Conceptual Model Notions – A Matter of Controversy

Conceptual Modelling and its Lacunas

Bernhard Thalheim[*,a]

[a] Department of Computer Science, Christian Albrechts University Kiel, Germany

Abstract. *The conception of a conceptual model is differently defined in Computer Science and Engineering as well as in other sciences. There is no common notion of this conception yet. The same is valid for the understanding of the notion of model. One notion is:* A model is a well-formed, adequate, and dependable instrument that represents origins and functions in some utilisation scenario. *The* conceptual model of an information system *consists 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. In a nutshell, a* conceptual model *is an enhancement of a model by concepts from a concept(ion) space.*
The variety of notions for conceptual model is rather broad. We analyse some of the notions, systematise these notions, and discuss essential ingredients of conceptual models. This discussion allows to derive a research program in our area.

Keywords. Model • Conceptual Model • Concept and Notion of a Model • Art of Modelling

1 What is a Conceptual Model

Modelling is a topic that has already been in the centre of research in computer engineering and computer science since its beginnings. It is an old sub-discipline of most natural sciences with a history of more than 2.500 years. It is often restricted to Mathematics and mathematical models what is however to much limiting the focus and the scope. Meanwhile it became a branch in the Philosophy of Science. The number of papers devoted to modelling doubles each year since the early 2000's.

It is often claimed that there cannot be a common notion of model that can be used in sciences, engineering, and daily life. The following notion covers all known so far notions in agriculture, archaeology, 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 that are deployed in certain scenarios (see Thalheim and Nissen 2015). A commonly acceptable statement for a general model notion is the following one[1]:
A model is a *well-formed*, *adequate*, and *dependable* instrument that represents *origins* and *functions* in some utilisation *scenario*. 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 fulfils in *utilisation scenarios*. The function determines the purposes and goals.

CS-conceptual modelling[2] is often related back to the introduction of the entity-relationship

* Corresponding author.
E-mail. thalheim@is.informatik.uni-kiel.de

[1] We refer to the model-to_model-modelling compendium (see Thalheim and Nissen 2015) for notions that are not introduced in this paper.
[2] In the paper we restrict ourselves to this kind of conceptual model and thus omit the CS acronym. In general, a conceptual model is a representation of a system in its widest sense on

model(ling language) for information systems development. It surprises nowadays that there is no commonly accepted notion of conceptual model yet. There have been several trials but none of them was sufficient and was able to cover the idea of the conceptual model.

The database and information systems research communities are extensively using the term "conceptual model"[3]. The notion of conceptual model still needs some clarification: what is a conceptual model and what not; which application scenario use which kind of conceptual model; is conceptual modelling only database modelling; do we need to have an understanding of modelling; is a conceptual database model only a reflection of a logical database model; is a conceptual model a model or not; etc. Let us illustrate the wide spread and understanding of conceptual models, the activity of conceptual modelling, and the modelling as a scientific and engineering process by some examples[4,5]:

Reality and world description: Conceptual modelling is the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication. Such descriptions, often referred as conceptual schemata, require the adoption of a formal notation, a conceptual model in our terminology[6]. (see Mylopoulos 1992)

Community description : Conceptual modelling is about describing the semantics of software applications at a high level of abstraction[7]. Specifically, conceptual modellers (1) describe structure models in terms of entities, relationships, and constraints; (2) describe behaviour 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 and information exchanged. In their typical usage, conceptual-model diagrams are high-level abstractions that enable clients and analysts to understand one another, enable analysts to communicate successfully with application programmers, and in some cases automatically generate (parts of) the software application. (see ER community 2017)

Conceptual database modelling: A data model is a collection of concepts that can be used to describe a set of data and operations to manipulate the data. When a data model describes a set of concepts from a given reality, we call it a conceptual model. (see Batini et al. 1992; Elmasri and Navathe 2000[8])

Instance-integrating conceptual modelling: A conceptual model consists of a conceptual schema

the basis of concept(ion)s that allow people to consciously act and being guided in certain situations of their systems.

[3] Facetted search for the term "conceptual model" in DBLP results in more than 5.000 hits for titles in papers (normal DBLP search also above 3.400 titles).

[4] The notion of conceptualisation, conceptual models, and concepts are far older than considered in computer science. The earliest contribution to models and conceptualisations we are aware of is pre-socratic philosophy.

[5] Wikiquote (see Wikiquote 2017) lists almost 40 notions. We add our list to this list.

[6] And continuing: These terms are introduced by analogy to data models and database schemata. The reader may want to think of data models as special conceptual models where the intended matter consists of data structures and associated operations.

[7] Some research challenges in conceptual modelling: Provide the right set of modelling constructs at the right level of abstraction to enable successfully communication among clients, analysts, and application programmers. Formalize conceptual-modelling abstractions so that they retain their ease-of-communication property and yet are able to (partially or even fully) generate functioning application software. Make conceptual modelling serve as analysis and development tools for exotic applications such as: modelling the computational features of DNA-level life to improve human genome understanding, annotating text conceptually in order to superimpose a web of knowledge over document collections, leveraging conceptual models to integrate data (virtually or actually) providing users with a unified view of a collection of data, extending conceptual-modelling to support geometric and spatial modelling, and managing the evolution and migration information systems. Develop a theory of conceptual models and conceptual modelling and establish a formal foundation of conceptual modelling.

[8] Another version is the following one: The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. A conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. A high-level data model or an implementation data model can be used at this level.

and an information base. A conceptual schema provides a language for reasoning about an object system, and it specifies rules for the structure and the behaviour of the system. A description of a particular state is given in an information base, which is a set of type and attribute statements expressed in the language of the conceptual schema. (see Boman et al. 1997)

System-representation models: A conceptual model is a descriptive model of a system based on qualitative assumptions about its elements, their interrelationships, and system boundaries. (see Business dictionary 2017)

Representational models: A conceptual model is a type of diagram which shows of a set of relationships between factors that are believed to impact or lead to a target condition; a diagram that defines theoretical entities, objects, or conditions of a system and the relationships between them. (see WordNet dictionary 2017)

Enterprise modelling and conceptual modelling: A conceptual is a model which represents a conceptual understanding (i. e. conceptualisation) of some domain for a particular purpose. A model is an artefact acknowledged by the observer as representing some domain for a particular purpose. (see Bjeković 2017)

Holistic view: In most cases, a model is also a conceptual model [9]. (see Pastor 2016)

Conceptual models as a result of an activity: We use the name of conceptual modelling for the activity that elicits and describes general knowledge a particular information system needs to know. The main objective of conceptual modeling is to obtain that description, which is called a conceptual schema. (see Olivé 2007)

[9] The slides of the keynote talk state: A conceptual model is a simplification of a system built with an intended goal in mind.
An abstraction of a system to reason about it (either a physical system or a real or language-based system). A description of specification of a system and its environment for some purpose. One main conclusion that we can reach is that the distinction between "model" and "conceptual model" is not always as precise as it should be.

