*in: Stefan Strecker and Jürgen Jung (eds): Informing Possible Future Worlds*
*Logos Verlag Berlin GmbH, 2024*
*ISBN 978-3-8325-5768-3, DOI 10.30819/5768*
*https://www.logos-verlag.de/ebooks/OA/978-3-8325-5768-3.pdf*

**Chapter 4**

# Conceptual Modeling: A Still Unfinished Saga

## About Prejudices, Aberrations, Solutions and Challenges

**Heinrich C. Mayr and Bernhard Thalheim**

Conceptual modeling has a long and chequered history that goes far back into the past. In Computer Science, it has been discussed since the 70s of the last century, when – after some preceding work on data and process abstractions – the fundamental paper by Peter P.-S. Chen appeared in which he introduced the Entity-Relationship Model. In honor of the 65$^{th}$ birthday of Ulrich Frank who throughout his career has been deeply and also critically engaged with conceptual modeling, we present in this paper a few observations and thoughts on this history.

## 4.1 Introduction

This paper is intended as a little thank-you for Ulrich Frank, who has over many years decisively shaped the modeling scene, especially in Business Informatics, and surely will continue to do so. We would like to point out right away, however, that this is not a thoroughly serious, purely scientific paper. It is rather a collection of observations, questions, theses, and examples on the state of modeling which we have dealt with in the course of our many years of involvement with modeling ans partly also discussed in keynote speeches at various occasions. Accordingly, we make no claim to completeness: There are many things we do not mention, for example because we do not know anything about them, because we did not think they were worth mentioning, or because we simply forgot about them. Ulrich's 65th birthday, on which we extend our sincere congratulations, is a welcome occasion to present this thank-you, as he has contributed significantly to a better understanding and foundation of modeling, especially through his critical analyses but also with his constructive concepts. Modeling and modeling methods are crucial to most disciplines. We cannot do without them, even if they are not always appreciated, often underestimated, and in practice often dismissed with '*it's useless*'. After all, modeling has been widely researched, taught, and also practiced with a certain systematic for many years. However, many concepts are regularly 'reinvented' or renamed, many mistakes are made again and again, and prejudices are readily cultivated.

But there is progress: both in the theoretical foundation and in the realization of what is needed for and in practice. We will sketch a picture of this here – with a focus on more recent developments. Of course, this cannot be done without some recourse to the gray

Middle Ages of the early Computer Science years and a sketch of the essence of modeling and its dimensions.

The paper is structured as follows: We first address some basic work that interested people either already know or can or should read up on. Then we try to answer the question '*Where do we stand?*' and contrast this with the situation in 1998, the year of the foundation of the German-speaking Modeling Conference, in which Ulrich Frank always played an important role. From this, we try to derive some hints for the future and shortly explain our contribution to the foundation of conceptual modeling by means of the Triptych paradigm (Mayr and Thalheim 2021).

## 4.2 Basic Work

Every discipline has its own foundational works – though, of course, opinions often vary widely about what is important and what is not. So, in order to not only out our own assessments, we reproduce here a list of works identified in a survey[1] conducted in 2021: This survey was distributed worldwide among representatives of the data, process, and software modeling communities and asked, among other things, what participants thought was the most important foundational work in their field.

A total of 153 participated in this survey and mentioned the following (at least twice each, in alphabetical order by first author):

- Batini, C., Ceri, S., Navathe, S. B. (1992). *Conceptual Database Design: An Entity-Relationship Approach* (Batini et al. 1992)
  The first comprehensive and systematic presentation of the ER methodology of database modeling for the development of database structures.
- Brambilla, M., Cabot, J., Wimmer, M. (2017). *Model-driven software engineering in practice* (Brambilla et al. 2017)
  The introductory book follows Stachowiak (1973) in an object-oriented setting and discusses the foundations of model-driven software engineering as well as its technical aspects.
- Broy, M., Stølen, K. (2012). *Specification and development of interactive systems: focus on streams, interfaces, and refinement* (Broy and Stølen 2001)
  A systematic and well-founded explanation for stepwise specification and development of interactive software and systems as programming in the large (FOCUS message interaction method).
- Carmona, J., van Dongen, B., Solti, A., Weidlich, M. (2018). Conformance checking (Carmona et al. 2018)
  The book investigates the process mining perspective of the relation between modelled behaviour and recorded behaviour of processes.
- Chen, P.P.S. (1976). *The entity-relationship model – toward a unified view of data* (Chen 1976) (reprinted in Embley and Thalheim 2012)
  The initial paper that introduces the notion of conceptual model(ing language) as documentation model for implementation-oriented relational database schemata, further used for the description modeling approach.

1 Michael, J.; Bork, D; Wimmer, M.; Mayr, H. C.: Quo Vadis Modeling? Findings of a Community Survey, an Ad-hoc Bibliometric Analysis, and Expert Interviews on Data, Process, and Software Modeling. To appear in the International Journal on Software and Systems Modeling (SoSym), Springer.

- Codd, E.F. (2002). *A relational model of data for large shared data banks* (Codd 2002)
  The initial paper on the relational approach to database structure everybody in the area should know.
- Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (2013). *Fundamentals of business process management* (Dumas et al. 2013)
  The book distills the entire business process management landscape.
- Elmasri, R., Navathe, S.B. (2000). *Fundamentals of Database Systems* (Elmasri and Navathe 2000)
  One of the main textbook for database analysis, design, and development based on the classical entity-relationship approach.
- Embley, D. W., Thalheim, B. (Eds.). (2012). *Handbook of conceptual modeling: theory, practice, and research challenges* (Embley and Thalheim 2012)
  A survey collection on different approaches, languages, and foundations to conceptual modeling.
- Ferstl, O. K., Sinz, E. J. (2015). *Grundlagen der Wirtschaftsinformatik* (Ferstl and Sinz 2015)
  The systematics of business information systems as the core of Business Informatics is systematically described for the level of tasks, the carriers of these systems and the design and operation.
- Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction* (Friedman 2017)
  A systematic presentation of the statistical basis and algorithmics of systematic learning from data.
- Gamma, E., Johnson, R., Helm, R., Johnson, R. E., Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software* (Gamma et al. 1994)
  One of the first systematic representations of generic object-oriented programming solutions as classes of pattern.
- Guarino, N. (1994). *The ontological level* (Guarino 1994)
  Ontologies should be used for improving the quality of knowledge bases.
- Halpin, T., Morgan, T. (2010). *Information modeling and relational databases* (Halpin and Morgan 2010)
  A fully developed representation of the fact-oriented NIAM/ORM approach to conceptual modeling of database structures.
- Karagiannis, D., Mayr, H.C., Mylopoulos, J. (2016). *Domain-specific conceptual modeling* (Karagiannis et al. 2016)
  A survey collection on various languages and approaches for conceptual modeling that can be used for model development within the ADOxx realisation toolbox.
- Kent, W. (1978). *Data and reality* (Kent 1978)
  The classical book explains how human beings perceive and process information about the world we operate in, and how we struggle to impose that view on our data processing machines in whatever modeling language.
- Olivé, A. (2007). *Conceptual modeling of information systems* (Olivé 2007)
  A systematic presentation of the object-oriented approach to modeling database structures.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie* (Stachowiak 1973)
  One of the most fundamental – but internationally little noticed – books on modeling with three properties characterising models: mapping, abstraction, pragmatic.
- Thalheim, B. (2000). *Entity-relationship modeling: foundations of database technology* (Thalheim 2000)
  A systematic compilation of achievements, extensions, and theory of entity-relationship modeling languages and their applications.

