesses, opportunities, and threats), SCOPE (situation, core competencies, obstacles, prospects, expectation), and SMART (how simple, meaningful, adequate, realistic, and trackable) analysis of methods and algorithms. Each of the algorithm classes has its strengths and weaknesses, its satisfaction of the tasks and the purpose, and its limits of applicability. Algorithm selection also includes an explicit specification of the order of application of these algorithms and of mapping parameters that are derived by means of one algorithm to those that are an input for the others, i.e. an explicit association within the model suite. Additionally, evaluation algorithms for the success criteria are selected. Algorithms have their own obstinacy, their hypotheses and assumptions that must be taken into consideration. Whether an algorithm can be considered depends on acceptance criteria derived in the previous two steps.

So, we ask: *What kind of model suite architecture suits the problem best? What are applicable development approaches for modelling? What is the best modelling technique to get the right model suite? What kind of reasoning is supported? What not? What are the limitations? Which pitfalls should be avoided?*

The result of the entire data mining process heavily depends on the appropriateness of the data sets, their properties and quality, and more generally the data schemata with essentially three components: application data schema with detailed description of data types, metadata schema [18], and generated and auxiliary data schemata. The first component is well-investigated in data mining and data management monographs. The second and third components inherit research results from database management, from data mart or warehouses, and layering of data. An essential element is the explicit specification of the quality of data. It allows to derive algorithms for data improvement and to derive limitations for applicability of algorithms. Auxiliary data support performance of the algorithms.

Therefore typical data-oriented questions are: *What data do we have available? Is the data relevant to the problem? Is it valid? Does it reflect our expectations? Is the data quality, quantity, recency sufficient? Which data we should concentrate on? How is the data transformed for modelling? How may we increase the quality of data?*

4.4 The Data Mining Process Itself

The data mining process can be understood as a coherent and stepwise refinement of the given model suite. The model refinement may use an explicit transformation or an extract-transform-load process among models within the model suite. Evaluation and termination algorithms are an essential element of any data mining algorithm. They can be based on quality criteria for the finalized models in the model suite, e.g. generality, error-proneness, stability, selection-proneness, validation, understandability, repeatability, usability, usefulness, and novelty.

Typical questions to answer within this process are: *How good is the model suite in terms of the task setting? What have we really learned about the application domain? What is the real adequacy and dependability of the models in the model suite? How these models can be deployed best? How do we know that the models in the model suite are still valid? Which data are supporting which model in the model suite? Which kind of errors of data is inherited by which part of which model?*

The final result of the data mining process is then a combination of the deep model and the normal model whereas the first one is a latent or hidden component in most cases. If we want, however, to reason on the results then the deep model must be understood as well. Otherwise, the results may become surprising and may not be convincing.

4.5 Controllers and Selectors

Algorithmics [6] treats algorithms as general solution pattern that have parameters for their instantiation, handling mechanisms for their specialization to a given environment, and enhancers for context injection. So, an algorithm can be derived based on explicit selectors and control rules [4] if we neglect context injection. We can use this approach for data mining design (DMD). For instance, an algorithm pattern such as regression uses a generic model of parameter dependence, is based on blind search, has parameters for similarity and model quality, and has selection support for specific treatment of the given data set. In this case, the controller is based on enablers that specify applicability of the approach, on error rules, on data evaluation rules that detect dependencies among control parameters and derive data quality measures, and on quality rules for confidence statements.

4.7 Data Mining and Design Science

Let us finally associate our approach with design science research [13]. Design science considers systematic modelling as an embodiment of three closely related cycles of activities. The *relevance cycle* initiates design science research with an application context that not only provides the requirements for the research as inputs but also defines acceptance criteria for the ultimate evaluation of the research results. The central *design cycle* iterates between the core activities of building and evaluating the design artifacts and processes of the research. The orthogonal *rigor cycle* provides past knowledge to the research project to ensure its innovation. It is contingent on the researchers' thoroughly research and references the knowledge base in order to guarantee that the designs produced are research contributions and not routine designs based upon the application of well-known processes.

The relevance cycle is concerned with the problem specification and setting and the matrix and agenda derivation. The design cycle is related to all other phases of our framework. The rigor cycle is enhanced by our framework and provides thus a systematic modelling approach.

5 Conclusion

The literature on data mining is fairly rich. Mining tools have already gained the maturity for supporting any kind of data analysis if the data mining problem is well understood, the intentions for models are properly understood, and if the problem is professionally set up. Data mining aims at development of model suites that allows to derive and to draw dependable and thus justifiable conclusions on the given data set. Data mining is a process that can be based on a framework for systematic modelling that is driven by a deep model and a matrix. Textbooks on data mining typically explore in detail algorithms as blind search. Data mining is a specific form of modeling. Therefore, we can combine modeling with data mining in a more sophisticated form. Models have however an inner structure with parts which are given by the application, by the context, by the commonsense and by a community of practice. These fixed parts are then enhanced by normal models. A typical normal model is the result of a data mining process.

The current state of the art in data mining is mainly technology and algorithm driven. The problem selection is made on intuition and experience. So, the matrix and the deep model are latent and hidden. The problem specification is not explicit. Therefore, this paper aims at the entire data mining process and highlights a way to leave the ad-hoc, blind and somehow chaotic data analysis. The approach we are developing integrates the theory of models, the theory of problem solving, design science, and knowledge and content management. We realized that data mining can be systematized. The framework for data mining design exemplarily presented is an example in Figure 4.

References

[1] G. Bell. The mechanism of evolution. Chapman and Hall, New York (1997)

[2] R. Berghammer and B. Thalheim., Methodenbasierte mathematische Modellierung mit Relationenalgebren. In: Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung, pp. 67–106. De Gryuter, Boston (2015)

[3] M.R. Berthold, C. Borgelt, F. Höppner, and F. Klawonn. Guide to intelligent data analysis. Springer, London (2010).

[4] A. Bienemann, K.-D. Schewe, and B.Thalheim. Towards a theory of genericity based on government and binding. In: Proc. ER'06, LNCS 4215, pp. 311–324. Springer (2006)

[5] L.B. Booker, D.E. Goldberg, and J.H. Holland. Classifier systems and genetic algorithms. Artificial Intelligence, 40 (1–3): pp. 235–282 (1989)

[6] G. Brassard and P. Bratley. Algorithmics - Theory and Practice. Prentice Hall, London (1988)

[7] A. Coleman. Scientific models as works. Cataloging & Classification Quarterly, Special Issue: Works as Entities for Information Retrieval, 33, pp. 3-4 (2006)

[8] A. Dahanayake and B. Thalheim. Co-evolution of (information) system models. In: EMMSAD 2010, LNBIB vol. 50, pp. 314–326. Springer (2010)

[9] D. Embley and B. Thalheim (eds). The Handbook of Conceptual Modeling: Its Usage and Its Challenges. Springer (2011)

[10] N.P. Gillett, F.W. Zwiers, A.J. Weaver, G.C. Hegerl, M.R. Allen, and P.A. Stott. Detecting anthropogenic influence with a multi-model ensemble. Geophys. Res. Lett., 29:31–34, 2002.

[11] E. Guerra, J. de Lara, D.S. Kolovos, and R.F. Paige. Inter- modelling: From theory to practice. In MoDELS 2010, LNCS 6394, pp. 376–391, Springer, Berlin (2010)

[12] H. Haken, A. Wunderlin, and S. Yigitbasi. An introduction to synergetics. Open Systems and Information Dynamics, 3(1): pp. 1–34 (1994)

[13] A. Hevner, S. March, J. Park, and S. Ram.