Purpose-oriented modelling: Conceptual modelling is about abstracting a model that is fit-for-purpose and by this we mean a model that is valid, credible, feasible and useful. (see Robinson 2010)

Documentation-oriented conceptual model: A conceptual data model is a summary-level data model that is most often used on strategic data projects. It typically describes an entire enterprise. Due to its highly abstract nature, it may be referred to as a conceptual model. (see InfoAdvisors 2017)

Semiotics viewpoint: Conceptual modelling is about describing syntax, and semantics (potentially also pragmatics) of software applications at a high level of abstraction. (see Embley and Thalheim 2011)

Documentation and understanding viewpoint: A conceptual model of an application is the model of the application that the designers want users to understand. By using the application, talking with other users, and reading the documentation, users build a model in their minds of how to use the application. Hopefully, the model that users build in their minds is close to the one the designers intended. (see Johnson and Henderson 2013)

Conceptualisations of models: 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 [10] consists of concepts (see Murphy 2001) 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. (see Thalheim 2017)

At the ER'2017 conference a special brainstorming and discussion session has been organised with the task to coin the notion of a conceptual model. It seems to be surprising that there is no commonly accepted notion of a conceptual model after more than 40 years of introduction of this concept into database research. One proposal of the brainstorming discussion was:

[10] We follow R.T. White (see Thalheim 2014; White 1994) and distinguish between concepts, conceptual, conceptional, and conceptions.

ER 2017 discussion proposal: A conceptual model is a partial representation of a domain that can answer a question.

As for a model, the purpose dimension determines the quality characteristics and the properties of a model.

In a nutshell, a *conceptual model* is an enhancement of a model by concepts from a concept(ion) space. It is formulated in a language that allows well-structured formulations, is based on mental/perception/domain-situation models with their embedded concept(ion)s, and is oriented on a modelling matrix that is a common consensus within its community of practice.

We thus meet a good number of challenges, e.g. the following ones: is there any acceptable and general notion of conceptual model; do conceptual models really provide an added and sustainable value; what are the differences between conceptual models and models; what is a model; what means conceptualisation; how to support language-based conceptual modelling; etc. This paper is oriented on these questions and tries to develop an answer to them. We restrict the investigation to conceptual models in computer science and computer engineering and thus do not consider conceptual modelling for product design, service design, other system's design, natural and social sciences. Physical conceptual models are also left out of scope.

2 Revisiting Conceptual Modelling

2.1 State-Of-Art and State-Of-Needs

Modelling offers the benefit of producing better and understandable systems. It is based on a higher level of abstraction compared to most programming languages. Whether a model must be formal is an open question. The best approach is to consider model suites (or ensembles) that consist of a coherent collection of models which are representing different points of view and attention. We observe a resurgence in domain specific approaches that are challenged by technical, organisational and especially language design problems. UML is not the solution yet because UML Models are not executable but MDA needs them to be. The vast majority of UML models we have seen in industrial project are mere sketches and are informal and incomplete. They are not yet a viable basis for precise and executable models. Without precise models, no formal checking can take place. Therefore, these issues must be addressed either if modelling is well-accepted and gains significant presence in applications.

From the other side, the large body of knowledge on conceptual modelling in computer science is a results of hundreds of research papers over the last three-score years, although different names have been used for it. Modelling is often based on a finalised-model-of-the-real-world paradigm despite the constant change in applications. Model quality has already been considered in a dozen papers. Modelling literacy is rarely addressed in education. Models must however be reliable, refinable, and translatable artefacts in software processes.

Conceptual modelling is supported by a large variety of tools. e.g. (see Karagiannis et al. 2016). However, few of them support executable models. Of that few, far fewer still are actually rewarding to use. Conceptual models are acknowledged as mediators in the software development process. However, they are used and then not evolving with the evolution of the software. Reuse, migration, adaptation, and integration of models is still a lacuna. The lack of robust, evolution-prone and convenient translators is one reason. An environment as a constituent part for modelling and translation into consistent, easy-to-use and -revise, seamless, and industry-quality tool is still on the agenda. Information and software systems become eco-systems. Modelling eco-systems are not properly addressed yet.

Models are also used for communication based on some injection of a name space, while the community of practice uses a wealth of terms and terminology, with which they express their nuances of viewpoints. So, we need a number of representation models beside the singleton graphical representation. At the same time, models must be properly formal and based on rules strictly to be followed or else having the risk of making illogical statements. Thus, modelling must be based on methodologies.

2.2 Myths of (Conceptual) Modelling

Modelling and especially conceptual modelling is not well understood yet and misinterpreted in a variety of ways. It has brought a good number of myths similar to those known for software development (see Ambler and Sadalage 2006):