- Weske, M. (2007). *BPM – Concepts, Languages, Architectures* (Weske et al. 2007)
  A systematic explanation of the complete business process management lifecycle from the modeling phase to process enactment and improvement.
- van der Aalst, W.: *Process Mining – Data Science in Action* (Van Der Aalst 2016)
  Compiling approaches to process mining as the missing link between model-based process analysis and data-oriented analysis techniques.

In our personal opinion, however, at least the following works should also be mentioned:

- Becker, J., Probandt W., Vering, O. (2012). *Grundsätze ordnungsmäßiger Modellierung - Konzeption und Praxisbeispiel für ein effizientes Prozeßmanagement* (Becker et al. 2012)
  Principles of proper models are correctness, relevance, economic efficiency, clarity, comparability, and a systematic structure.
- Cabot, J., Vallecillo, A. (2022). *Modeling should be an independent scientific discipline* (Cabot and Vallecillo 2022)
  A vision according to which modeling is to be developed into an independent sub-discipline within computer science with high disciplinary gains.
- Frank, U. (2014). *Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges* (Frank 2014)
  Multilevel multi-perspective modeling is a powerful object-oriented framework for coherent co-development of models at various abstraction levels (meta …meta-meta-meta) with integrated perspectives (design, engineering, development, user, economic, enterprise).
- Guarino, N., Guizzardi, G., Mylopoulos, J.(2019): *On philosophical foundations of conceptual Models* (Guarino et al. 2019)
  The paper proposes an ontology-backed general grounding to conceptual modeling.
- Henderson-Sellers, B. (2015). *Why Philosophize; Why not Just Model?* (Henderson-Sellers 2015)
  Enhancement of the MOF metamodel approach by placing another ontological or definitional layer between the M1 model and the M2 meta-model layer.
- Ludewig, J. (2002). *Modelle im Software Engineering - eine Einführung und Kritik* (Ludewig 2002)
  The key fundamental concepts of modeling are critiqued based on their potential for meaningful use.
- Mahr B. (2015). *Modelle und ihre Befragbarkeit – Grundlagen einer allgemeinen Modelltheorie* (Mahr 2015)
  The generalization, systematization, and formalization of the approaches of H. Stachowiak and of the Berlin 20xy circle for the model-being presented in an axiomatic form.
- Moody, D. (2009). *The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations* (Moody 2009)
  Develops a set of principles (semiotic clarity, perceptual discriminability, semantic transparency, manageable complexity, cognitive integration, expressiveness, dual coding, economy, cognitive fit) for designing cognitively effective visual notations.

And in all immodesty

- Mayr, H.C., Thalheim, B. (2021) *The Triptych of Conceptual Modeling - A framework for a better understanding of conceptual modeling* (Mayr and Thalheim 2021)
  Develops a foundation for conceptual modeling based on three dimensions: model dimension, instrumentation dimension (e. g., language dimension), and meaning or integrateable knowledge dimension.

## 4.3 Where Do We Stand

One should think that with all these intelligent works, hardly any fundamental questions remain open. Especially since there are many more publications on modeling. For example, in the proceedings of the three main modeling conferences ER (Int. Conference on Conceptual Modeling), MODELS (Int. Conference on Model Driven Engineering Languages and Systems) and BPM (Int. Conference on Business Process Management) alone, more than 3,400 scientific papers by more than 4600 authors have been published since their respective existence, not including workshops, all other conferences and journals. So there is obviously a huge body of knowledge. And yet central questions are still open. For example, '*When is a model a conceptual one?*' (although Mayr and Thalheim 2021 make a proposal for this). Given the many modeling languages and tools offered as 'conceptual', such questions may seem quixotic and academic to practitioners. However, a scientist should be concerned if he or she researches and teaches in the field of modeling but cannot give an universally valid or at least broadly accepted answer to this question. We will now try to identify some reasons for these deficits.

### 4.3.1 Model: A Working Definition

Before we delve further into the world of modeling and its stumbling blocks here, we must at least briefly explain what we understand as a '*model*' – because even on this there are many different opinions and definitions. Here we go beyond the definition presented in Mayr and Thalheim (2021) and Mayr and Thalheim (2022) by distinguishing between the terms '*model*' and '*mental model*'. According to Mahr (2021) and theories of Cognitive Science, a mental model is a component of a person's *cognitive structure* (Forstmann and Wagenmakers 2015; Haier 2016) consisting of *ideas* and *judgments* which has at least the following characteristics:

1. Relation to an origin: *'A model is a model of something'.*
2. Concern and usage: e. g., understanding, communicating, agreeing.
3. Purpose and function, e. g., (a) descriptive: analyze, assess, explain, (b) prescriptive: plan, design, (c) explorative: search, predict.
4. Domain: the modeled 'perceived world'.
5. Context: the relevant personal, environmental, social and spatio-temporal circumstances.
6. Focus: the relevant aspects of the origin(s) for the given purpose.

Figure 4.1 shows a very simplified representation of our underlying conceptual framework: we first consider the derivation of the term 'mental model': It consists, as we think, of 'mental objects' that humans form with the help of 'cognitive processe' and integrate into the 'cognitive structure' of their brain (not shown here). According to model characteristic 1, there is an origin, i. e., an object in the perceivable world.