Design science in information systems research. MIS Quaterly, 28(1): pp. 75–105 (2004)

[14] P.J. Hunter, W. W. Li, A. D. McCulloch, and D. Noble. Multiscale modeling: Physiome project standards, tools, and databases. IEEE Computer, 39(11), pp. 48–54 (2006)

[15] ISO/IEC 25020 (Software and system engineering - software product quality requirements and evaluation (square) - measurement reference model and guide). ISO/IEC JTC1/SC7 N3280 (2005)

[16] H. Jaakkola, B. Thalheim, Y. Kidawara, K. Zettsu, Y. Chen, and A. Heimbürger. Information modelling and global risk management systems. In: Information Modeling and Knowledge Bases XX, pp. 429–446. IOS Press (2009)

[17] K. Jannaschk. Infrastruktur für ein Data Mining Design Framework. PhD thesis, Christian-Albrechts University, Kiel (2017)

[18] F. Kramer and B. Thalheim. A metadata system for quality management. In: Information Modelling and Knowledge Bases, pp. 224–242. IOS Press (2014)

[19] O. Nakoinz and D. Knitter. Modelling Human Behaviour in Landscapes. Springer (2016)

[20] J. Pardillo. A systematic review on the definition of UML profiles. In: MoDELS 2010, LNCS 6394, pp. 407–422, Springer, Berlin (2010)

[21] D. Petrelli, S. Levin, M. Beaulieu, and M. Sanderson. Which user interaction for cross-language information retrieval? Design issues and reflections. JASIST, 57(5): pp. 709–722 (2006)

[22] O.H. Pilkey and L. Pilkey-Jarvis. Useless Arithmetic: Why Environmental Scientists Cant't Predict the Future. Columbia University Press, New York (2006)

[23] A.S. Podkolsin. Computer-based modelling of solution processes for mathematical tasks (in Russian). ZPI at Mech-Mat MGU, Moscov (2001)

[24] M. Pottmann, H. Unbehauen, and D.E. Seborg. Application of a general multi-model approach for identification of highly nonlinear processes – a case study. Int. Journal of Control, 57(1): pp. 97–120 (1993)

[25] B. Rumpe. Modellierung mit UML. Springer, Heidelberg (2012)

[26] A. Samuel and J. Weir. Introduction to Engineering: Modelling, Synthesis and Problem Solving Strategies. Elsevier, Amsterdam (2000)

[27] G. Simsion and G.C. Witt. Data modeling essentials. Morgan Kaufmann, San Francisco (2005)

[28] M. Skusa. Semantische Kohärenz in der Softwareentwicklung. PhD thesis, CAU Kiel, (2011)

[29] B. Thalheim. Towards a theory of conceptual modelling. Journal of Universal Computer Science, 16(20): pp. 3102–3137, (2010)

[30] B. Thalheim. The conceptual model ≡ an adequate and dependable artifact enhanced by concepts. In: Information Modelling and Knowledge Bases XXV, pp. 241–254. IOS Press (2014)

[31] B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In: Encyclopedia of Database Systems, Springer (2017)

[32] B. Thalheim and M. Tropmann-Frick. Wherefore models are used and accepted? The model functions as a quality instrument in utilisation scenarios. In: I. Comyn-Wattiau, C. du Mouza, and N. Prat, editors, Ingenierie Management des Systemes D'Information (2016)

[33] B. Thalheim, M. Tropmann-Frick, and T. Ziebermayr. Application of generic workflows for disaster management. In: Information Modelling and Knowledge Bases, pp. 64–81. IOS Press (2014)

[34] B. Thalheim and Q. Wang. Towards a theory of refinement for data migration. In: ER'2011, LNCS 6998, pp. 318–331. Springer, (2011)

[35] T. Zeugmann. Inductive inference of optimal programs: A survey and open problems. In: Nonmonotonic and Inductive Logics, pp. 208–222. Springer, Berlin (1991)

Acknowledgement. We thank for the support of this paper by the CRC 1266. We are very thankful for the fruitful discussions with the members of the CRC.

The Rigor Cycle of Conceptual Modelling

Ajantha Dahanayake[1] and Bernhard Thalheim[2]

[1] School of Business and Management, Lappeenranta University of Technology, Finland
`Ajantha.Dahanayake@lut.fi`
[2] Department of Computer Science, Christian Albrechts University Kiel, D-24098 Kiel
`thalheim@is.informatik.uni-kiel.de`

Abstract. Modelling is still conducted as the work of an artisan and workmanship. While a general notion of the model and of the conceptual model has already been developed, the modelling process is not investigated so well. It is currently based on modelling methodologies. Modelling has to be based on principles and a general theory of modelling activities. The modelling activities need however a conceptualisation. We combine approaches developed in design science, ontology engineering, decision processes, and conceptual modelling for development of general stages, phases and steps of modelling. The main elements of our approach discussed in this paper are the way how a modelling decision is made and which phases and steps are commonly observed during modelling.

Keywords: Conceptual modelling, modelling actions, modelling decisions, phases and steps of modelling

1 Introduction

Design science research and conceptual modelling research have attracted a lot of research. Design science "is the scientific study and creation of artefacts as they are developed and used by people with the goal for solving practical problems of general interest. " [20] "Conceptual modeling is about describing" (syntax,) "semantics" (, and pragmatics) "of software applications at a high level of abstraction. Specifically, conceptual modelers (1) describe structure models in terms of entities, relationships, and constraints; (2) describe behavior or functional models in terms of states, transitions among states, and actions performed in states and transitions; and (3) describe interactions and user interfaces in terms of messages sent and received, information exchanged, and look-and-feel navigation and appearance." [9].

Comparing these general statements, we observe a good overlap whenever information systems are the target of development. The design of an IT artefact includes explication of the problem, definition of requirements, development of the artefact, demonstration of the artefact, and evaluation of the artefact. A similar flow of activities can be distinguished for modelling. Both approaches to development of artificial artefacts [41] may be based on methodological frameworks, on general paradigms and principles, and on general ethical, economical, ecological etc. principles.

Therefore, it is beneficial to integrate the two approaches. In general, design and modelling are partially different and - at the same time - largely similar activities[1]. It seems that the two development approaches share many issues and can benefit from each other.

The controversy [1, 21, 25, 32] discuss the differences between design science and information systems research. It seems that the two research directions are completely different and do not have too much in common. Design science has its background in industrial and interior design and in psychology. Conceptual modelling started with database modelling and is more directly influenced by computer engineering.

Conceptual modelling is a specific form of modelling. Models become conceptualised due to incorporation of concepts - or more generally, conceptions - into the model. These concepts are commonly shared within a community of practice that is involved in the modelling process. Models are a universal vehicle or better instrument in almost all sciences and engineering. They can be understood as the ‘third’ dimension of science [5, 52].

Therefore, we can compare design science research for information systems development with conceptual modelling of information systems. Design science distinguishes the relevance cycle, the modelling cycle, and finally the rigor cycle. In this paper we look more specifically into the rigor cycle and use it for development principles that can be incorporated into for conceptual modelling. Since neither the rigor cycle nor the principles of conceptual modelling have led to an accepted theory, we start our research with one specific aspect of systematic modelling: support for design decisions.

1.1 Models in Design Science

Models, modelling languages, modelling frameworks and their background have dominated conceptual modelling research and information systems engineering for the last four decades. Design science research considers artefacts. It is understood as an object or thing made by humans with the intention that it will be used to address a practical problem. Artefacts are, for instance, physical objects, drawings or blueprints. Models are also artefacts whenever they are not virtual. Artefacts are used in development scenarios. Their functions are what they can do for members in their community of practice, what role they can play for them, and how they can support them in their activities.

Conceptual models are mediators between the application world and the implementation or system world [6]. Design science distinguishes the relevance

---

[1] *To design* means, for instance, (1) to create, fashion, execute, or construct according to a plan, (2) to conceive and plan out in the mind driven by a purpose and devised for a specific function or end, and (3) to develop an artefact.
To model means, for instance, (1) to plan or form after a pattern, (2) to shape or fashion, and (3) to construct a model guided by an origin. [38].

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 [14]. Research in design science and on conceptual modelling has resulted in a large body of knowledge, practices, and techniques. Modelling is based on modelling activities. 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 [49] used for system development in computer engineering. The separation into the application world, the modelling and model world and the knowledge or design science world [13, 48] supported an assessment of the results of modelling and an evaluation of the results of research on modelling.

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 syntactic, semantics and pragmatics of model, and on evaluation and quality guaranteeing methods [16, 31, 34, 48]. The application world is used as a starting point for the development of systems that solve some problems of the application domain under consideration. By analyzing these two directions we come to a conclusion similar to [60]. In reality design science research and research on conceptual modelling are two research issues that may benefit from each other. The two communities are already engaged in a discussion of the added value of each side [3, 4, 24, 26, 33, 35, 36, 42, 58, 57, 59].

1.2 The Three Perspectives of Conceptual Modelling

Based on the notions in the Encyclopedia Britannica [38], we distinguish between the conception of a *model*, the conception of a *model activity*, and the conception of systematic, reflected and well-organised *modelling*.

The model as an artifact: A model is a well-formed, adequate, and dependable instrument that represents origins. [2, 8, 50, 51]

To model as an activity: 'To model' is a scientific or engineering activity beside theoretical or experimental investigation. The activity is an additive process. Corrections are possible during this activity. Modelled work may be used for construction of systems, for exploration of a system, for definition and negotiation, for communication, for understanding and for problem solving.

Modelling as a systematically performed, reflected, technological process: Modelling is a technique for systematically using knowledge from computer science and engineering to introduce technological innovations into the planning and development stages of a system.