1. *Modelling is mainly for documentation*. The introduction of conceptual modelling for database systems has been motivated by a documentation scenario. A conclusion might be that modelling is a superfluous activity, especially in the case that documentation is not an issue.
2. *Modelling is finished with the use of the model and an initial phase*. Historic development of software started with requirements which were frozen afterwards and with modelling and specifications that were complete and became frozen before realisation begins.
3. *Modelling is only useful for heavyweight V-style software development*. Modelling and especially conceptual modelling is abandoned due to its burden and the discovery of the complexity of the software that is targeted.
4. *The collection of origins must be "frozen" before starting with modelling*. Models should be plastic and stable (one of the justification and thus dependability properties), i. e. the collection of origins to be modelled could change.
5. *The model is carved in stone and changes only from time to time if at all*. The realisation becomes 'alive' and thus meets continuous change requests. The model can have some faults, errors, misconceptions, misses etc. Extensions and additional services are common for systems. So, the model has to change as well.
6. *Modelling starts with selecting and accommodating a CASE tool*. Although CASE tools are useful, they impose their own philosophy, language, and treatment. Moreover, CASE tools allow to become too detailed. Instead, conceptual modelling should allow to create the model that is simple as possible and as detailed as necessary.
7. *Conceptual modelling is a waste of time*. Developers are interested in quick success and have their own perception model in mind. It seems to be superfluous to model and better to focus solely on how to write the code.
8. *Conceptual data modelling is a primary concern*. Data- and structure-driven development without consideration of the usage of the data in applications results in 'optimal' or 'normalised' data structure models and bad database performance. One must keep in mind the usage of the data, i. e. use a co-design method, e.g. (see Thalheim 2000).
9. *The community of practice has a common understanding how to conceptually model*. Modelling skills evolve over years and are based on modelling practice and experience. Further, conceptual models are based on a common domain-situation model that has to be shared within the community of practice. So, the perception models of modellers should match.
10. *Modelling is independent on the language*. Modelling cannot be performed in any language environment. Language matters, enables, restricts and biases (see Whorf 1980).

Understanding these and other myths allows to better understand the modelling process and the models. One way to overcome them is the development of sophisticated and acknowledged frameworks. Model-centred development (see Mayr et al. 2017) uses models as a kernel for the development of systems. Conceptual modelling is still taught as modelling in the small whereas modelling in the large is the real challenge.

2.3 Specifics of Notions

Let us return to the list of notions given in Section 1. Each of these notions has its graces, biases, orientations, applicability, acceptability, and specifics.

Scopes of conceptual models may vary from very general models to fine-grained models. General models allow to reason on system properties whereas fine-grained models serve as a blueprint for development.

Result-oriented viewpoint: Conceptual models can be seen as the final result and documentation of an activity that follows a certain development strategy such as agile, extreme, waterfall etc. methodologies.

Communication viewpoint: Conceptual models are a means for communication and negotiation among different stakeholders.

System construction orientation: Database, information and software system development is becoming more complex, more voluminous, requires higher variety, and changes with higher velocity. So a quick and parsimonious comprehension becomes essential and supports higher veracity and an added value for the system itself.

Perception and domain-situation models are specific mental models either of one member or of the community of practice within one application area. It is not the real world or the reality what is represented. It is the common consensus, world view and perception what is represented.

Conceptual models as documentation: Models provide also quality in use, i. e. they allow to survey, to understand, to negotiate, and to communicate.

Conceptual modelling with prototypes: Models can be enhanced by prototypes or sample populations. A typical approach is sample-based development (see Halpin 2009).

Visualisation issues: Conceptual models may be combined with representation models, e.g. visualisation models on the basis of diagrammatic languages.

Biased conceptual modelling approaches: Conceptual models are often models with a hidden background, especially hidden assumptions, that are commonly accepted in a community of practice in a given context and utilisation scenario.

Semiotics and semiology of conceptual modelling: Conceptual models are often language-based. The language selection is predetermined and not a matter of consideration in the modelling process.

Quality models: Conceptual models should be well-formed and satisfy quality requirements depending on their function in utilisation scenarios.

Concepts, conceptions: The elements in a conceptual models are annotated by names from some name space. These names provide a reference to the meaning, i. e. a reference to concepts and conceptions in a concept space.

Conceptual model suites: Models can be holistic or consist of several associated models where in the latter case each of them represents different viewpoints. For instance, a conceptual database model consists of a schema and a number of derived views which represent viewpoints of business users.

Normal models: Conceptual models represent only certain aspects and are considered to be intentionally enhanced by elements that stem from common sense, consensuses, and contexts.

A normal model (called 'lumped' model in Zeigler et al. 2000) is a part of the model that is considered to be essential and absolutely necessary. The *normal model* has a context, a community of practice that puts up with it, a utilisation scenario for which is is minimally sufficient, and a latent – or better deep – model on which it is based (see Zeigler et al. 2000 for 'base' model). The *deep model* combines the unchangeable part of a model and is determined by the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and the basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, common sense) of modelling. The (modelling) *matrix* consists of the deep model and the modelling scenarios. The last ones are typically stereotyped in dependence on the chosen *modelling method*.

This variety of viewpoints to conceptual models illustrates the different requirements and objectives of models. So, we might ask whether a common notion of a conceptual model exists or whether we should use different notions.

2.4 Problems and Challenges

Conceptual modelling techniques suffer from a number of weaknesses. These weaknesses are are mainly caused by concentration on database modelling and by non-consideration of application domain problems that must be solved by information systems. We follow the state-of-the-art analysis of A. van Lamsweerde (see van Lamsweerde 2000, 2008) who gave a critical insight into software specification and arrive with the following general weaknesses for conceptual modelling of information and database systems:

Limited scope. The vast majority of techniques are limited to the specification of data structuring, that is, properties about what the schema of the database system is expected to do. Classical functional and non-functional properties are in general left outside or delayed until coding.

Poor separation of concerns. Most modelling approaches provide no support for making a clear separation between (a) intended properties of the system considered, (b) assumptions about the environment of this system, and (c) properties of the application domain

Low-level schematology. The concepts in terms of which problems have to be structured and formalized are concepts of modelling in the small - most often, data types and some operations. It is time to raise the level of abstraction and conceptual richness found in application domains.

Isolation. Database modelling approaches are isolated from other software products and processes both vertically and horizontally. They neither pay attention to what upstream products in the software might provide or require nor pay attention to what companion products should support nor provide a link to application domain description.

Poor guidance. The main emphasis in database modelling literature has been on suitable sets of notations and on a posteriori analysis of database schemata written using such notations. Constructive methods for building correct models for complex database or information systems in a safe, systematic, incremental way are by and large non-existent.

Cost. Many information systems modelling approaches require high expertise in database systems and in the white-box use of tools.

Poor tool feedback. Many database system development tools are effective at pointing out problems, but in general they do a poor job of (a) suggesting causes at the root of such problems, and (b) proposing better modelling solutions.

Modern modelling approaches must not start from scratch. We can reuse achievements of database modelling in a systematic form and thus maintain theories and technologies while supporting new paradigms.

Constructiveness. Models of information systems can be built incrementally from higher-level ones in a way that guarantees high quality by construction. A method, is typically made of a collection of model building strategies, paradigm and high-level solution selection rules, model refinement rules, guidelines, and heuristics. Some of them might be domain-independent, some others might be domain-specific.