In order to be able to communicate about ideas (i. e., their mental objects) with others, humans designate them with physically recognizable symbols, for example, of textual or graphic kind, sound waves (e. g., spoken or sung words or texts), sequences of light or even smoke signals as they are said to have been used by Indians. We use the umbrella denomination 'description' for these symbols and assume that they basically belong to a

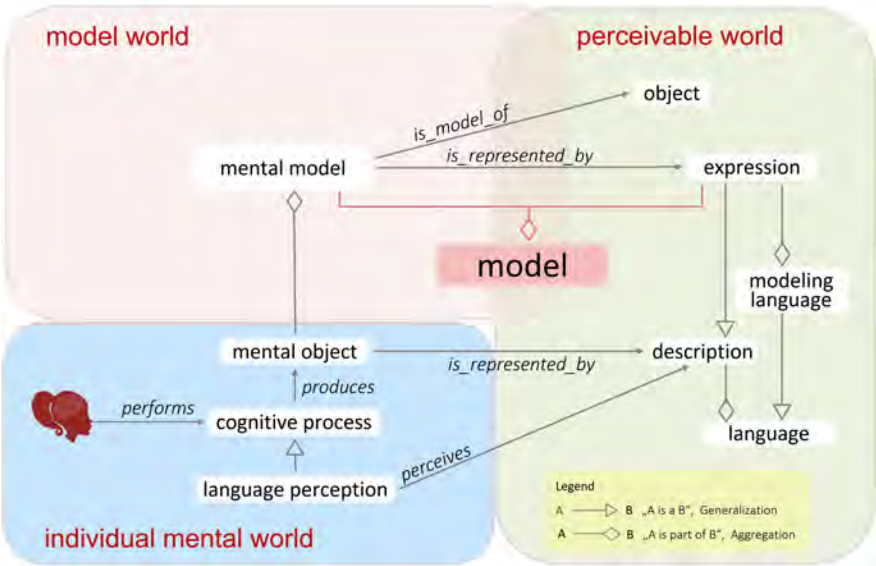



Figure 4.1: About the model concept

kind of (natural, i. e., evolved) language in a very broad sense (see Figure 4.1: descriptions are part of a language and represent mental objects).

In order to infer from a perceived description its meaning (i. e., the represented mental object), humans have the ability of 'language perception' (Tillmann 2012), with which they can understand, for example, the signal of a police siren not only as a sequence of tones but as a warning, i. e., language perception again is a kind of a cognitive process.

This allows us now to sharpen the notion of 'model' itself: We see it as an entity consisting of a mental model and an 'expression' representing it. We use 'expression' here (as a specialization of 'description'), to express that it is a more formal construct belonging to a modeling language. For more details and in particular the modeling and representation rule systems underlying model building we refer (Mayr and Thalheim 2021).

Modeling languages are used to simplify communication about models by enabling the communication partners to develop as similar an idea as possible about a model object. The simplest form are technical languages whose elements are explained in natural language. More sophisticated are formal languages. They consist of a set of modeling concepts, usually defined in terms of a metamodel, and a representation framework with notation and syntactic rules. The BPMN[2] can be mentioned here as an example.

In practice, the word 'model' is often also used for its representation (i. e., the term representing it). Colloquially, this is fine, e. g., when one speaks of a model car, a model train, the prototype of a product, or a 'model' that is supposed to give the viewer an idea of how a fancy dress or suit would look on him. Therefore, we will also practice this homonymic usage in what follows, specifying more precisely only when it might be unclear to which meaning we are referring.

2 https://www.omg.org/spec/BPMN/

### 4.3.2 Modeling, the Computer Science Cinderella

On the basis of the preceding clarification of terms, we can assume that human beings are continuously creating models (they are 'modeling'), since they permanently form ideas about phenomena that they perceive through their senses and form judgments about them. Mostly, however, they does this unconsciously – not only in everyday life but also in professional activities.

Consciously dealing with modeling, i. e., defining modeling languages and using them consistently, on the other hand, is not everyone's cup of tea. In engineering, and especially where life-critical systems are concerned, this is a matter of course. No electrical engineer, for example, would think of building a large electrical system without first designing and analyzing a model, e. g., an electronic circuit diagram. Accordingly, modeling and simulation play an important role in engineering education.

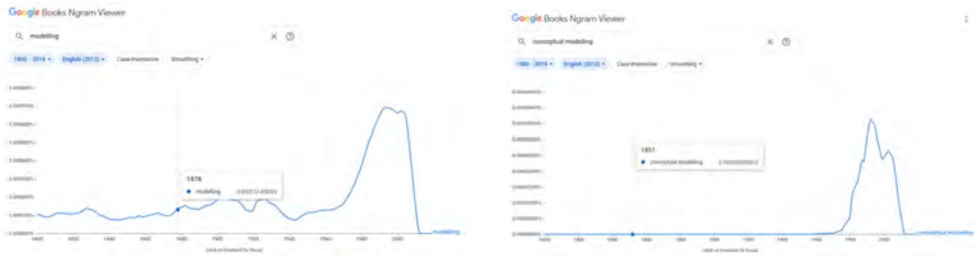
In Computer Science, surprisingly, this applies only to partial areas: Especially in software business practice, one often encounters the opinion that the effort for systematic modeling does not pay off, that the customer does not pay for it, or that the productivity or the effort of modeling cannot be measured 'in lines of code' (Ludewig 2002) like with programs, although other metrics, e. g., the function point method (Garmus and Herron 2001), could easily be adopted. Often also, the object under development is estimated so complex that it cannot be described with a modeling language, but must be programmed right away. This is of course a contradiction in terms. For example, programs are representations of ideas about processes which are to be processed by a computer. They are, therefore, representations of models, in short: *Programming is modeling*! Accordingly, requirements engineering also consists to a large extent of modeling, which is actually recognized in practice. Unfortunately, however, this is only true to the extent that requirements models are usually lost sight of during development: Necessary adjustments are no longer made in the model, so that the original design document is not a valid model of the end product.

Researchers live under the mistaken belief that description, and thus descriptive models, are the central modeling task in programming. Practitioners, however, need models as building templates, i. e., they need prescriptive models. And such models should not only reflect the end result but also show how one came to the model, why one chose this way and with what justification this way is good. Data analysis, for example, works with hidden and sometimes deliberately burrowed assumptions about the approach, the methods, the data and the tools. In contrast, database scientists have had to learn painfully how important metadata with its many facets is.

The two NGrams in Figure 4.2 show that there has been a general downward trend in modeling and conceptual modeling over the last two decades. And this despite the challenges posed by the breadth and diversity of applications. However, that might not be the whole truth, because models are now appearing under different names, for example as 'digital twins'. In the next subsection, we will address some reasons that we believe have led to this shadowy existence of modeling.

### 4.3.3 What are the Problems?

In our opinion, the problems of modeling can be an be illustrated by three observations which we will explain below:



(a) Modeling (−24 per cent since 1992) (b) Conceptual modeling (−43 per cent since 2005)

Figure 4.2: The seemingly waning interest in (conceptual) modeling

- Observation 1: *Entropy* (Ben-Naim 2007) is growing in the domain of modeling, i. e., substantial efforts (‘energy’) are necessary to put things in order.
- Observation 2: The *law of logistic growth* (Verhulst 1838) is valid for modeling methods, i. e., they start with a manageable set of instruments, grow stronger and stronger until the turning point, which heralds the decay.
- Observation 3: Fundamental questions are repeatedly reformulated, twisted and turned by the community, but not conclusively answered.

### 4.3.4 Entropy

Clues to which entropy in the Informatics field of modeling research can be traced are

- an unmanageable variety of methods and tools which are mostly not ‘interoperable’,
- a sparse methodology,
- fuzzy terminology.