1.3 Modelling as an Activity

Modelling includes two different kinds of activities:

Model deployment is based on activities such as

- adaption, concept enrichment, optimisation, specialisation, instantiation, refinement, grinding,
- applicability studies (evaluation, assurance, composition for application),
- integration, selection, renovation, modernisation, (r)evolution, migration,
- problem solution, classification, practice, understanding, theory or paradigm (r)evolution, and
- explanation.

Model development is typically based on another set of activities such as

- abstraction of origin, scoping, validation, verification, testing, optimisation,
- construction, composition, definition, integration, classification, invention,
- enrichment, adaption, mutation, recombination, refinement, reuse, preparation for deployment, and
- understanding, theory or paradigm injection.

1.4 Objectives and the Storyline of the Paper

In this paper, we discuss modelling foundational principles and theoretical underpinnings for purpose-oriented models and modelling. Our approach is based on the three cycles of design science research activities of research artifact creation. We thus combine conceptual modelling, design science approaches, decision processes, ontology engineering, and the theory of information system models. We do not intent to review all relevant literature in the rich body of knowledge developed in design science research or conceptual modelling research. There are conference series such as DESRIST, ER, and Models etc. and journals such as DKE, EJIS, and MISQ etc. Instead, we follow the approach [58] and use design science research for conceptual modelling of information systems.

Section 2 provides an account of design science, its position on modelling, and the stages of design. Section 3 describes the modelling decisions and its parallels to systematic decision support and the modelling act leading to models and solution imperfection. Section 4 gives an account of systematic conceptual modelling, exploration, and model amalgamation leading to formal model foundation. Section 5 summarises the conclusions of this research.

2 Design Science and Modelling

Design science originated in the area of IT development. It concentrates on novel artifacts in the form of models, methods, and systems that support people while developing, using, maintaining, reconsidering, and migrating IT solutions. It considers four perspectives [20]: (1) people, practices and problems; (2) artifacts as solutions to problems in IT practices; (3) the context and anatomy of artifacts; and (4) the study of artifacts.

2.1 The Relevance Cycle in Design Science

Design science research requires the creation of an innovative, purposeful artifact for a special problem domain. The artifact must be evaluated in order to ensure its utility for the specified problem. The *relevance cycle* initiates design science research with an application context that not only provides the requirements for the research as inputs but also defines acceptance criteria for the ultimate evaluation of the research results. The rigor cycle provides past knowledge to the research project to ensure its innovation. It is contingent on the researchers to thoroughly research and reference the knowledge base in order to guarantee that the designs produced are research contributions and not routine designs based upon the application of well-known processes. The central design cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research.

2.2 The Modelling or Design Cycle

Modelling is a crucial activity in the creation of the design, the artifact. The models and modelling itself implies an ethical change from describing and explaining of the existing world to shaping it. One can question the values of this type of models and modelling oriented design research, i.e. whose values and what values dominate it, emphasizing that research may openly or latently serve the interests of particular dominant groups. The interests served may be those of the host organization as perceived by its top management, those of IS users, those of IS professionals or potentially those of other stakeholder groups in society. Therefore, in order to define the acceptance criteria for ultimate evaluation of the research, modelling and models need to be mapped to a theoretical foundation.

2.3 The Rigor Cycle

The *rigor cycle* is considered as the conceptualisation and generalisation or knowledge development cycle [56]. 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 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 which is somehow orthogonal 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 patterns or reference models and for extraction of model and modelling knowledge beyond the actual modelling activity. In this paper, we concentrate on the rigor cycle of conceptualization and the knowledge development for modelling foundational principles and theoretical underpinning to validate the purposeful values of models and modelling within the design science research activities.

2.4 Stages Of "To Model"

Based on foundations of conceptual modelling [46], ontology engineering [45], and design science for information systems development (e.g. [29]) and summarising, we distinguish three stages of modelling activities:

Stage I: ***Model development*** is based on four phases: description, formulation, ramification, and validation. In the description phase, individual perception and situation models involved into the modelling situation, are isolated and the corresponding primary properties are identified and represented. We realise in the next sections that this phase includes exploration and model amalgamation. In the formulation phase, properties are interrelated, integrated and combined into a preliminary, initial model. This model is analysed in a ramification phase in order to check whether the model is a proper solution and to interpret and to consider its implications. Finally, the model and its capability and capacity are assessed in a validation phase.

Stage II: ***Model deployment*** considers the developed model within the given application situation, assesses this model in other application contexts in order to evaluate its stability and plasticity, and derives its added value.

Stage III: The rigor cycle also investigates the experience we have gained during developing the given model. Conceptual modelling uses this experience as a hidden intuitive basis for further development. We may however use this experience within a ***paradigmatic synthesis*** for recapitulation and consolidation of conceptualisation concept gathering, ontologisation, grounding and tagging, i.e. for knowledge acquisition.

3 Modelling Decisions

The main question is now how, when, why, on what, in which way and why design decisions are made beside the organisation of the design process itself, its flow of activities, and the involvement of actors into the design process.

3.1 Systematic Decision Support

According to [22], modelling and modelling decisions enhancement (DE) activities are encouraged within the studio concept. A DE studio has five main components:

- *Studio style*: Learning, Enquiry and Participative.

- *Decision process coordinators*: these include facilitators, domain experts, and suite.

- *Scripting*: the balance between improvisation and formalized methods.

- *Suites and development and support expertise.*

- *Location and rooms*: the options here range from fixed point to distributed Web conferencing and from simple technology infrastructure to multimedia heaven.

[22] consider the mix of skills required to fulfill the demands of a studio for modelling decisions listed as landscaping, facilitation, recipes, suites and process as a means of a complete package of developing an architecture - a solution. Let us combine this approach with the technology proposal for change management in [19].

- *Landscaping* is the domain of expertise of the business strategist and domain expert. In terms of both understanding the decision issues and decision-makers, information resources, processes and the basics of what to model, why and how. In addition, the landscaper has to have some credibility, whether as an insider or outside adviser, with senior managers and stakeholders. Otherwise, the studio is just an exercise or a “pilot”, “prototype” or “lab”, all of which are euphemisms for “dont take this too seriously.”
- *Facilitation*: Behavioral knowledge and process skills are a key for the process of arriving at a solution.
- *Recipes* apply wherever possible proven recipes that include effective scripts. Recipes are proven, repeatable and transferable, specify ingredients and sequencing, permit variations and innovations, and result in something people eat and are likely to come back for another meal. Building recipes requires research and writing and the willingness to place “secrets” and “methodology” in the public domain. It demands teaching as well: developing a body of knowledge and building a critical mass of skilled practitioners. Since technology moves so fast, each new generation of software draws on a new generation of developer and there is little passing on of experience and knowledge.
- *Suites* ensure that tools are designed and implemented within an overall distributed architecture. The goal of suite development is to make the “system” as transparent, easy to access, reliable as the electrical system, where any breakdown is a news item and crisis.
- *Processes* make commitment to a decision of the explicit target and agenda.

The blockage here is organizational culture, management style, stakeholder relationships and legacy of existing decision processes.

3.2 The Modelling Action and Design Decisions

The modelling action is similar to the speech act and consists of

1. a selection and construction of an appropriate model depending on the task and purpose and depending on the properties we are targeting and the con-

text of the intended system and thus of the language appropriate for the system,
2. a workmanship on the model for detection of additional information about the original and of improved model,
3. an analogy conclusion or other derivations on the model and its relationship to the real world, and
4. a preparation of the model for its use in systems, to future evolution and to change.

Therefore, the DE studio approach provides a specific tactics to modelling.

3.3 Modelling Knowledge and Decisions Imperfection

In the case of conceptual modelling, the rigor cycle can be based on knowledge obtained within the five consecutive phases [11]:

1. *exploration*,
2. *model amalgamation and adduction*,
3. *model formulation*,
4. *model deployment*, and
5. *paradigmatic synthesis*.

Modelling decisions have to be based on transparent and realistic objectives [11, 52, 54, 55]. The correspondences of elements of the conceptual model to particular pattern in the real world or the perception models must be based on conformity criteria. Modelling can be considered as progressive cognition within the context, for the purpose of development and within the concept space. Models cannot be developed in its full scientific rigor and are thus objects of evolution. Modelling actions balance between exploratory decisions (description, explanation, prediction) and inventive aspects (reification, refinement). Modelling kits are supporting the quality of modelling decisions. Modelling actions suffer from the breadth-depth paradox. We want to have as much detail as necessary and want to be as broad as sufficient. Modelling decisions must be continuously evaluated within a modelling process by either mode or all three modes of assessment (coherence, correspondence, commensurability). They are conditionally anchored to the experience, knowledge and intuition gained so far. Modelling also includes negotiation within the community of practice and with the stakeholders of the information system.