Support for comparative analysis. Database models depend on the experience of the developer, the background or reference solutions on hand, and on preferences of developers. Therefore, the results within a team of developers might need a revision or a transformation to a holistic solution. Beyond modelling qualities we may develop precise criteria and measures for assessing models and comparing their relative merits.

Integration. Tomorrow's modelling should care for the vertical and horizontal integration of models within the entire analysis, design, development, deployment and maintenance life cycle - from high-level goals to be supported by appropriate architectures, from informal formulation of information system models to conceptual models, and from conceptual models to implementation models and their integration into the deployment of information systems.

Higher level of abstraction. Information systems modelling should move from infological design to holistic co-design of structuring, functionality, interactivity and distribution. These techniques must additionally be error-prone due to the complexity of modern information systems. These abstraction techniques may be combined with refinement techniques similar to those that have been developed for abstract state machines.

Richer structuring mechanisms. Most modelling paradigms of the modelling-in-the-small approach, available so far for modularising large database schemata, have been lifted from software engineering approaches, e. g., component development. Problem-oriented constructs are developed as well as model suites that provide a means for handling a variety of models and viewpoints.

Extended scope. Information system development approaches need to be extended in order to cope with the co-design of structuring, functionality, interactivity and distribution despite an explicit treatment of quality or non-functional properties.

Separation of concerns. Information system modelling languages should enforce a strict separation between descriptive and prescriptive properties, to be exploited by analysis tools accordingly.

Lightweight techniques. The use of novel modelling paradigms should not require deep theoretical background or a deep insight into information systems technology. The results or models should be compiled to appropriate implementations.

Multi-paradigm modelling. Complex information systems have multiple facets. Since no single modelling paradigm or universal language will ever serve all purposes of a system. The various facets then need to be linked to each other in a coherent way.

Multilevel reasoning and analysis. A multi-paradigm framework should support different levels of modelling, analysis, design and development - from abstract and general to deep-level analysis and repairing of detected deficiencies.

Multi-format modelling. To enhance the communicability and collaboration within a development and support team, the same model fragment must be provided in a number of formats in a coherent and consistent way.

Reasoning in spite of errors. Many modelling approaches require that the model must be complete before the analysis can start. We claim that is should be made possible to start analysis and model reasoning much earlier and incrementally.

Constructive feedback from tools. Instead of just pointing out problems, future tools should assist in resolving them.

Support for evolution. In general, applications keep evolving due to changes in the application domain, to changes of technology, changes in information systems purposes etc. A more constructive approach should also help managing the evolution of models.

Support for reuse. Problems in the application domain considered are more likely to be similar than solutions. Models reuse should therefore be even more promising than code reuse.

Measurability of modelling progress. To be more convincing, the benefits of using information models should be measurable as well as their deficiencies.

This list of theories, solutions and methodological approaches is not exhaustive. It demonstrates, however, that modelling in the large and modern information systems modelling require specific approaches beyond integration of architectures into the analysis, design and development process.

2.5 The Research Issue

Let us reconsider the notions presented in Section 1. Table 1 compares essential properties of models. Missing model elements are denoted by n(ot).g(iven).

We observe that dependability is often either implicit or not considered in the model notion. Implicitness is mainly based on the orientation to normal models. The model matrix and especially the deep model are considered to be agreed before developing the model.

The origin is too wide in most cases. Models are not oriented towards representing some reality or the world. They are typically based on some kind of agreement made within a community of practice and according to some context, i. e. they reflect some domain-situation model[11] or more generally some mental model[12]. They might represent a perception model of some members of the community practice. They say what the phenomena in the given domain are like.

Table 1 directs to a conclusion that the function is mainly oriented towards description and partially prescription for systems development. The notion of the conceptual model has, however, mainly considered in system construction scenarios.

Concepts are often hidden behind the curtain of conceptual models. A conceptual model does not reflect the reality. Instead it reflects the mental understanding within its utilisation scenario.
These observations show now directly some open issues that should be solved within a theory and practice of conceptual modelling. Let us state some of them.

Research question 1.
What are the *origins* for conceptual models? Are these mainly domain-situation and perception models from one side and systems on the other side?

Research question 2.
How tightly are conceptual models bound to their *modelling matrix* and especially their *deep model*? To what extent are conceptual models normal models that are intentionally combined with their deep models?

Research question 3.
Which functions have conceptual models in which utilisation scenarios? Which properties must be satisfied by conceptual models in these scenarios? Which purposes and goals can be derived?

Research question 4.
What is the role of the *community of practice* in conceptual modelling? Which kind of model supports which community in which context?

Research question 5.
Conceptual modelling is less automated and more human dependent than any other development, analysis, and design process for information systems. It is a highly creative process. Is there any formalisation and foundation for this process?

[11] We restrict consideration to our field and thus to domain-oriented models. These models describe the application domain and more specifically the understanding, observation, and perception of an application domain that is accepted within a community of practice. In general, a situation model is a mental representation of a described or experienced situation in a real or imaginary world (see Radvansky and Zacks 1997).

[12] Mental models are out-of-scope in this paper. Those consist of an evolving model suite with small-scale and parsimonious models carried in human head (see Forrester 1971; Johnson-Laird 1983). They support various kinds of observation, information acquisition and filtering, reasoning, storage and information (de)coding, and communication. They are dependent on the observations, imaginations, and comprehension a human has made. Unlike conceptual models, mental models must neither be accurate, nor complete, and not consistent.

Table 1: Orientation of notions of conceptual models according model properties

version	adequate	dependable	origin	function	scenario	concepts
reality, world	reflection, truncation	formal, reflection	world	describe	communication, understanding	n.g.
community	abstraction, mapping	semantic invariance	software application	describe	construction	n.g.
conceptual database	mapping, homomorphy	n.g.	data, operations	describe	construction, documentation	reality concepts
system & instance	mapping, abstraction	n.g.	system, objects	n.g.	construction	n.g.
system representation	reflection,	qualitative assumptions	system	describe	representation	system concepts
represen- tional	mapping	n.g.	relation- ships	represent	visualisation	impact factors
enterprise	mapping, abstraction	faithful	domain	purpose- determined	understanding	concept space
result of activity	mapping,	n.g.	system knowledge	describe	acquisition, elicitation	domain knowledge
purpose- oriented	abstraction purposeful	viable, fit	any	elicitate	n.g.	n.g.
documen- tation	summary, abstraction	n.g.	data system	represent, survey	strategy development	n.g.
semiotics	syntax abstraction	semantics, pragmatics	software application	describe	representation	n.g.
document understand	mapping	closeness	application	understand by users	design	n.g.
conceptualise	formal representation	semantics	any	describe	representation	concept(ion) space
ad-hoc	selective mapping	n.g.	domain	consider problem	solving	n.g.