There are a few methods that have become a certain standard over time and have endured, e. g., UML, ER modeling, Petri nets or BPMN. Apart from this, however, the review of relevant literature suggests that the pressure for ‘innovation’ at scientific institutions leads to a *continuous stream of new approaches or variants of existing ones* which differ from each other only in nuances but by varying wording and graphical notations. The ‘not invented here’ syndrome seems to prevail, according to which I prefer to invent my own (insufficient) method before helping to improve an insufficient method of someone else. Even the ‘standard methods’ mentioned above are not immune to this problem: new variants have been and are being introduced again and again, but they mostly burn out quickly like fireflies. Even worse, sometimes what is already known is recycled under a new name, or reinvented, because the publications in question date back too far (and are thus possibly considered out-of-date), or are from a related field. A good example of this is the OMG Metaobject Facility MOF[3] which is referenced in practice today in the context of metamodeling. The abstraction levels of MOF, however, correspond (just in inverse numbering) to the layers of the ANSI standard X3.138-1988 for information resource dictionary systems (IRDS) (Parker 1992). This is not indicated in MOF. All this tends to confuse practitioners rather than make them enthusiastically embrace them as the method inventors would like to see. So it is no

3 https://www.omg.org/spec/MOF/

wonder that in business practice, instead of any modeling tool, visualization programs such as Visio™ are used to represent models.

Another reason for the reluctance of practitioners, however, is probably that inventors of new modeling methods are usually happy with presenting the method's conception including the metamodel, the notation (and thus the language), and a corresponding software tool. However, as was recognized very early on, this is not enough to make a method practicable. Rather it requires also instructions, i. e., a *'proceeding model'* (Kaschek and Mayr 1976), how the language is to be used, for example:

- hints which language expressions are to be preferred or rejected (a 'style guide'),
- rules for applying the style appropriate for a particular context, and
- construction rules for the efficient production of expressions of good style.

Modeling language, tool and proceeding model together with a set of goal templates (for deriving the goals of a particular usage) then constitute what we mean by a *'modeling methodology'* (Jaakkola and Thalheim 2011). The provision of such modeling methodologies could form the basis of a *'modeling lore'* or 'modeling theory', which we believe is urgently needed and is taken for granted in other engineering disciplines as 'design theory'.

In our opinion, the lack of a modeling theory is also reflected in the almost Babylonian variety and vagueness of terms: Hard definitions of the form *'A is a ...'* are rarely found, mostly paraphrases are used like *'A is characterized by ...'* or 'A *is comprised of ...*'. A very recent example of this can be found in : *'Conceptual models are comprised of constructs, such as entities, events, goals, attributes, relationships, roles, and processes, connected by well-defined rules'*. Of course, such attempts of explanation have their justification in informal settings, but they lead to fuzzy semantics and thus inconsistent interpretations. To give some samples:

- still, the terms 'conceptional model' and 'conceptual model' are used synonymously,
- the term 'parallelism' is used where 'concurrency' is meant,
- the term 'role' (as an aspect of associations) is used very differently,
- the situation is similar for the term 'attribute',
- for cardinalities and multiplicities there are many variants and different notations,
- foreign words are used incorrectly and blurred, such as 'semiformal' or 'semi structured'. One talks for example of an 'semiformal language' if it is about an informal one, which may have a few construction rules. These seem to be magic words of Computer Science (you hardly find them in Mathematics or technical sciences) if something cannot be grasped exactly.

### 4.3.5 Logistic Growth

In the middle of the seventies, Teichroew and Hershey (1976) introduced the *'Program Statement Language'* and the *'Program Statement Analyzer'* (PSL/PSA) within the framework of the 'ISDOS' project (Information System Design and Optimization System). The focus was on requirements modeling and requirements analysis. At the beginning, the language was lean and transparent, but in the course of time new concepts were added, so that later 19 different modeling concepts for 'objects' and no less than 102 different relation types were distinguished. Originally, this method was quite used, especially in practice, but with the growth, transparency, maintenance, and availability problems seem to have arisen, so that today there is still a reference website [4], but the status is unclear.

The *'Structured Analysis and Design Technique', SADT* (Ross and Schoman 1977) developed by D.T. Ross and his company SofTech at the end of the sixties had a similar history. Initially, it came with only a few elements and a transparent graphical and hierarchically structured notation, but then was continuously extended. In 1981, the language IDEF0 based on it was published as part of the IDEFxx family of modeling languages in the field of Software Engineering, and in 1983 it was registered by the National Institute of Standards and Technology as a Federal Information Processing Standard (FIPS). In 2008, this standard was withdrawn. SADT was also the subject of academic teaching for a long time, but today it is hardly known. Related and caught by a similar fate is also the method *'Structured Analysis and Systems Specification'* (DeMarco 1978) presented by Tom de Marco at the end of the 1970s for which later IDEF could also be used.

A more recent example is provided by the *'Unified Modeling Language UML'* [5]. Originating from the merging and unification of various predecessors (methods of 'Object Oriented Analysis'), in 1996 it provided 5 different model views, which were represented by so-called *diagrams* (Use Case, Class, Sequence, Collaboration, State). In 2012 (UML 2.5), there were already 18 official and 7 unofficial diagrams. It is therefore no wonder that in corporate practice few people seem to have a comprehensive and deep understanding of the overall concept and, according to our experience at least with small and medium-sized software manufacturers, there is no consistent use. So the hype has softened and it remains to be seen when the next, simpler and more transparent method will be invented.

Incidentally, this up and down of methods could also be described quite well with the so-called Kondratjev cycles (Metz 2006), i. e., with the theory of long waves: The starting point for such long waves are paradigm shifts and the associated massive investments in innovation. Once an innovation has become generally accepted, investments decrease and a downturn occurs. During the downturn, however, work is already underway on a new paradigm.

### 4.3.6 Fundamental Questions

In this subsection, we give a few quotations from former times by which one can recognize that fundamental questions are not completely answered until today. The preface to the proceedings of the 1979 founding event of today's SIG EMISA in the Gesellschaft für Informatik GI states: *'The constantly growing complexity of operational and socio-technical information systems today touches the limits of our planning capacity. A formally sound set of tools for the planning, design and operation of such systems is therefore absolutely necessary. For example, methods for determining the requirements to be placed on an information system are needed, as are formal modeling concepts for its description, specification, and analysis, and finally techniques for evaluating its behavior'.* Note that the term 'formal' is used here, not 'semi-formal'.