Adequateness of models is based on analogy, focusing, and purposefulness of the model. Focusing provides a means for explicit modelling of the divergence from the real world with incompleteness, open issues and potential errors [15]. Therefore, a model is imperfect [17] due to exceptional states that are not considered, incompleteness due to limitations of the modelling language and the scope of modelling, and due to errors, which are either based on real errors or exceptional states or on biases by the community of practice.

4 Systematic Development of Conceptual Models

Let us now consider the modelling phases and steps and highlight the decisions that must be made during modelling. We concentrate in this paper on the first two model development phases: description and formulation. Ramification and validation extend the approach in [55]. The two next stages (model deployment and paradigmatic synthesis) are deferred to a forthcoming paper due to space limitations. The section is based on our entire experience on conceptual modelling and on the experiences of several decades of database realisation[2]. The body of knowledge developed so far and used in real practice is very large. It needs however a systematisation, categorisation and generalisation. There are very few publications (e.g. [10, 12, 30, 40, 39, 43]) that provide such systematisation of the experience gained so far. The generalisation and the categorisation is however an open research field so far.

Modelling of structures is a systematically performed technological process. It is a technique for applying knowledge from other branches of engineering and disciplines of science in effective combination to solve a multifaceted engineering problem. In addition to structure development, it is important to define databases systems themselves. The systems are first of all man-made. migration-

---

[2] Due to involvement of the second author into the development and the service for the CASE workbenchs $(DB)^2$ and ID^2 we have collected a large number of real life applications. Some of them have been really large or very large, i.e., consisting of more than 1.000 attribute, entity and relationship types. The largest schema in our database schema library contains of more than 19.000 entity and relationship types and more than 60.000 attribute types that need to be considered as different. Another large database schema is the SAP R/3 schema. It has been analysed in 1999 by a SAP group headed by the second author during his sabbatical at SAP. At that time, the R/3 database used more than 16.500 relation types, more than 35.000 views and more than 150.000 functions. The number of attributes has been estimated by 40.000. Meanwhile, more than 21.000 relation types are used. The schema has a large number of redundant types which redundancy is only partially maintained. The SAP R/3 is a very typical example of a partially documented system. Many of the design decisions are now forgotten. The high type redundancy is mainly caused by the incomplete knowledge on the schema that has been developed in different departments of SAP over several decades.

resistant. Modelling and especially information system modelling[3,4] is a creation and production process, an explanation and exploration process, an optimisation and variation process, and a verification process. This distinction allows to relate the specific purpose with macro-steps of modelling and with criteria for approval or refusal of modelling results. Modelling is thus at the same time problem solving and engineering.

4.1 The Model Description Phase

In this paper we concentrate our investigation of the model description phase on two (macro-)steps: model exploration and model amalgamation.

Main Phases of Model Description for Starting from Scratch. The exploration step is based on state-of-affairs and the functions a model should play during information system development. The state-of-affairs is typically represented by a reality model that already abstracts from the state-of-affairs and perception models that are used in the community of modellers and business users. We may distinguish in this step a number of activities: the situation and the perception models are disassembled. Later, monstration may be applied to some situation model. This situation model is negotiated within the community of practice and users and represented by a nominal model.

---

[3] We develop our approach here on the approach that is established and widely practised and taught in almost all textbooks. We leave out the more sophisticated approach in [53]. At the business user level, user viewpoints can be represented by user viewpoint schemata or more generally views. At the conceptual level, these viewpoints are going to be harmonised and mapped to a conceptual schema. It is assumed that the user viewpoints are then sub-schemata of the conceptual schema. This conceptual schema is mapped to a logical and later to a physical schema. The viewpoints are cut down to logical views which typically consist of single-table definitions on the basis of a query to the logical schema. A user viewpoint is then called external view. The query might be more complex and thus not be based on a sub-schema of the conceptual schema. The database structure architecture consists of the logical schema, external views defined on top of the logical schema and an implementation or physical schema. With the introduction of the conceptual model, the architecture description has been changed by considering the logical and the physical as an implementation schema and using the conceptual schema as the mediator between views and the implementation schemata. It creates a mismatch since the views are defined on top of the implementation schema. [18] breaks with this three-layer architecture by proposing the conceptual view tower mechanism where business user viewpoints are represented by conceptual views. [53] rounds off this approach by considering the conceptual model to consist of a conceptual schema and a collection of conceptual views.

[4] We concentrate the investigation of the modelling process to ‘greenfield’ modelling called modelling from scratch and to model gardening called evolutionary modelling. We do not investigate ‘brownfield’ modelling and modernisation for already operating database systems based on modelling and redevelopment for legacy systems based on macro-modelling methods and especially migration strategies [23]

The next step is model amalgamation. The result is a real model. Amalgamation is oriented on the justification of the model and on the quality criteria for the model. It integrates also criteria for well-formedness of models.

Modelling typically also results in modelling experience that can be elicited during or after model development. This modelling experience elicitation and acquisition is part of paradigmatic synthesis and therefore of the rigor cycle.

Evolutionary Model-Based and Background-Aware Modelling. Modelling is often not performed from scratch. Rather we start with an explication of experience that uses stereotypical or generic models. We may also start with existing models for an already existing information system. We thus elaborate artifacts of interest, e.g. reference or existing models. We explicitly extract the background of these models. Next we explicate their purpose, their background, their context and compare the result with the reality models and the objectives of development. The result is again a situation model.
This situation model is now assessed and evaluated. It is typically reformulated by specialisation and refinement. We thus explicitly describe why the model is adequate and dependable. This model can also be enhanced by formal methods.

In a similar form, the experience gained is incorporated into the body of modelling knowledge.

4.2 The Exploration (Macro-)Step

Exploration start with a well-defined modelling task, a well-defined scope, and consider choices. It is often assumed to be based on deductive approaches. It seems to better however to consider inductive and abductive approaches first.

It is based on the following three steps:

Disassemble: The perception and situation models are converted into constituent parts in dependence on the specific assumptions, specific reality and state-of-the-art properties, and specific foci and scopes. Methods are dissolution, segmentation, analysis of coherent units, refinement and categorisation, and examination.

Monstration/manifestation is the act of demonstrating, exhibiting, and demonstration. It often considers familiar situations and examples. We consider typical application situations, typical phenomena, typical system states, concepts and conceptions. During monstration we are more interested in specific kinds of models, Galilean models oriented on improvement of the state-of-the-art.

Monstration may follow the W*H specification framework [7]. Typical questions answered are: What is the demonstrated situation about? What systems and phenomena are involved into the situation? What is the state of every system? What concepts are necessary to describe and/or explain? What is the reference system? How can these concepts be represented?

Manifestation and reflection consider model properties, model variants and model capabilities. Based on these considerations we may derive obligations for model revision.

Nominal models in the exploration step use parameters and variables that can be instantiated during further model development and progressive model refinement. The glossary, namespace, agreements on conceptions, assumptions on the model background, decisions on model structures, and composition pattern are often imported from the application domain.

The result of this step is now a nominal (or perception) model that is a generalisation of a subsidiary models. This model explicitly represents the background via its grounding (paradigms, culture, background, foundations, theories, postulates, restrictions, authorities, conventions, commonsense) and its basis (concepts, language, routine, training, infrastructure, assumptions, though community, thought style, pattern, methodology, guidelines, practices).

4.3 The Model Amalgamation (Macro-)Step

Amalgamation aims at combination or unification of model elements into one form [47]. It includes merger, consolidation, and mixing or blending of different elements. We typically focus on one real model during conceptual data modelling of information systems. We might also use several models but leave it out of scope within this paper.

Amalgamation is mainly based on inductive reasoning. It might incorporate abductive and adduction reasoning based on the association to the situation and perception models. It can be enhanced by methods of plausible reasoning. Classical development strategies (top-down, bottom-up, modular, inside-out, mixed) provide means for combination and unification of elements. In this case we can use local top-down and bottom-up development operations for ER models [46].

Model composition can follow a number of strategies and tactics, especially for unification. Since we are interested in adequacy and dependability of models we explicitly propagate these properties during amalgamation. Typical ER-based composition principles are: global-as-design, unification of viewpoints, explicit consideration of realisability, empiric evaluation by sample data, homomorphic mappings from the situation and perception models, and consideration of specific elements of the ER modelling language. This step is often governed by practical guidelines and rational constraints (general guiding principles, acceptable tolerance (approximation limits, precision intervals, data preciseness), convenient modes for logging and handling of data, appropriate mathematical or formal representations and operations, norms and corroboration for the real model and criteria for evolution (refinement, modernisation, modification, replacement); level of detail, type system and mapping style to type system, handling of exceptions and deviations (NULL, default), treatment of hierarchies, controlled redundancy, ground type system, quantity matrix (Mengengerüst), constraint enforcement, treatment of cardinality, inherent constraints, naming conventions, abbreviation rules, kind of semantics (set or pointer), weak types, translation and tolerance for complex attributes, handling of identification). We may also require that structures used correspond to natural situations (good design is functional, useful, aesthetic, innovative, good business, honest, long-lasting, minimalistic, understandable, user-oriented, unobtrusive, simple as possible, thorough down to the last detail, and focused).