Research question 6.
Since models must not be conceptual models (see models in Thalheim and Nissen 2015), we might ask whether there exists a set of characteristics or criteria that separate a conceptual model from a model that is not conceptual. What is the concept space that can be used for an enhancement of a model by concepts or conceptions?

3 The Nature of Models

3.1 The Notion of a (Conceptual) Model

The model is an utterance and also an imagination. As already stated above (see also Thalheim and Nissen 2015), a model is a *well-formed*, *adequate*, and *dependable* instrument that represents *origins* and *functions* in some utilisation *scenario*. A model is a representation of some origins and may consist of many expressions such as sentences. *Adequacy* is based on satisfaction of the purpose or function or goal, analogy to the origins 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 macro-model, e.g. for explanation, exploration, construction, documentation, description and prescription.

Models *function* as *instruments* or tools. Typically, instruments come in a variety of forms and fulfil 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.

Model development is often targeted on normal models and implicitly accepts the deep model. A model is developed for some modelling scenarios and thus biased by its modelling matrix. The deep model and the matrix thus 'infect' the normal model.

Within the scope of this paper, we concentrate on representation models as proxies. So, a model of a collection of origins, within some context, for some utilisation scenario and corresponding functions within these scenarios, and for a community of practice is

- a relatively enduring,
- accessible
- but limited
- internal and at the same time external
- representation of the collection of origins.

The model becomes *conceptual* by incorporation of concepts and conceptions commonly accepted, of ideas provided by members from the community of practice, or of general well-understood language-like semiotic components. One main utilisation scenario for a conceptual database model is system construction[13]. In this case, the conceptual model thus becomes predictively accurate for the system envisioned and technologically fruitful. The model is an utterance and also an imagination. Other scenarios for conceptual models are: system modernisation, explanation, exploration, communication, negotiation, problem solving, supplantation, documentation, and even theory development.

Conceptual models must not be limited to the representation of static aspects of systems. They can also be used for the representation of dynamic aspects such as business stories, business processes, and system behaviour. The carrier of representation is often some language. In this case, a conceptual model can be considered to be an utterance with a collection of speech acts. The model itself can be then build on well-formedness rules for its syntax, semantics, and pragmatics, or more general of semiotics and semiology. According to J. Searle (see Searle 1969), a speech act consists of uttering elements, referring and predicating, requesting activities, and causing an

[13] Notice however that the first introduction of conceptual data models has been oriented on a documentation scenario.

effect. Whether at all and which language is going to be used is a matter of controversy as well.

3.2 Facets of a Conceptual Model

1. The conceptual model is a result of a perception and negotiation process.
The conceptual model represents mental models, especially domain-situation models or a number of perception models. Domain-situation models represent a settled perception within a context, especially an application. Perception models might differ from the domain-situation model. They are personal perceptions and judgements of a member of the community of practice. Maturity of conceptual models is reached after the community of practice negotiated different viewpoints and has found an agreement.

2. The conceptual model represents its collection of origins.
Considerations, about what to model and what to model not, are expressed via the adequacy criteria, especially for analogy to its origins, for focusing on specifics of the origins, and also on well-formedness of the model. The conceptual model does not represent the real world or a problem domain. It is already based on perception models of users about this problem domain or on domain-situation models of a user community on this problem domain.

3. The conceptual model is an instrument.
The conceptual model is used in some utilisation scenario by its users. So it functions in this utilisation scenario. It should describe, in a more abstract way compared to the origins, how the user conceives it and thus does not target on describing the origins.

4. The deep model underpins the conceptual model.
The deep model consists of all elements that are taken for granted, are considered to be fixed, and are common within the context for the community of practice. Elements of this model are symbolic generalizations as formal or readily formalisable components or laws or law schemata, beliefs in particular heuristic and ontological models or analogies supplying the group with preferred or permissible analogies and metaphors, and values shared by the community of practice as an integral part and supporting the choice between incompatible ways of practising their discipline. There is no need to redevelop this model. So, the normal model only display those elements that are additionally introduced for the model.

5. The conceptual modelling matrix.
The modelling matrix combines the deep model with typical utilisation scenarios, that are accepted by a community of practice in a given context. It specifies a guiding question as a principal concern or scientific interest, that motivates the development of a theory, and techniques as the methods a developer uses to persuade the members of the community of practice to his point of view. Although often not explicitly stated, the model matrix consists of a number of components: the objectives, inputs (or experimental factors), outputs (or responses), content requests, grounding, basis, and simplifications. The matrix sets a definitional frame for the normal model. It might support modelling by model stereotypes. The agenda of the modelling method is derived from the matrix. The matrix determines also a specific treatment of adequacy and dependability for a model.

6. The performance and quality criteria.
The model is a persistent and justified artefact that satisfy a number of conditions according to its function such as 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. The quality characteristics bound the model to be valid, credible, feasible, parsimonious, useful, and at the same time as simple as possible and as complex as necessary.

7. The model is the main ingredient of a modelling method.
Sciences and technologies have developed their specific deployment of models within their investigation, analysis, development, design etc. processes. The deep model and the matrix are often

agreed. The central element of all modelling methods is the model, that is used as an instrument in scenarios which have been stereotyped for the given modelling method. The modelling method typically also includes the design of a representation model (or a number of such). The representation model of the (conceptual) model may be based on approaches such as diagramming and visualisation. It uses a set of predefined signs: icons, symbols, or indexes in the sense of Peirce.