In a keynote address to the first German Modeling Conference 'Modellierung' held in Münster in 1998, it was stated, among other things:

4 https://requirementsanalytics.com/index.php/consulting-and-training/pslpsa
5 https://www.omg.org/spec/UML/2.5.1

- *'Quantitative aspects associated with performance requirements or security aspects are still hardly covered by conceptual terms. For the developer, however, they are of great importance, a design document should therefore show them'.* and
- *'Another problem still lies in the abstraction distance between natural language requirements specifications and conceptual designs. Although almost every inventor of a conceptual modeling method (...) has claimed that his approach provides a suitable basis for understanding between system analysts/developers. However, practice shows that conceptual models are often often intransparent for the users and thus cannot be sufficiently validated by them'.*
- and finally: *'In conclusion, it can be stated that the field of operational information systems modeling is developing slowly but steadily. The basic framework of generally accepted concepts and terminology is growing, but we are still far away from a design methodology or even design theory'.*

In the foreword to EMISA Forum 1/98, Helmut Thoma writes: *'Modeling is still treated stepmotherly (in practice); This after decades of desperate struggle to put thinking and constituting before realizing'.* If one now goes through the list of topics on which papers are solicited in the Call for Papers of the ER Conferences of the last 10 years, one finds among others regularly (partly in different wording):

- Foundations and Concept Formalization
- Quality and Metrics of Conceptual Models
- Data Semantics and Integrity Constraints
- Interleaving Modeling and Development

The CFP for the upcoming 2023 ER also reads: *'Submissions that lead to new foundations, links, applications, or enlarge current boundaries of conceptual modeling are especially welcome'.* The MODELS conference 2023 is also looking for contributions to, among other things, *'Fundamentals of model-based engineering, including the definition of syntax and semantics of modeling languages and model transformation languages'* or *'Development of model-based systems engineering approaches and modeling-in-the-large, including interdisciplinary engineering and coordination'.*

To sum up: the set of instruments claimed since the 70ies and its foundation still exists only in rudimentary form. Consolidation has not yet taken place, so that academic teaching and practical training are also rather superficial: the languages that are in fashion at the moment are taught, but the theoretical foundations are rarely studied in depth. Moreover, the modeling languages do not keep pace with the technological development either. For example, new paradigms would have to be found and new techniques developed for Big Data and Data Analytics. Moreover, an automatic translation of models into code exists only in rudiments. Consequently, there is a mountain of unfinished homework from theory and practice of (conceptual) modeling, i. e., we face a lot of challenges that we need to address if modeling is not to continue to lose its importance.

## 4.4 Opportunities for Progress

The weaknesses pointed out in the previous chapter seem to have shaken up the community a bit. In any case, recently there have been an increasing number of contributions dealing with the state of the discipline, (e. g., Castellanos et al. 2018; Lima et al. 2020; Recker et al. 2020; Storey et al. 2023), foundations (e. g., Yu 2009; Guizzardi et al. 2015; Guarino et al. 2019; Mayr and Thalheim 2021), focal points and corresponding communities (e. g., Sandkuhl et al. 2018; Khatri and Samuel 2019; Lukyanenko et al. 2019; Eriksson et al. 2019), or even trying to give starting points for the characterization of research in conceptual modeling (Delcambre et al. 2021). So, we seem to be heading where the engineers already are: They have learned to develop and use blueprints, schematics, etc., which gives them great advantages regarding quality and productivity. Therefore, it is impossible to imagine engineering without models: the Cinderella is a beauty there - and should become so now in Computer Science. Accomplishments that suggest this but have yet to be worked through in disciplinary terms include the following:

- Even if the term ‘model’ is not the focus, requirements engineering is actually modeling. ‘Brownfield’ development and system evolution cannot exist without models. Web information systems design and development is model-backed. User-centered development of systems must also integrate models of users and usage.
- Large and complicated systems are created with action instructions, i. e., activity or ‘proceeding’ models, respectivels, possibly even with a waterfall architecture. Many methodologies have emerged that await generalization but have already resulted in low-maintenance systems.
- Process engineering, process elicitation, and process elaboration led to powerful tools and an improvement of business support.
- Systematic and thoughtful programming is always accompanied by and supported by models. Object-oriented and object-relational approaches demonstrate the silver bullet of stepwise specification for programming in the large.
- Despite the huge variety of modeling languages, systematic domain-specific design of languages is becoming acknowledged and more widely used.
- Generic, pattern-based, and reference solutions are the basis for macro- and meso-models as a reliable starting point instead of ‘greenfield’ development from scratch.
- Database and information systems development and design can not be imagined without models and modeling. User viewpoints are essentially derived models.
- Models are going to be used as prescription of coding. Models can be translated to code fragments as kernel elements of software.
- Quality assessment and improvement of models is becoming a common technique.
- ‘Semantification’ and conceptualisation results in models that are meaningful to all stakeholders. Ontology engineering is becoming a technique for semantic foundation and sense-making elaboration.
- Due the the large body of knowledge on models, modeling and model usage, claims are raised that modeling is becoming a sub-discipline and modeling is essentially the fourth dimension of Computer Science.

This marks the path from modeling as a luxury to a daily, deliberate, and purposeful activity. It is rumored from IBM’s main office that there was a saying on the wall behind the CEO, ‘Think, think, think, before doing anything’ but then it was obscured by a cabinet so that only the first half was seen. Together it adds up, however: we are not rich enough to invent everything over and over again, sometimes even worse.

## 4.5 The Triptych Foundation of Conceptual Models

At the end of Section 4.2, we noted our contribution (Mayr and Thalheim 2021) to the foundations of conceptual modeling, which we had laid out as *'as a contribution to the "anatomy" of conceptual models'* and with which we intended to help better understand the nature of conceptual modeling. In particular, with the explanatory paradigm of the triptych we especially want to give a starting point for conceiving conceptual models as a special kind of models and to have a clear distinguishing criterion, i. e., we want to be able to say whether a given model is a conceptual one or not. This would solve one of the major deficits (see introduction to Section 4.3) in our field.

The basis of our considerations is that conceptual modeling consists of three parts: (1) a central model part, (2) the instrumentation by a language, and (3) the inherited and selected world semantics within the given application. The triptych paradigm illustrates these three parts through its three dimensions (wings):

- The *linguistic dimension* of language representation, where 'language' here is quite broad and can include textual, graphical, audio-visual, biosemiotic, and other physical forms of representation. In the case of well-known modeling languages such as UML, ER, or BPMN, it is the respective notation together with the grammatical composition rules. Languages enable and hinder, have to match with necessities in applications, have to be evolve together with technology, and have to reflect the accepted culture in the application domain.
- The *model dimension* defines the structure of the models at different levels of abstraction, forming a hierarchy via stepwise intension/extension relationships: Each level provides modeling concepts (elements, relationships, etc.) that can be used at the next level for model building. Usually four levels are considered (Metametamodel, Metamodel, Model, Individuals), but in general there can be any number of levels, e. g., if on the individuals level again concepts are offered, to which there are extensions[6]. Consider, as an example, the original entity-relationship model: As a metamodel it includes modeling concepts like 'Entity Set', 'Relationship Set', 'Attribute', 'Value Set', 'Role'. For a concrete application, extensions can be derived, e. g., an entity set 'Customer' to which certain attributes such as have extensions in the form of individuals (models of concrete objects of a considered mini-world), for example the 'customer' *Smith.*