Finally, maturity of models based on SPICE or CMM has to be provided [55]. The real model should be fully defined, must be well-understood, provide all semantics in an explicit form, and use explicit concepts. We might use different definitional frames but in a coherent form. General modelling principles are modularisation, abstraction, and explicit coupling. It is a good approach to use best practices within the modelling framework. They should allow to preserve also design principles that are given for the realisation environment.

4.4 The Model Formulation Phase

The model formulation phase aims at formulation of the well-formed, adequate and dependable model. This model may use several representation forms, e.g. conceptual data models combine diagrammatic and formal representations. It may also contain several sub-models that represent viewpoints of business users [53]. Finally, the assessment of the model is explicitly given.

The formulation is based on decisions such as depiction of the elements of previous models with an explicit consideration of the model function and purpose. We develop criteria for adequacy and dependability of the model and start to explicitly represent the model grounding and background. Since the model should also represent various viewpoints of business users we have to enhance it by explicit view schemata and aggregations as well as abstractions. Reasoning on justification might be based on an argument calculus [52] or argument logics [27].

The model must be completed. Typical drivers for completion are application domain requirements, the background behind the situation and perception models, the specifics of the modelling language, and the generic model behind the model. We need to adjust the scope of modelling elements. The goal is also to develop a well-formed model that fits well to the situation and perception models.

The model is also assessed by an elementary deployment and tested against the real world. This assessment is often backed by some test data based on an experimentation strategy. It might also be tested by elementary utilisation. The first main result of assessment is a justification of the model by an explanatory statement, by confirmation of rational coherence, by a validation of the model against the state-of-affairs, and by explicit consideration of stability of the model against non-essential deviations of the state-of-affairs. The second main result of assessment is an explicit statement on model quality based on quality characteristics for quality in use, external quality and internal quality. The assessment allows to reason on the model capacity and potential.

5 Conclusion

5.1 Combining Conceptual Modelling and Design Science

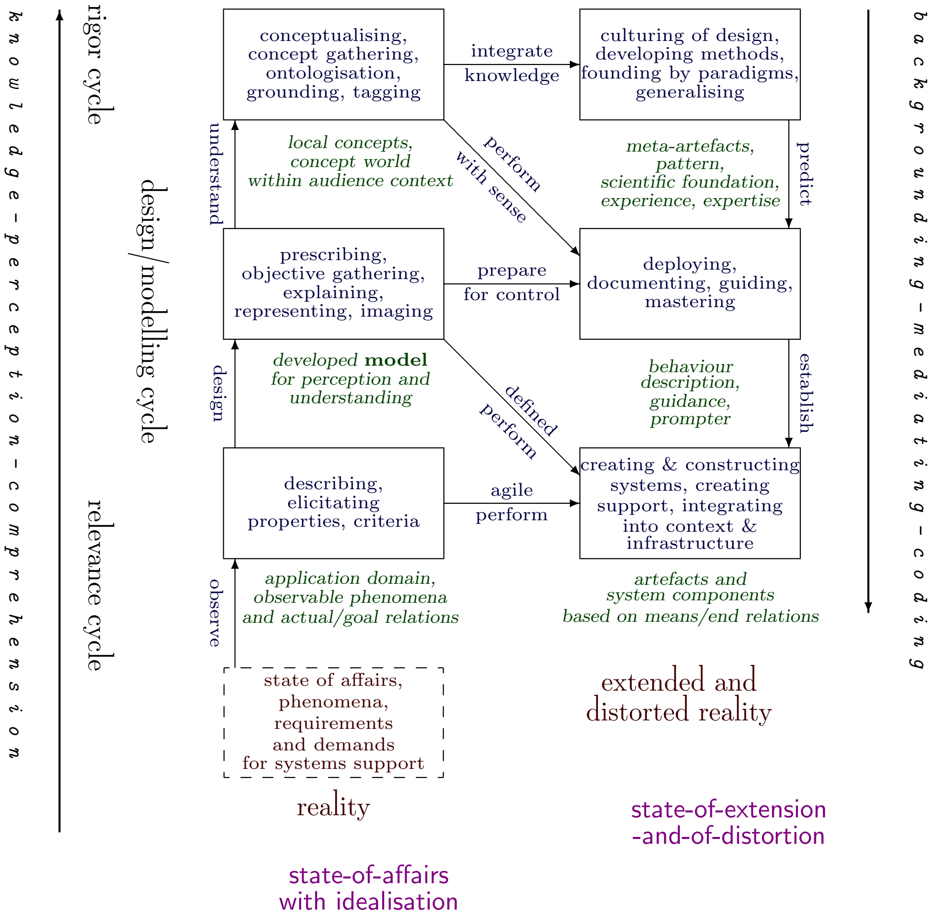


Fig. 1. The relevance/design/rigor and the state-of-affairs/augmentation dimensions [49]

This paper shows how design science and conceptual modelling can benefit from each other. More specifically we discuss how conceptual modelling can benefit from design science research. Let us use and enhance Figure 1 from [49]. Modelling of IT artifacts typically starts with an understanding of the state of affairs, with objectives and consideration of requirements. This perception may be described and directly used for the development of new IT artifacts without any model alike agile approaches. The modelling cycle results however in a model that can be used for IT development either directly or in a more reflected way. The rigor cycle in design science and modelling as a systematically performed, reflected and well organised process is based on an understanding of all actions undertaken. This process can also be used for development of new knowledge and its integration into the existing body of knowledge.

Therefore, we observe that the rigor cycle and systematics of modelling may each other enhance and complement.

5.2 Contributions of Design Science to Conceptual Modelling

Since design science research and conceptual modelling are tackling the same problem - proper development of (information) systems - we discussed in this paper how design science research can be used for an underpinning of modelling activities. The decision steps we presented are the basis for a general stepwise procedure of systematic design.

The formalisation of this approach is delayed to a forthcoming paper. Formalisation also includes approaches to a general theory of modelling such as [11, 28, 37, 44]. The main issue was so far the development of the combined approach.

References

1. R. Baskerville, K. Lyytinen, V. Sambamurthy, and D. Straub. A response to the design-oriented information systems research memorandum. *European Journal of Information Systems*, pages 1–5, 2010.
2. M. Bichler, U. Frank, D. Avison, J. Malaurent, P. Fettke, D. Hovorka, J. Krämer, D. Schnurr, B. Müller, L. Suhl, and B Thalheim. Theories in business and information systems engineering. *Business & Information Systems Engineering*, pages 1–29, 2016.
3. D. Bjørner. *Domain engineering*, volume 4 of *COE Research Monographs*. Japan Advanced Institute of Science and Technolgy Press, Ishikawa, 2009.
4. J. Vom Brocke and C. Buddendick. Reusable conceptual models - requirements based on the design science research paradigm. In *DESRIST*, pages 576–604, 2006.
5. S. Chadarevian and N. Hopwood, editors. *Models - The third dimension of science*. Stanford University Press, Stanford, California, 2004.
6. A. Dahanayake and B. Thalheim. Co-evolution of (information) system models. In *EMMSAD 2010*, volume 50 of *LNBIP*, pages 314–326. Springer, 2010.
7. A. Dahanayake and B. Thalheim. *Correct Software in Web Applications and Web Services*, chapter W*H: The conceptual Model for services, pages 145–176. Texts & Monographs in Symbolic Computation. Springer, Wien, 2015.
8. D. Embley and B. Thalheim, editors. *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*. Springer, 2011.
9. D. Embley and B. Thalheim. Preface. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, pages v–ix. Springer, Berlin, 2011.
10. M. Fowler. *Refactoring*. Addison-Wesley, Boston, 2005.
11. I.A. Halloun. *Modeling Theory in Science Education*. Springer, Berlin, 2006.
12. D. C. Hay. *Data model pattern: Conventions of thought*. Dorset House, New York, 1995.