3.3 Sources for Conceptual Models: Domain-Situation and Perception Models

The domain-situation model is build by a community of practice on a semantical level. It refers to the world-as-described-and-conceived-by-the-deep-model. It thus forms the deep understanding behind the conceptual model. This deep internal structure of the conceptualisation is commonly shared in the community, abstracts from accidental origins, uses a partial interpretation, exhibits (structural) hidden similarities of all origins under consideration, and presents the common understanding in the community. It gives thus a literal meaning to the domain. The context for the conceptual model is typically governed by domain-situation models. The domain-situation model is thus one source for the conceptual model.

A domain-situation model might or might not exist. It shapes, however, what is seen in an application domain and how to reason about what is seen. It represents some common negotiated understanding in the application domain. It may represent the application domain as it is, or the application domain as it makes sense to be characterised, categorised or classified in one way rather than another, given certain interests and aptitudes or more generally given certain background.

The second source for conceptual models is a collection of perception models, that are provided and acknowledged by members of this community of practice. A perception model is one kind of epistemological mental model with its verbal, visual and other information compiled on the basis of cognitive schemata. It organises, identifies, and interprets observations made by the member. It does not need to know the deep facts or essential properties of the origins in order to succeed in communicating about them or to reason. The perception model typically follows the situation that it represents. It is, however, often undetermined and thus may also partially contradictory. So it parallels and imitates parts of the reality ('Gestalt' notion of the model). They provide a partial understanding, refer to some aspect, may use competing sub-models about the same stuff, and may set alternatives on meaning. It is build by intuitive, discursive and evidence-backed perception, by imagination, and by comprehension. It is shaped by learning, memorisation, expectation, and attention. Perception models serve as an add-on beyond domain-situation models.

These two sources for conceptual models depend on the community of practice. So, different communities might use different kinds of verbal and non-verbal representation. Although they provide a literal meaning to the conceptual model, they must not be explicitly stated within the conceptual model. They serve as the origin for the conceptual model and thus might not be explicitly incorporated into the conceptual model. The conceptual model may have its deep background, i. e. its basis and especially its grounding.

Both origins are not complete. Typically the scope of both models is not explicit. There are unknown assumptions applied for description, unknown restrictions of the model, undocumented preferences and background of the community of practice, and unknown limitations of the modelling language. Classically, we observe for members of a community of practice that:

- they base their design decisions on a "partial reality", i. e. on a number of observed properties within a part of the application,
- they develop their models within a certain context,
- they reuse their experience gained in former projects and solutions known for their reference models, and

- they use a number of theories with a certain exactness and rigidity.

The conceptual model to be developed is deeply influenced by these four hidden factors.

4 Conceptualisation of Models

The domain-situation model and also partially the perception model are using concepts commonly. Conceptual models reuse such concepts from these origins and thus inherit semantics and pragmatics from these models. Further, conceptualisation may also be implicit and may use some kind of lexical semantics of these models, e. g. word semantics, within a commonly agreed name space.

4.1 Concepts and Conceptions

Various notions of concept has been introduced, for instance, by J. Akoka, P. Chen, H. Kangassalo, R. Kauppi, A. Paivio, and R. Wille (see Chen et al. 1998; Ganter and Wille 1998; Kangassalo and Palomäki 2015; Kauppi 1967; Paivio 1986). Artificial intelligence and mathematical logics use concept frames. Ontologies combine lexicology and lexicography. Concepts are used in daily life as a communication vehicle and as a result of perception, reasoning, and comprehension. Concept definition can be given in a narrative informal form, in a formal way, by reference to some other definitions etc. Some version may be preferred over others, may be time-dependent, may have a level of rigidity, is typically usage-dependent, has levels of validity, and can only be used within certain restrictions. We also may use a large variety of semantics (see Schewe and Thalheim 2008), e.g., lexical or ontological, logical, or reflective.

We distinguish two different meanings of the word ‘concept’ (see White 1994):

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.

Conceptions (see White 1994) are systems of explanation. They are thus more difficult to describe.

The typical definition frame we observed is based on definition items. These items can also be classified by the kind of definition. Concepts may have different descriptions simultaneously. A competing description may represent the same concept differently depending on the context (e. g. time, space), validity, usage, and preferences of members of the community of practice. A concept may have elements that are necessary or sufficient, that may be of certain rigidity, importance, relevance, typicality, or fuzziness. Based on the generalisations of the approach, that has been proposed by G.L. Murphy (see Murphy 2001; Thalheim 2007), concepts are defined in a more sophisticated form as a tree-structured structural expression.

```
SpecOrderedTree(StructuralTreeExpression
        (DefinitionItem,
          Modality(Sufficiency, Necessity),
          Fuzziness, Importance, Rigidity,
          Relevance,
          GraduationWithinExpression,
          Category))) .
```

Concept may be regarded as the descriptive and epistemic core units of perception and domain-situation models. These origins govern the way how a concept can be understood, defined, and used in a conceptual model. The conceptual model inherits thus concepts and their structuring within a concept space, i. e. conceptions.

4.2 Conceptualise

Conceptualisation and semantification are orthogonal concerns in modelling. *Conceptual modelling* is based on concepts that are used for classification of things. Concepts have fuzzy boundaries. Additionally, classification depends on the context and deployment. Conceptual[14] modelling uses *conceptions*, which are systems of explanation.

Semantification (see Duží et al. 2009) improves the comprehensibility of models and explicit reasoning on elements used in models. It is based on name spaces or ontologies which are commonly accepted in the application domain. Conceptual models are models enhanced by concepts and integrated in a space of conceptions.

Conceptualisation injects concepts or conceptions into models. These enriched models reflect those concepts from commonly accepted concept space. The concept space consists of a system of conceptions (concepts, theoretical statements (axioms, laws, theorems, definitions), models, theories, and tools). A concept space may also include procedures, conceptual (knowledge) tools, and associated norms resp. rules. Is is based on paradigms which are corroborated.

4.3 Dependability of Conceptual Models

Models must be dependable, i. e. justified from one side and and qualitatively certified from the other side. Justification can be based on the domain-situation and perception models and the relation of the conceptual models to these models. If, however, such models are not available or of low quality, justification will become an issue. Quality certification is an issue of pragmatism and of added value of the conceptual model. So, we target on a high quality conceptualisation.

Conceptualisation may be based on the seven principles of Universal Design (see Patil et al. 2003). Typical mandatory principles are usefulness, flexibility, simplicity, realisability, and rationality. Optional conceptualisation principles are perceptibility, error-proneness, and parsimony.