  By using the metamodel as a 'controlled vocabulary', i. e., by using only words from this pool, and by using natural language identifiers at the various levels, a so-called 'a priori semantics' is created, i. e., the models and their extensions are interpretable on the basis of the meanings of the natural language designators used, but not necessarily unambiguous. Keep in mind that these designators correspond to the 'expressions' in the 'perceivable world' (see Figure 4.1), with which together mental models become models. It should be obvious that the linguistic dimension is part of the perceivable world.
- The *semantics dimension* provides an unambiguous meaning of the elements of the model dimension (at all levels and including the associated expressions from the linguistic dimension) through an ontological or encyclopedic grounding of each single

6 Note that unlike the MOF and IRDS metadata frameworks, we are not talking here about the class/instance relationship borrowed from object orientation, but rather about semantic intension/extension relationships.

element. With this we can define: *A conceptual model is a model that has a complete grounding in the semantic definition, i. e., if all its constituents are defined there.*

If we restrict ourselves to the model and linguistic dimension, we find ourselves in the traditional world of modeling, which provides the framework for many useful activities: With the addition of the semantic dimension, however, we create the basis for a semantically unambiguous form of modeling, as it is needed in Computer Science and especially for the specification and realization of complex systems.

The semantics dimension was ignored for a long time because world knowledge was taken for granted and it was not realized that each application brings in its specific background knowledge. Without knowledge of world knowledge, however, interoperability and integration of systems remains a dream.

In Figure 4.3, we use the example of business balance sheets to illustrate the dimensions and the model and language hierarchies to be considered in conceptual modeling. In the middle column (the *'model wing'* of the triptych), we see four model levels $H^n$ - $H^{n+3}$ as described above. The metamodel at $H^{n+2}$ provides the desired concepts from accounting theory; at the model level $H^{n+1}$, extensions of this metamodel are derived as balance sheet schemas, whose extensions at the individuals level $H^n$ are concrete balance sheets for a specific key date.

If we now look at the right column (the *'language wing'*), we see a somewhat different hierarchy, namely that of the language definitions (where we mean by 'grammar' a complete syntactic definition of a language, i. e., including terminal symbols/literals and all composition rules for valid words and expressions. Thus, (formal) languages for the definition of metamodel, model and data/object expressions are to be provided.

Picture (1) in this wing shows the (language of) terminology introduced by the accounting theory of business economics: it represents the accounting metamodel. Picture (2) shows the representation of a model derived from the metamodel, namely the balance sheet scheme of the company MaTha Limited, which is the basis for the preparation of the company's balance sheet at certain reporting dates: The grammar here apparently provides for a tabular

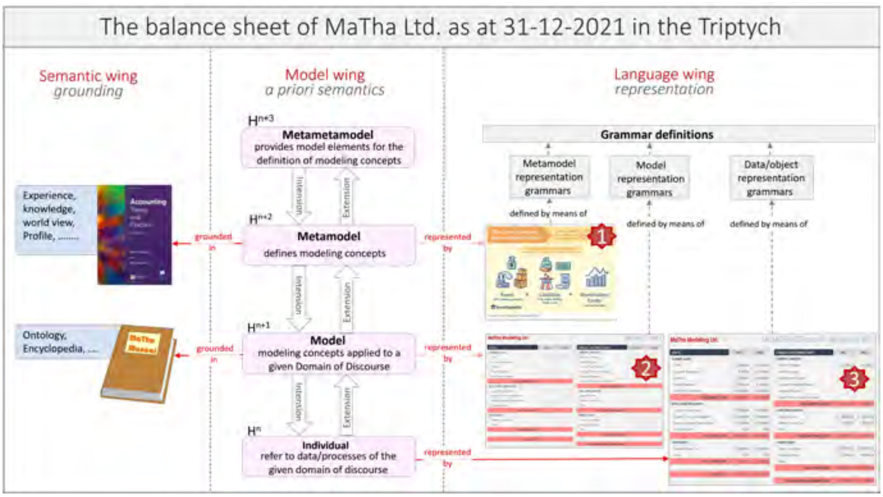



Figure 4.3: Illustration of the triptych paradigm

representation with the columns 'Assets' and 'Liabilities', as well as further horizontal subdivisions. An extension of this model is, for example, the 'Balance sheet of the company MaTha as of 31.12.2021' as shown in picture (3): Here it is no longer a kind of scheme, but a table occupied with concrete values. Quite obviously, then, we can say that this table is an expression representing a mental model of our object of observation MaTha Limited, so together they form the model 'Balance sheet of the company MaTha as of 31.12.2021'. MaTha is therefore the object/origin of our model. Of course, it is also possible to imagine completely different models of a company, for example in terms of their organizational and process structure – you just have to configure the metamodel and the associated languages accordingly. Everything we have said in this example up to this point corresponds to the standard approach to the definition of models and technical languages in a wide variety of application disciplines. It is important for us to note that we have not talked about 'conceptual model' so far, nor did we need to.

This only happens if we look at the left column, the *'semantic wing'*: There we find, among other things, codified knowledge about accounting (for example, in an encyclopedic work on accounting) and an organizational manual of MaTha Limited, in which every term occurring in this company is clearly defined, for example, in the form of a corporate ontology. If every element used in the metamodel and in the MaTha accounting model is grounded in this semantic dimension, then we speak of a *conceptual model* – both at the level of the metamodel and at the level of the model.

Two things should be mentioned here: First, it becomes clear that the same model can easily be applied in different settings: Let's think of a tax advisor who, in addition to MaTha Limited, also looks after the company StoPa Inc. and uses the same accounting scheme for both: all he has to do is to change the literal of the company name on the language wing – and to replace the company manual of MaTha Limited with that of StoPa Inc. on the semantic wing. So the 'reference model' in the middle remains the same, but the conceptual semantics coorespond to the new company. Second, even if we use the terms metamodel, model etc. in the singular, this does not mean that on each level of abstraction only one model can exist. On the contrary, usually – with the exception of the top level which should ensure a certain comparability of the underlying extensions – many models are allowed on arbitrary levels; think, e. g., of different DDL relational schema definitions for a multi-database application. But of course this is also true for the metamodel level, for instance, if one wants to consider structural, process and functional models in a modeling compound and specify their interdependencies within that compound.

In other words, *'multimodeling'*, *'multilevel modeling'*, and *'multiperspective modeling'*, as advocated by Ulrich Frank and the Duisburg school (Frank 2014), are intrinsic to the model dimension. Multilevel modeling attempts to overcome the classic MOF architecture with class-instance layering between the various levels. Multiperspective modeling supports the variety of views within a community of practice. It turns the global-as-design approach from head to toe with explicit orientation on language and modeling support for each of the community members. Ulrich's MEMO approach integrates these and uses a model suite (Thalheim 2010) with overview, goal system, process, organisational structure, process control flow, IT infrastructure, decision support, performance indicator, and business process models.

To sum up: The Triptych paradigm is an explicit foundation of conceptual modeling. Terms imputed from linguistics and notions imported from semantics are the background of conceptual models.