13. A. Hevner, S. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quaterly*, 28(1):75–105, 2004.
14. A. R. Hevner. The three cycle view of design science. *Scandinavian J. Inf. Systems*, 19(2), 2007.
15. H. Jaakkola, J. Henno, T. Welzer-Družovec, J. Mäkelä, and B. Thalheim. Why information systems modelling is so difficult. In *SQAMIA'2016*, pages 29–39, Budapest, 2016. CEUR Workshop Proceedings.
16. H. Jaakkola and B. Thalheim. A framework for high quality software design and development: A systematic approach. *IET Software*, pages 105–118, April 2010.
17. H. Jaakkola and B. Thalheim. Exception-aware (information) systems. In *Information Modelling and Knowledge Bases*, volume XXIV, pages 300–313. IOS Press, 2013.
18. H. Jaakkola and B. Thalheim. Multicultural adaptive systems. In *Information Modelling and Knowledge Bases*, volume XXVI of *Frontiers in Artificial Intelligence and Applications, 272*, pages 172–191. IOS Press, 2014.
19. K. Jannaschk, H. Jaakkola, and B. Thalheim. Technologies for database change management. In *ADBIS Vol. II*, volume 312 of *Advances in Intelligent Systems and Computing*, pages 271–286. Springer, 2014.
20. P. Johannesson and E. Perjons. *An introduction to design science.* Springer, Cham, 2014.
21. I. Junglas, B. Niehaves, S. Spiekermann, B.C. Stahl, T. Weitzel, R. Winter, and R. Baskerville. The inflation of academic intellectual capital: the case for design science research in europe. *European Journal of Information Systems*, pages 1–6, 2010.
22. P.G.W. Keen and H.G. Sol. *Decision enhancement services : rehearsing the future for decisions that matter.* IOS Press, 2008.
23. M. Klettke and B. Thalheim. Evolution and migration of information systems. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, chapter 12, pages 381–420. Springer, Berlin, 2011.
24. B. Kuechler and V. Vaishnavi. On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, 17(5):489–504, 2008.
25. K. Lyytinen, R. Baskerville, J. Iivari, and D. Téeni. Why the old world cannot publish? overcoming challenges in publishing high-impact is research. *EJIS*, 16(4):317–326, 2007.
26. L. Maciaszek. *Requirements analysis and design.* Addison-Wesley, Harlow, Essex, 2001.
27. L. Magnani, W. Carnielli, and C. Pizzi, editors. *Model-Based Reasoning in Science and Technology: Abduction, Logic, and Computational Discovery.* Springer, 2010.
28. B. Mahr. On judgements and propositions. *ECEASST*, 26, 2010.
29. S.T. March and V.C. Storey. Design science in the information systems discipline: An introduction to the special issue on design science research. *MIS Quarterly*, 4:725–730, 2008.
30. D. Marco and M. Jennings. *Universal meta data models.* Wiley Publ. Inc., 2004.
31. A. Olivé. *Conceptual modeling of information systems.* Springer, Berlin, 2007.
32. H. Österle, J. Becker, U. Frank, T. Hess, D. Karagiannis, H. Krcmar, P. Loos, P. Mertens, A. Oberweis, and E.J. Sinz. Memorandum on design-oriented information systems research. *European Journal of Information Systems*, pages 1–4, 2010.
33. K. Pohl. *Process centred requirements engineering.* J. Wiley and Sons Ltd., 1996.

34. N. Prat, I. Comyn-Wattiau, and J. Akoka. Artifact evaluation in information systems design-science research - a holistic view. In *18th Pacific Asia Conf. on Inf. Syst., PACIS 2014, Chengdu, China, June 24-28, 2014*, page 23, 2014.
35. S. Robertson and J. Robertson. *Requirements-led project management.* Pearson, Boston, 2005.
36. C. Rolland. From conceptual modeling to requirements engineering. In *Proc. ER'06*, LNCS 4215, pages 5–11, Berlin, 2006. Springer.
37. A. Rutherford. *Mathematical Modelling Techniques.* Dover publications, 1995.
38. J.E. Safra, I. Yeshua, and et. al. *Encyclopædia Britannica.* Merriam-Webster, 2003.
39. L. Silverston. *The data model resource book. Revised edition*, volume 2. Wiley, 2001.
40. L. Silverston, W. H. Inmon, and K. Graziano. *The data model resource book.* John Wiley & Sons, New York, 1997.
41. H.A. Simon. *The science of the artificial.* MIT Press, Cambridge, 1996.
42. G. Simsion. *Data modeling essentials - Analysis, design and innovation.* Van Nonstrand Reinhold, New York, 1994.
43. G. Simsion and G.C. Witt. *Data modeling essentials.* Morgan Kaufmann, San Francisco, 2005.
44. B.Ja. Sovetov and S.A. Jakovlev. *Systems Modelling.* Vysschaja Schkola, 2005. In Russian.
45. V. Sugamaran and V. Storey. The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM TODS*, 31(3):1064–1094, 2005.
46. B. Thalheim. *Entity-relationship modeling – Foundations of database technology.* Springer, Berlin, 2000.
47. B. Thalheim. 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`.
48. B. Thalheim. 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*, chapter 17, pages 547–580. Springer, Berlin, 2011.
49. B. Thalheim. The science and art of conceptual modelling. In A. Hameurlain et al., editor, *TLDKS VI*, LNCS 7600, pages 76–105. Springer, Heidelberg, 2012.
50. B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.
51. B. Thalheim. Conceptual modeling foundations: The notion of a model in conceptual modeling. In *Encyclopedia of Database Systems.* 2017.
52. B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung.* De Gruyter, Boston, 2015.
53. B. Thalheim and M. Tropmann-Frick. The conception of the conceptual database model. In *ER 2015*, LNCS 9381, pages 603–611, Berlin, 2015. Springer.
54. B. Thalheim and M. Tropmann-Frick. Models and their capability. In C. Beierle, G. Brewka, and M. Thimm, editors, *Computational Models of Rationality*, volume 29 of *College Publications Series*, pages 34–56. College Publications, 2016.
55. B. Thalheim and M. Tropmann-Frick. Wherefore models are used and accepted? The model functions as a quality instrument in utilisation scenarios. In I. Comyn-Wattiau, C. du Mouza, and N. Prat, editors, *Ingenierie Management des Systemes D'Information*, 2016.

56. J. R. Venable. Design science research post Hevner et al.: Criteria, standards, guidelines, and expectations. In *DESRIST*, volume 6105 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2010.
57. Y. Wand, D. E. Monarchi, J. Parsons, and C. C. Woo. Theoretical foundations for conceptual modeling in information systems development. *Decision Support Systems*, 15:285 – 304, 1995.
58. R. Wieringa. *Design science methodology for information systems and software engineering*. Springer, Heidelberg, 2014.
59. R. Winter. Design science research in Europe. *European Journal of Information Systems*, 17(5):470–475, 2008.
60. S. Zelewski. Kann Wissenschaftstheorie behilflich für die Publikationspraxis sein? In F. Lehner and S. Zelewski, editors, *Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik*, pages 71–120. GTO, 2007.

Conceptual Modeling: Enhancement through Semiotics

Veda C. Storey[1] and Bernhard Thalheim[2]

[1]Computer Information Systems, J. Mack Robinson College of Business
Georgia State University, Atlanta, GA, USA
E-mail: vstorey@gsu.edu
[2]Department of Computer Information Systems, Kiel University, Germany
E-mail: thalheim@is.informatik.uni-kiel.de

Abstract. Conceptual modeling uses languages to represent the real world. Semiotics, as a general theory of signs and symbols, deals with the study of languages and is comprised of syntax, semantics, and pragmatics. Pragmatics includes the explicit representation of the intentions of users. A common assumption is that all levels of database design (user, conceptual, logical, and physical) can be modeled using the same language. However, languages at the conceptual level are often enhanced by concepts that attempt to capture inherent pragmatics. This research proposes that concepts from semiotics can provide the background needed to understand an application. Specifically, pragmatics and semantics are considered at both the user and conceptual level, based on proposed constraints.

Keywords: Conceptual modeling, languages, semiotics, semantics, constraints

1 Introduction

Conceptual models act as mediators between the application and an implementation [11]. Conceptual modelers often attempt to model situations that occur in the real world using one language as a construction mechanism, and a model for a schema. Representing how the world operates must be described at the right level of specification. This tends to be done, for example, using an entity-relationship diagram as a modeling tool. However, it is difficult to expect one language to be able to handle all phases of modeling. Semantic issues need to be captured and modeled during both the design phases. The objective of this research, therefore, is to understand how to create better conceptual models by considering these different levels of abstraction and how they might be addressed. Although language is usually the main vehicle for modeling, additional understanding is needed for collaboration among stakeholders. Semiotics, as a general theory of signs and symbols, deals with the study of languages, and could serve as the needed background. The contributions are to: propose that models should be defined from the perspective of semiotics, and propose an additional set of constraints.

ER'17, LNCS 10650,, Cham, 2017, Springer

2 Modeling Challenges in Conceptual Modeling

Levels of abstraction. Many modeling languages are applied at different levels of abstraction. Business issues might be applied at the application level. Prescription issues for implementation are at a detailed level of specification. Although different, they are often all represented by an entity-relationship diagram.