The *principle of conceptualisation* is considered to be one - if not the main - of the seven fundamental principles for conceptual modelling (see Griethuysen 2009). The other six principles are: Helsinki, Universe of discourse, searchlight, 100%, onion, and three level architecture principles. They can be questioned further. These principles can be enhanced by the principles of understanding, of abstraction, of definition, of refinement, evaluation, and of construction (see Thalheim 2010). Conceptualisation can be considered to be completed if: A conceptual schema should only include conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access, as well as all aspects of particular external user representation such as message formats, data structures, etc.
Based on Section 3.3, the principle of conceptualisation can be stated as follows:
A conceptual model should only include conceptually relevant aspects of the domain-situation and perception models. It does not consider neither aspects of realisation nor of representation. It includes, however, different viewpoints of business users and concepts from the common concept space.

[14] Conceptual modelling is performed by a modeller that directs the process based on his/her experience, education, understanding, intention and attitude. Conceptual models are using/incorporating/integrating concepts (see White 1994) Conceptional modelling aims at development of concepts.

5 Conclusion: Towards a Notational Frame for Conceptual Models

Conceptual modelling is not yet a science or culture. It is rather a craft or even an art. It can be learned similar to craft learning. It is however based on understanding and abstraction throughout the perception and domain-situation models, i. e. of mental models in general. Perception is dependent on deep models and thus incomplete, revisable, time-restricted, activity-driven, and context-dependent.

5.1 Slim, Light, and Concise Versions for Conceptual Models

Conceptual models are widely used in system construction scenarios. They function as description of the phenomena of interest within the context for its community of practice. So, conceptual models are normal models with rather specific modelling matrices and deep models. A slim notion of a conceptual model should only reflect such normal models and refer to a specific modelling matrix. A light version needs to refer to some elements of the basis and to some context. A concise version must explicitly represent all the hidden details of a model, especially its relationships to the concept space, to the perception of this space by members of the community of practice, and to the utilisation scenario.

5.2 A Proposal for a Light Version: Conceptual Model $\sqsupseteq$ Model $\bigoplus$ Concepts

Conceptual modelling is not yet a common method in science (see Robinson 2010). Systems can be build without any conceptual model. It seems that there is no need for a formal conceptual modelling process. It seems to be too restrictive to require a full conceptual model. Performance and quality criteria are not commonly agreed. The science of conceptual modelling is still missing.

The main bottleneck is however the missing notion of a conceptual model. The conceptual model is a specific model and is based on conceptualisation. It might be language-bound. It is probably the most important aspect of system construction in computer science and computer engineering. It is however the most difficult and least understood. Minimal justification characteristics of models are classical viability, i. e. corroboration, validity, credibility, rational coherent and conform, falsifiable, stability against origin collection change. Minimal quality characteristics of models are the one for quality in use (e.g. usability, aptness for the function and purpose, value for the utilisation scenario, feasibility). Minimal performance characteristics are timely, elegant and feasible usage within the given context for their community of practice according to their utilisation scenario and their competencies or more general their profiles.

So, we might conclude for a *light version*: *A conceptual model is a well-formed, adequate and dependable instrument that functions within its specific utilisation scenario, that represents origins, and that is enhanced by concepts from a concept(ion) space.*

Therefore, the incorporation of concepts and the conceptions is one main difference to the model.

5.3 Lacunas of Conceptual Modelling

Since conceptual modelling is still more an art than a science and a culture of conceptual modelling is still beyond the horizons, we need

- an understanding of the area of conceptual modelling;
- a theory, techniques, and engineering of conceptualisation;
- an integrated multi-view approach for the needs and the capabilities of the members of the community of practice;
- a refinable definition of the conceptual model with all three versions, i. e. a simplified version, a fully fledged version, and an assessable version;
- a working approach with intentional and thus latent matrices and deep models for daily practice; and
- an understanding of language use in conceptual modelling.

These lacunas do not limit usability, usefulness, and utility of conceptual models. Conceptual database models improve, from one side system, comprehension. They allow to indicate associations among system elements, reduce the effect of bad implementation, provide abstraction mechanisms, support prediction of system behaviour, provide an elegant and adequate overview of the system at various levels of abstraction, support the construction of different user views, and cross-reference multiple viewpoints. From the other side, they reduce developers, maintainers and programmers overhead. They support a simple and free navigation through components of the database system, provide an easy deduction of various viewpoints, that represent the needs of business users, support concentration and focusing in evolution and maintenance phases, display the decisions made during development, indicate opportunities for further development and system maintenance, reduce the effort by reuse of design and development decisions that have already been made, and use a comfortable and effective visualisation. So, conceptual models are not restricted to construction scenarios or to database modelling.

We realise, that the development and the acceptance of a notion of conceptual model follows the 13 commandments stated (see Bowen and Hinchey 2009):

1. Thou shalt choose an appropriate notation.
2. Thou shalt formalise but not over-formalise.
3. Thou shalt estimate costs.
4. Thou shalt have a formal methods guru on call.
5. Thou shalt not abandon thy traditional development methods.
6. Thou shalt document sufficiently.
7. Thou shalt not compromise thy quality standards.
8. Thou shalt not be dogmatic.
9. Thou shalt test, test, and test again.
10. Thou shalt reuse.
11. Thou shall meet intentions of all members of the community of practice
12. Thou shall provide a usable notation, i. e. for verification, validation, explanation, elaboration, and evolution.
13. Thou shall be robust against misinterpretation, errors, etc.