## 4.6 Conclusio: Some Hints and Suggestions for the Future

As we have already indicated in Section 4.4, we assume that the importance of modeling will not decrease but rather increase if we do our homework in our community. The importance can be characterized with a Kondratieff wave sequence: A single wave includes phase states such as underappreciated and neglected, overappreciated and renamed, disappointed and devastating, enlightening practical, and uninteresting. Therefore, it is important that the knowledge and experience pool still is processed and systematized. Such a Body of Knowledge (BoK) is processed and systematized towards a comprehensive body of knowledge. Then a science and culture of models, model design and use, and modeling will continue to be an independent and at the same time fundamental sub-discipline of modern Computer Science (Cabot and Vallecillo 2022; Thalheim 2022a; Thalheim 2022b).

To be able to achieve this goal, modeling will have to provide convincing answers to questions like the following in the near future:

- Can we really avoid complexity through use of models?
  *One of the main benefits of models is complexity reduction and focusing. This requires a mastery of the game of adequacy and concentration on the essential.*
- What means capability-oriented application and usage of models?
  *Use models whenever you can handle them in a proper way. They are not luxury goods, but goods that should facilitate life, activities, and practice purposefully.*
- Are models supporting humanised thinking and acting instead being driven be the artificial?
  *Technical and artificial worlds are far too complex and upsetting for humans. Models allow an understanding design and meaningful usage in the simplest possible and valuable way.*
- Can models be used for generation, e. g., as powerful companion for all phases of programming, maintenance, and system realisation?
  *Modeling as higher-order programming will be the real future of modeling in Computer Science. Models will become core elements of software and hardware after development of appropriate theories and sophisticated compilers.*
- Do we have means for quality-aware modeling?
  *Models have not to be true, but of good quality according to the form of use, i. e., right quality on the right place; but no more than that.*
- What means to enhance models by interpretability?
  *Models have to be intuitive for their Community of Practise. They are man-made and man-oriented instruments in all human endeavours.*
- What benefit we could gain by model apprenticeship and eruditeness (Konstruktionslehre – modelology)?
  *One should enjoy the benefits of each advancement. Let us try to develop a culture of models and modeling.*
- Can we construct models for any scenario and function?
  *No! But this is not necessary. We should focus on the core tasks of our discipline.*
- Can we rely on a model?
  *At least as long as it is a good model. As long as the model is designed for its proper use.*

- Are models gifting us with conceptualisations?
  *Hopefully, but this requires work. Conceptualisation and 'semantification' is, however, a nice task but not a must.*
- What means skilled apprenticeship and mastery?
  *Pure lore and teachings can be useful. But the real effect comes with use. Apprenticeship and mastery will become the killer modeling usage.*
- Can we detect and avoid misuse, misapplication, and manipulation through models?
  *Models are embedded in their own landscape and not arbitrarily transplanted or even misused and misapplied, To avoid this requires efforts and should be done with care on necessity.*

## References

Batini, C., Ceri, S. and Navathe, S. B. (1992). *Conceptual Database Design: An Entity-Relationship Approach.* Benjamin/Cummings.

Becker, J., Wolfgang, P. and Vering, O. (2012). *Grundsätze ordnungsmäßiger Modellierung: Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement.* Berlin, Heidelberg: Springer.

Ben-Naim, A. (2007). *Entropy Demystified.* World Scientific.

Brambilla, M., Cabot, J. and Wimmer, M. (2017). *Model-Driven Software Engineering in Practice, Second Edition.* Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers. DOI: 10.2200/S00751ED2V01Y201701SWE004.

Broy, M. and Stølen, K. (2001). *Specification and Development of Interactive Systems – Focus on Streams, Interfaces, and Refinement.* Monographs in Computer Science. Springer. ISBN: 978-1-4612-6518-4. DOI: 10.1007/978-1-4613-0091-5. URL: https://doi.org/10.1007/978-1-4613-0091-5.

Cabot, J. and Vallecillo, A. (2022). 'Modeling should be an independent scientific discipline'. In: *SoSyM Software and Systems Modeling* 21, pp. 2101–2107.

Carmona, J., Dongen, B. van, Solti, A. and Weidlich, M. (2018). *Conformance checking.* Springer.

Castellanos, A., Samuel, B., Recker, J., Jabbari, M. and Lukyanenko, R. (2018). 'Conceptual modeling research: revisiting and updating Wand and Weber's 2002 Research Agenda'. In: *AIS SIGSAND*, pp. 1–12.

Chen, P. P.-S. (1976). 'The entity-relationship model—toward a unified view of data'. In: *ACM transactions on database systems (TODS)* 1.1, pp. 9–36.

Codd, E. F. (2002). 'A relational model of data for large shared data banks'. In: *Software pioneers.* Springer, pp. 263–294.

Delcambre, L. M., Liddle, S. W., Pastor, O. and Storey, V. C. (2021). 'Articulating Conceptual Modeling Research Contributions'. In: *Advances in Conceptual Modeling.* Springer International Publishing, pp. 45–60.

DeMarco, T. (1978). *Structured Analysis and System Specification.* Yourdon Press.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A. et al. (2013). *Fundamentals of business process management.* Vol. 1. Springer.

Elmasri, R. and Navathe, S. B. (2000). *Fundamentals of Database Systems, 3rd Edition.* Addison-Wesley-Longman.

Embley, D. W. and Thalheim, B. (2012). *Handbook of conceptual modeling: theory, practice, and research challenges.* Springer.

Eriksson, O., Johannesson, P. and Bergholtz, M. (2019). 'The case for classes and instances - a response to representing instances: the case for reengineering conceptual modelling grammars'. In: *European Journal of Information Systems* 28.6, pp. 681–693. doi: 10.1080/0960085X.2019.1673672.

Ferstl, O. K. and Sinz, E. J. (2015). *Grundlagen der Wirtschaftsinformatik.* Oldenbourg Wissenschaftsverlag.

Forstmann, B. U. and Wagenmakers, E.-J., eds. (2015). *An introduction to model-based cognitive neuroscience.* Vol. 614. Springer.

Frank, U. (2014). 'Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges'. In: *Softw. Syst. Model.* 13.3, pp. 941–962.

Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction.* springer open.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley.

Garmus, D. and Herron, D. (2001). *Function point analysis: measurement practices for successful software projects.* Addison-Wesley Longman Publishing Co., Inc.

Guarino, N. (1994). 'The ontological level'. In: *Philosophy and the cognitive sciences.* Holder-Pichler-Tempsky, pp. 443–456.

Guarino, N., Guizzardi, G. and Mylopoulos, J. (2019). 'On the Philosophical Foundations of Conceptual Models'. In: *Information Modelling and Knowledge Bases, IOS Press* XXXI, pp. 1–15.