Semantics. Semantics (meaning of terms) is challenging [5]. Constraints are often used as a surrogate for business rules [6]. Attempting to capture and represent semantics in terms of first-order predicate logic seems restrictive. Implicit or lexical semantics contribute to complete semantics.

Inclusion constraints. These could be class-based; for example, a student is a person. The person identification is reused for student as a co-existence constraint, expressible via identification (becoming a foreign key constraint in the relational model). Then an enforcement mechanism can be: 1) canonically declared based on reference existence and reference enforcement; or 2) expressed by the on-*event*-if-*condition*-then-*action* (ECA) paradigm. The enforcement can be refined for control, application, optimization, and exception handling. If the inclusion constraint is not class-based, but value-based, then support and enforcement become more challenging. For example, the *Student* type may use an attribute *Name,* which corresponds to a person's *Name* in a type *Person*.

Cardinality constraints. These have two main approaches to define their semantics: look-up and participation. Look-up works well for binary associations without relationship attributes. Participation constraints mix two different kinds of semantics with rigidity for extreme cases, despite the need to represent normal cases. 'Min/Max' captures the absolute extreme for all potential cases. The 'min' captures a (generalized) inclusion constraint; 'max' is intended to capture a (generalized) multiplicity constraint. For a relationship where the minimum participation could be '0' (someone is a student but not taking courses yet), a null value would be allowed in an implementation. However, a "*normal*" interpretation of the relationship is that a student must be registered for at least one course (null not allowed). Cardinality constraints impact other constraints in the schema [3].

Implicit constraints. Constraints can be implicit or hidden due to syntax construction. The eER modeling language uses relationship types with inherent (construction) inclusion and existence constraints as *based-on constraints*. Relationship objects reference their component objects; for example, entity objects. Therefore, the relationship objects can only exist if the corresponding entity object exists, making the semantics implicit, based upon the way in which relationships are constructed and used. They become explicit in the corresponding SQL specification.

Type semantics. eER modeling uses a Salami-slice strategy, oriented on the homogeneity of types and thus on decomposition into small, meaningful semantic units. Things in the application domain are multifaceted. A human is represented via a *Person* type that is separated from the *Student* type, which is associated via an IsA relationship (or subclass), to the *Person* type. At the same time, *Student* can be associated with other types, such as: *student_engagement*, *student_facilities*, *dormitory*, etc. Depending upon the view, a student might best be considered using the notion of a student or the notion of the more general object, person. Research has analyzed classification challenges [4].

Implicit representation of viewpoints. At the application level, it might be beneficial to consider user viewpoints that are represented as views [11]. For instance, a student might best be considered, including more general objects, e.g. person.

Separation of syntax and semantics. The separation of syntax and semantics is generally problematic. Most modelers learn a language using simple problems. However, real world problems are complex, so one language, or modeling technique, is not appropriate for all. It is impossible to represent a business problem at an application level of abstraction and implementation issues based on a singleton diagram. The problem is understanding and representing semantics.

Restricted and mixed semantics. Instead of general constraint frames, specific cases are often considered; e.g., mapping ratios (1:N, N:M, 1:1) to capture some binary relationship semantics. Sometimes, N:M ratios declare the maximum to be higher than 1. Look-up and participation cardinalities may be used with the same syntactic notion.

3 Models, Expressions, and Stakeholder Levels

Models and Conceptual Models. The notion of a *model* is complex and not necessarily well understood; similarly, for the process of modeling. Consider four perspectives: 1) the origins to be considered by the model; 2) the profile of the model (e.g. its function, purpose, or goal); 3) the stakeholders or the community of practice that the model must satisfy; and 4) the context within which the model and the origins are considered. The first two perspectives are internal; the second two, external.

A model is guided on its background [10]: the *grounding* of the model (paradigms, postulates, theories, culture, and conventions); and the *basis* for the model (e.g. languages used, concepts and conceptions, community, and commonly accepted practices). The *basis* of a model may change on demand. The perceptions of users might need to be represented in a model. Multiple coherent perceptions, a description of a system, or an augmented system might also be useful. A model can have many different purposes: to describe or explain a situation; specify and represent a concept someone has in mind; to aid in communication among stakeholders; or to decompose complex situations. A *model* is a well-formed, adequate and dependable artifact, commonly accepted by its community of practice within a given context [10], [11].

Semiotics of Signs: Icons, Symbols and Indexes. Semiotics, the study of the theory of signs, emphasizes the properties of things in their capacity. It is reasonable to apply semiotics to aid in this understanding since, before using a modeling language, it is first necessary to understand the language and its inherent bias.

Syntax refers to the arrangement of words in sentences and phrases. Syntax should be simple, parsimonious, and harmonic.

Semantics is concerned with the meaning of sentences and defines the interpretation of a sentence in the real world, depending on its context. It refers to the meaning of signs and what they represent in the real world.

Pragmatics considers the relationship between parts of sentences or signs and their users within a situation and context. It is user-dependent.

Although language is the main vehicle for modeling, semiotics is the background needed for understanding so that collaboration among stakeholders can result. Syntax, semantics, and pragmatics may follow different paradigms, leading to some effective use. The strictness of first-order predicate logic might be inappropriate during modeling. It is, however, needed in the final result. For example, natural utterances use the connective "and/or" with the meaning of logical OR. Similar observations can be made for all connectives, especially, for quantifiers.

Syntax has been well investigated for formal languages. Semantics can be defined in a variety of ways; e.g. for evaluation of variables, incorporation of context, scope of states, exceptions, and matching between syntactic language and semantic structure [8]. Problems arise when pragmatics is taken into consideration because the pragmatic interpretation depends on the community of practice, its culture, scope and attention.

Syntax, semantics and pragmatics of models are all important issues, and depend upon the needs of a model and its context.

Abstraction Levels of Stakeholders. At the application level, the perceptions of the users must be considered and combined with the context. At the conceptual modeling level, the resulting conceptual model must be based on what was developed at the application level. The logical level is typically based on an understanding of the platform, with the best practice being to use models that are mappings or compilations of the conceptual model.

4 Illustrative Example

A *conceptual database model* consists of a conceptual schema and a number of view schemata [11]. The view schemata are the result of transformations [1] [9] that map the viewpoints of the application level to sub-schemata of the conceptual schemata.

Consider a student-dormitory-course schema in Figure 1. Suppose a student is enrolled in several programs at a university. The dormitory association is dependent upon the program that a student takes. Specifically, a student lives in a dormitory that corresponds to the program (business, music, etc.) in which the student is enrolled. A student might obtain some financial support from a program, depending upon the level of completion of the program. A student makes courses that are required for a given program. The credit hours assigned to a course, may vary across courses, depending upon whether the course is intended for one program, or whether it is a mandatory or elective course. Any course can only be counted one time towards one program. A student is required to take a minimum number of classes per term. If a student fails a course, then the student may retake the course, up to a maximum of three times. A course has an associated tuition fee that must be within the limits of a given term, which may vary from term to term.

There are, however, some aspects of this situation that are difficult to model.

- A student can only take a course a maximum of three times. This might be overcome by adding a separate entity, called class or section, and a relationship; Course has Classes, with min/max cardinalities of (0,3) from student to class.
- A course can have different credit hours depending upon the program.
- A student can have multiple majors, which requires a decision about the dormitory to which a student should be assigned.
- The normal case for *enrolled in* does not capture freshmen who are not enrolled.
- The student must take courses that are required by the program.

These problems are at the application level. Someone must represent the university situation correctly and implement the corresponding results into a database. Also, involved is the end-user, a student. The database designer must attempt to models these in one conceptual model.

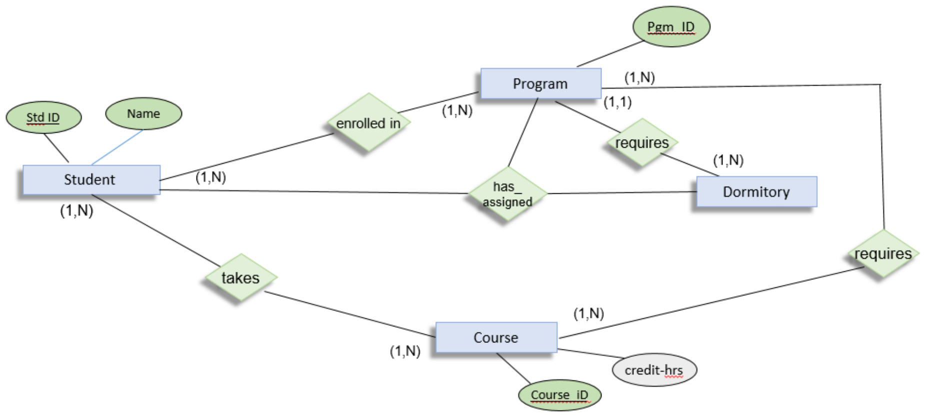


Fig. 1. Entity-Relationship Model of Student-Dormitory Application

5 Semiotics Reconsidered

Semantics and Pragmatics at the Application Level. Models at the application level have their own origins that they represent, profile, context, and community. The origins are consolidated perception models, enhanced by situation models that are commonly accepted in the application domain. Each community has a community-specific model; that is, a "local-as-design" approach. Objects under consideration are not homogeneous, for example, a department is considered together with its department head. Or, a student view incorporates all of the classes a student takes and refers to a university program class view from the university administration. A student is typically enrolled in one and only one program. There might be other students. Generalization and specialization follow natural semantics.