References

Ambler S., Sadalage P. (2006) Refactoring databases - Evolutionary database design. Addison-Wesley

Batini C., Ceri S., Navathe S. (1992) Conceptual database design (an entity-relationship approach). Benjamin/Cummings, Redwood City

Bjeković M. (2017) Pragmatics of Enterprise Modelling Languages: A Framework for Understanding and Explaining. PhD thesis, Radboud University Nijmegen

Boman M., Bubenko Jr. J., Johannesson P.and Wangler B. (1997) Conceptual modelling. Prentice Hall, London

Bowen J. P., Hinchey M. G. (2009) Ten Commandments Ten Years On: Lessons for ASM, B, Z and VSR-net. In: Rigorous Methods for Software Construction and Analysis. Lecture Notes in Computer Science Vol. 5115. Springer, pp. 219–233

Business dictionary (2017) Conceptual model. http://www.businessdictionary.com/definition/-conceptual-model.html. Last Access: Assessed Nov. 21, 2011

Chen P., Akoka J., Kangassalo H., Thalheim B. (eds.) Conceptual Modeling: Current Issues and Future Directions. LNCS Vol. 1565. Springer

Duží M., Heimburger A., Tokuda T., Vojtas P., Yoshida N. (2009) Multi-Agent Knowledge Modelling. In: Information Modelling and Knowledge Bases XX. Frontiers in Artificial Intelligence and Applications, 190. IOS Press, pp. 411–428

Elmasri R., Navathe S. (2000) Fundamentals of database systems. Addison-Wesley, Reading

Embley D., Thalheim B. (2011) Preface. In: The Handbook of Conceptual Modeling: Its Usage and Its Challenges. Springer, Berlin, pp. v–ix

ER community (2017) Homepage. http://conceptualmodeling.org. Last Access: Assessed Oct. 29, 2017

Forrester J. (1971) Collected papers of J.W. Forrester In: Wright-Allen Press, Cambridge chap. Counterintuitive behaviour of social systems, pp. 211–244

Ganter B., Wille R. (1998) Formal concept analysis - Mathematical foundations. Springer, Berlin

Griethuysen J. J. V. (2009) The Orange Report ISO TR9007 (1982 - 1987) Grandparent of the Business Rules Approach and SBVR Part 2 - The Seven Very Fundamental Principles. In: Business Rules Journal 10(8) http://www.brcommunity.com/a2009/b495.html

Halpin T. A. (2009) Object-Role Modeling. In: Encyclopedia of Database Systems. Springer US, pp. 1941–1946

InfoAdvisors (2017) What are Conceptual, Logical and Physical Data Models? http://www.datamodel.com/index.php/articles/what-are-conceptual-logical-and-physical-data-models/. Last Access: Assessed Nov. 21, 2011

Johnson-Laird P. (1983) Mental models: Towards a cognitive science of language, inference, and consciousness. Cambridge University Press, London

Johnson J., Henderson A. (2013) Conceptual modelling in a nutshell. http://boxesandarrows.com. Last Access: Assessed Jan. 29, 2013

Kangassalo H., Palomäki J. (2015) Defintional conceptual schema - The core for thinking, learning, and communication. Keynote given at 25th EJC Conference, Maribor, Slovenia

Karagiannis D., Mayr H. C., Mylopoulos J. (eds.) Domain-Specific Conceptual Modeling, Concepts, Methods and Tools. Springer

Kauppi R. (1967) Einführung in die Theorie der Begriffssysteme. In: Acta Universitatis Tamperensis, Ser. A, Vol. 15, Tampereen yliopisto, Tampere

Mayr H. C., Michael J., Ranasinghe S., Shekhovtsov V. A., Steinberger C. (2017) Model Centered Architecture. In: Conceptual Modeling Perspectives. Springer, pp. 85–104

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

Mylopoulos J. (1992) Conceptual modeling and Telos. In: Conceptual modeling, databases, and CASE: An integrated view of information systems development. LNCS 10509. Wiley, Chichester, pp. 49–68

Olivé A. (2007) Conceptual modeling of information systems. Springer, Berlin

Paivio A. (1986) Mental representations: A dual coding approach. Oxford University Press, Oxford

Pastor O. (2016) Conceptual Modeling of Life: Beyond the Homo Sapiens; Keynote at ER 2016. http://er2016.cs.titech.ac.jp/assets/slides/ER2016-keynote2-slides.pdf. Last Access: Assessed Nov. 15, 2016

Patil B., Maetzel K., Neuhold E. J. (2003) Design of International Textual Languages: A Universal Design Framework. In: IWIPS. Product & Systems Internationalisation, Inc., pp. 41–56

Radvansky G., Zacks R. (1997) Cognitive models of memory In: Convey M. (ed.) MIT Press, Cambridge chap. The retrieval of situation-specific information, pp. 173–213

Robinson S. (2010) Conceptual modelling: Who needs it? In: SCS M & S Magazine (2), pp. 1–7

Schewe K.-D., Thalheim B. (2008) Semantics in Data and Knowledge Bases. In: SDKB 2008. LNCS 4925. Springer, Berlin, pp. 1–25

Searle J. (1969) Speech Acts: An essay in the philosophy of language. Cambridge University Press, Cambridge, England

Thalheim B. (2000) Entity-relationship modeling – Foundations of database technology. Springer, Berlin

Thalheim B. (2007) The Conceptual Framework To User-Oriented Content Management. In: Information Modelling and Knowledge Bases. Frontiers in Artificial Intelligence and Applications Vol. XVIII. IOS Press

Thalheim B. (2010) Towards a Theory of Conceptual Modelling. In: Journal of Universal Computer Science 16(20) http://www.jucs.org/jucs_16_20/towards_a_theory_of+, pp. 3102–3137

Thalheim B. (2014) The Conceptual Model ≡ An Adequate and Dependable Artifact Enhanced by Concepts. In: Information Modelling and Knowledge Bases. Frontiers in Artificial Intelligence and Applications, 260 Vol. XXV. IOS Press, pp. 241–254

Thalheim B. (2017) General and Specific Model Notions. In: Proc. ADBIS'17. LNCS 10509. Springer, pp. 13–27

Thalheim B., Nissen I. (eds.) Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung. De Gruyter, Boston

van Lamsweerde A. (2000) Formal specification: a roadmap. In: ICSE - Future of SE Track, pp. 147–159

van Lamsweerde A. (2008) Requirements engineering: from craft to discipline. In: SIGSOFT FSE. ACM, pp. 238–249

White R. (1994) Commentary: Conceptual and conceptional change. In: Learning and instruction 4, pp. 117–121

Whorf B. (1980) Lost generation theories of mind, language, and religion. Popular Culture Association, University Microfilms International, Ann Arbor, Mich.

Wikiquote (2017) Conceptual model. https://en.wikiquote.org/wiki/Conceptual_model. Last Access: Assessed Nov. 21, 2017

WordNet dictionary (2017) Conceptual model. http://www.dictionary.com/browse/conceptual-model. Last Access: Assessed Nov. 21, 2011

Zeigler B. P., Kim T. G., Praehofer H. (2000) Theory of Modeling and Simulation. Elsevier Academic Press