Guizzardi, G., Wagner, G., Almeida, J. P. A. and Guizzardi, R. (2015). 'Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story'. In: *Appl. Ontology* 10, pp. 259–271. doi: 10.25300/MISQ/2021/16027.

Haier, R. J. (2016). *The neuroscience of intelligence.* Cambridge University Press.

Halpin, T. and Morgan, T. (2010). *Information modeling and relational databases.* Morgan Kaufmann.

Henderson-Sellers, B. (2015). 'Why Philosophize; Why not Just Model?' In: *LNCS, Springer* 9381, pp. 3–17. issn: 1619-1366. doi: 10.1007/978-3-319-25264-3_1.

Jaakkola, H. and Thalheim, B. (2011). 'Architecture-driven modelling methodologies'. In: *Information Modelling and Knowledge Bases.* Vol. XXII. IOS Press, pp. 97–116.

Karagiannis, D., Mayr, H. C. and Mylopoulos, J., eds. (2016). *Domain-Specific Conceptual Modeling, Concepts, Methods and Tools.* Springer. isbn: 978-3-319-39416-9. doi: 10.1007/978-3-319-39417-6. url: https://doi.org/10.1007/978-3-319-39417-6.

Kaschek, R. and Mayr, H. C. (1976). 'A characterization of OOA tools'. In: *Proc. 4th International Symposium on Assessment of Software Tools.* IEEE, pp. 59–67. isbn: 0-8186-7390-7. doi: 10.1109/AST.1996.506478.

Kent, W., ed. (1978). *Data and Reality.* North-Holland.

Khatri, V. and Samuel, B. M. (2019). 'Analytics for managerial work'. In: *Communications of the ACM* 62(4), p. 100. doi: 10.1145/3274277.

Lima, H. C., Laender, A. H., Moro, M. M. and Oliveira, J. P. de (2020). 'An analysis of the collaboration network of the international conference on conceptual modeling at the age of 40'. In: *Data & Knowledge Engineering* 130. doi: 10.1016/j.datak.2020.101866.

Ludewig, J. (2002). 'Modelle im Software Engineering - eine Einführung und Kritik'. In: *Lecture Notes in Informatics, GI* P-12, pp. 7–22.

Lukyanenko, R., Castellanos, A., Parsons, J., Tremblay, M. C. and Storey, V. C. (2019). 'Using conceptual modeling research to support machine learning'. In: *Information Systems Engineering in Responsible Information Systems (CAiSE Forum 2019)*, pp. 70–81. DOI: 10.1007/978-3-030-21297-1_15.

Mahr, B. (2015). 'Modelle und ihre Befragbarkeit – Grundlagen einer allgemeinen Modelltheorie'. In: *Erwägen-Wissen-Ethik (EWE)* Vol. 26, Issue 3, pp. 329–342.

Mahr, B. (2021). *Schriften zur Modellforschung*. Herausgegeben von K. Robering. Brill Mentis, Michaelisbund.

Mayr, H. C. and Thalheim, B. (2021). 'The Triptych of Conceptual Modeling – A Framework for a Better Understanding of Conceptual Modeling'. In: *Softw. Syst. Model.* 20.1, pp. 7–24.

Mayr, H. C. and Thalheim, B. (2022). 'The Anatomy of Conceptual Models'. In: *Advanced Information Systems Engineering - 34th Int. Conf., CAiSE 2022.* Ed. by X. Franch, G. Poels, F. Gailly and M. Snoeck. Vol. 13205. Lecture Notes in Computer Science, pp. 551–553. DOI: doi.org/10.1007/978-3-031-07472-1.

Metz, R. (2006). 'Empirical Evidence and Causation of Kontratieff Cycles'. In: *Kondratieff Waves, Warfare and Worls Security.* IOS Press, pp. 91–99.

Moody, D. (2009). 'The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations'. In: *IEEE Trans. on Software Engineering* 35,6, pp. 756–779.

Olivé, A. (2007). *Conceptual modeling of information systems.* Springer Science & Business Media.

Parker, B. (1992). 'Introducing ANSI-X3.138-1988: a standard for information resource dictionary system (IRDS)'. In: *[1992] Proceedings of the Second Symposium on Assessment of Quality Software Development Tools*, pp. 90–99. DOI: 10.1109/AQSDT.1992.205841.

Recker, J., Lukyanenko, R., Jabbari, M., Samuel, B. M. and Castellano, A. (2020). 'From representation to mediation: a new agenda for conceptual modeling research in a digital world'. In: *MIS Quarterly* 45(1), pp. 269–300. DOI: 10.25300/MISQ/2021/16027.

Ross, D. T. and Schoman, K. (1977). 'Structured Analysis for Requirements Definition'. In: *IEEE Transactions on Software Engineering* 1, pp. 6–15. DOI: 10.1109/TSE.1977.229899.

Sandkuhl, K., Fill, H.-G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö. and Winter, R. (2018). 'Analytics for managerial work'. In: *Business & Information Systems Engineering* 60, pp. 69–80. DOI: 10.1007/s12599-017-0516-y.

Stachowiak, H. (1973). *Allgemeine Modelltheorie.* Springer.

Storey, V. C., Lukyanenko, R. and Castellanos, A. (2023). 'Conceptual Modeling: Topics, Themes, and Technology Trends'. In: *ACM Computing Surveys* 55.

Teichroew, D. and Hershey, E. A. (1976). 'PSL/PSA a computer-aided technique for structured documentation and analysis of information processing systems'. In: *ICSE'76: Proceedings of the 2nd international conference on Software engineering.* ACM.

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

Thalheim, B. (2010). 'Model Suites For Multi-Layered Database Modelling'. In: *Information Modelling and Knowledge Bases XXI.* Vol. 206. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 116–134.

Thalheim, B. (2022a). 'Auf dem Wege zur Modellkunde'. In: *Modellierung 2022.* Ed. by M. Riebisch and M. Tropmann-Frick. Vol. P-324. LNI. Gesellschaft für Informatik e.V., pp. 11–32.

Thalheim, B. (2022b). 'Models: The Fourth Dimension of Computer Science – Towards Studies of Models and Modelling'. In: *Software & Systems Modeling* 21, pp. 9–18.

Tillmann, B. (2012). 'Music and Language Perception: Expectations, Structural Integration, and Cognitive Sequencing'. In: *Topics in Cognitive Science* 4.4, pp. 568–584. DOI: 10.1111/j.1756-8765.2012.01209.x.

Van Der Aalst, W. (2016). *Process mining: data science in action*. Vol. 2. Springer.

Verhulst, P.-F. (1838). 'Notice sur la loi que la population suit dans son accroissement'. In: *Correspondance Mathématique et Physique* 10, pp. 3–21.

Weske, M. et al. (2007). 'Concepts, Languages, Architectures'. In: *Business Process Management*.

Yu, E. (2009). 'Social Modelling and i*. Conceptual Modeling: Foundations and Applications'. In: *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, pp. 99–121. DOI: 10.1007/978-3-642-02463-4_7.