Models at this level of abstraction can be used at the conceptual level for communication and negotiation within and between communities of practice. Semantics and pragmatics differ based on the perception and understanding within the communities. Models may not be complete. Semantics may not be rigid. Objects are often considered to be holistic; for example, students together with their courses based on their programs. Therefore, we are not bound to normal data type construction. Constraints typically consider normal cases instead of extreme ones. Class planning might not require that students take classes, but student planning is based on the minimum and maximum credit hours a student must acquire in a given term.

Models at the application level have their own coherence. The underlying model allows us to integrate the different models. Models at the user level are typically not denotative but connotative, and follow cultural or community interpretations. For this reason, ontologies are appropriate for specifying domain-specific content [2].

Model Semantics at the Conceptual Level. A conceptual data model reflects, integrates and harmonizes the user views. Types specify homogeneous classes and are decomposed accordingly. The functionality definition is based on an entity-relationship algebra and given only after the structure model is complete. Constraints refine the structure; that is, semantics are defined only after the syntax is complete. The entity-relationship schema uses a diagram that is assumed to be complete, and represents its component at the same level of granularity and precision. Pragmatics tend to be hidden in a conceptual model, even though it is, in essence, an underlying model. It is assumed to be defined though external views.

Constraints at the Application Level and Conceptual Level. Constraints are generally considered valid for all of an application. However, a user's community might consider the 'normal' case or abstract (generalize) from exceptions, or omit them. Users use different scope, context, origins, and purposes. E.g., cardinality constraints represent some aspect, within specific semantics and pragmatics.

The Nature of Constraints. At the conceptual level, pragmatics must be handled by syntax and semantics. Cardinality constraints can do so, but are rigid and based on participation or lookup definition [7]. In the participation approach, extreme cases are included, in an attempt to represent exceptional cases. For example, an (1,N) constraint states that a corresponding relationship must exist for all entity classes. One solution is to use a harmonization of all user models and integrate them into the conceptual model. In this "global-as-design" approach, user views represent the external views of users, resulting in the challenge of properly representing finer semantics and pragmatics of these views. Due to the "local-as-view" design, constraints are introduced from the user's point of view. A conceptual model should harmonize all of these views to provide a holistic view of all constraints. A similar harmonization can occur at the logical level.

In Figure 1, a freshman could be enrolled in a program or not. If the freshman is enrolled, then a dormitory can be assigned based on the program enrolled. Later the freshman might also take courses. Then, a student is either a normal student, a student who does not take courses, or a student who does not have yet a dormitory. At the logical level, we can use tables for each of these specific cases and define a view that combines them. At the logical level, horizontal decomposition can be applied [10]. A relation type can be decomposed by selection expressions E1, …, En into separate types, provided this decomposition forms a partition on the class for this type. Therefore, we might also use a conceptual type, made up of conceptual base types. The base type has semantics without any context, but all subclasses are identified.

Objectives for Developing Better Constraints. Semantics can vary, depending on the user. This results in problems when mapping to a conceptual model, so the conceptual model should be more flexible. In most practices, normalization deals with the exceptional case where semantics causes a change of structure and the schema. That is, semantics drives syntax, in contrast to "semantics follows syntax." DBMS provide a much finer means for integrity maintenance. Maintenance can be deferred (eager or lazy integrity enforcement). Consistency can be supported at the row level. Integrity constraints can be maintained at the application level. Integrity can be made through views. Finally, flexible strategies may be used, besides the no-action and rollback approach; for example, on the basis of triggers or stored procedures.

These observations show that conceptual integrity constraints can be more elaborated if we can map the constraints to DBMS features. Here, we simply aim to show how semantics and syntax can be developed in a holistic approach. We further assume that pragmatics is defined at the application level, based on views, leading to the following observations and requirements.

(1) DBMS technology must provide a better way of treating syntax and semantics at the conceptual level, which captures pragmatics at the user level.
(2) A holistic view is needed for integrated usage of syntax together with semantics.
(3) Flexibility is required for changes needed to accommodate new technology.
(4) A mapping procedure for advanced integrity constraints should be supported.

Proposed Extensions of Integrity Constraints by Context as Part of Semantics.

1. *Actions* on a database are insert, delete and update for: a single object, one class, or objects tightly bundled via class inclusion constraints. Actions might be defined as an *action pattern*. This extends single-object actions to a complex object action while disabling the basic actions whenever a complex pattern exists.
2. The *scope pattern* is a view-defining query. This query defines either a single type view or, in general, the view schema on the conceptual schema.
3. *Enforcement style pattern* is for constraints that are timed as eager (default) or lazy (with(out) delay) enforcement, after an action (default), or as control before an action, with a level statement (e.g. DBMS, transaction, and interface levels).
4. *Reaction pattern* is for immediate enforcement or exception handling with a timed exit sub-pattern or timed enforcement, based on an enforcement obligation.

The above illustrates the need to deal with structure versus semantics. They can be formally defined and implemented. Then, in contrast to traditional approaches in which "semantics follows syntax," syntax and semantics may be treated as a whole.

Holistic View. A *conditional integrity constraint* is a pair of a context and a constraint. Constraints can be combined to partition a problem based on a scope pattern. For example, cardinality constraints Card(R,R') = (1,1) are for R = *enrolled_in*, and R' = *Student* with a selection predicate for: freshmen, a student who does not yet have an assigned dormitory, and students who did not yet take courses. The cardinality constraint is only valid for "normal" students. Adding an attribute *term* to the type *takes* could ensure that a student has not taken a course more than three times.

For example, for freshman with a dormitory, we may use a relaxed enforcement style. For freshman without a dormitory, we might use an interface style. That is, an insertion of such a student is only possible by an encapsulated insertion of the student, the programs, and the dormitory with a temporary insertion into the corresponding basic types; and a transfer of the object to another basic class whenever additional data are inserted. However, problems that exist or can be deduced for these constraints are not usually considered. All user needs cannot be represented by semiotics. View integration is difficult with global constraints, and usually completed based on user views. From a semiotics perspective, the user view should be considered as much as possible.

6 Conclusion

Many problems arise from the need to carry out modeling at multiple levels, depending upon the stakeholders. Since semiotics deals with language, it is proposed as an underlying basis from which to understand and capture semantics at different levels of abstraction. Additional conditional constraints are needed to model context, namely, action, scope, enforcement style and reaction.

Acknowledgements. This research was supported by the J. Mack Robinson College of Business, Georgia State University. Thanks to Melinda McDaniel for her assistance.

References

1. Embley, D. W., & Mok, W. Y. (2011). Mapping conceptual models to database schemas. In *Handbook of Conceptual Modeling* (pp. 123-163). Springer Berlin Heidelberg.
2. Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, *5*(2), 199-220.
3. Hartmann S. (1996). On the characterization and construction of entity-relationship database populations obeying cardinality constraints. PhD thesis, University Rostock. 1996.
4. Parsons, J., & Wand, Y. (2008). Using cognitive principles to guide classification in information systems modeling. *MIS Quarterly*, 839-868.
5. Storey, V. C. (1991). Relational database design based on the Entity-Relationship model. *Data & knowledge engineering*, *7*(1), 47-83.
6. Storey, V. C. (1993). Understanding semantic relationships. *The VLDB Journal*, *2*(4), 455-488.
7. Storey, V.C. (2005). Comparing relationships in conceptual modeling: mapping to semantic classifications. *IEEE Transactions on Knowledge and Data Engineering*, *17*(11), 1478-1489.
8. Schewe, K.-D. and B. Thalheim (2010): About Semantics. SDKB 2010: LNCS 6834, Springer, 1-22
9. Thalheim B. (2000). Entity-relationship modeling- Foundations of database technology. Springer, Berlin.
10. Thalheim, B. (2012). Syntax, semantics and pragmatics of conceptual modelling. In *Proc. NLDB 2012* 1-10. Springer Heidelberg.
11. Thalheim B. and M. Tropmann (2015). The conception of the conceptual database model. ER 2015, LNCS 9381, 603-611, Berlin, Springer.