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

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

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

Content of This Third Collection

   See also [DT18].

   See also [KT17a,JHWD +16].

28. B. Thalheim. Qualitative and quantitative models [Tha18c] ..................................................... 26
   See also [KT18a,Tha18b,Tha17a].

   See also [JTH +18] and our papers on culture.

   See also [KT18a,Tha18b,Tha17a].

   See also [KT18b].

Remark 1: The collection is work on progress and will be extended after additional papers became accepted in 2019.
Remark 2: See our previous work on the *entity-relationship approach to modelling*, e.g. [ET11, Tha00, MST09b, ST15], on semantics [ST13, Tha11b], on programming, on the theory of databases and information systems and on the technology of information systems at
http://dblp.uni-trier.de/pers/hd/t/Thalheim:Bernhard
or https://www.researchgate.net/profile/Bernhard_Thalheim/publications or
https://scholar.google.com/citations?user=lkH3h9gAAAAJ or
http://independent.academia.edu/BernhardThalheim or ...
### Content of The Second Collection

available through research gate and academia

### Table of Content of the Second Collection

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

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compare with [DT11, TN15, Tha11a, Tha12b].</td>
</tr>
<tr>
<td></td>
<td>See also previous work on model suites, model composition [MST09a, ST10, Tha10a].</td>
</tr>
<tr>
<td>15.</td>
<td>B. Thalheim and M. Tropmann-Frick. Models and their capability [TTF16b]</td>
</tr>
<tr>
<td></td>
<td>Compare with the BPMN foundation work in the BPMN collection, especially with [BST09, KST09, TT13],</td>
</tr>
<tr>
<td></td>
<td>[TTFZ14,TFTL +14].</td>
</tr>
<tr>
<td></td>
<td>Compare with the foundational book [Tha00, Tha09c, Tha09a, Tha09b] and my previous work on entity-relationship approach.</td>
</tr>
<tr>
<td></td>
<td>Theories in business and information systems engineering [BFA +16]</td>
</tr>
<tr>
<td></td>
<td>Compare with our research on semantics and logical foundation.</td>
</tr>
<tr>
<td>18.</td>
<td>B. Thalheim and A. Dahanayake. A conceptual model for services [TD15]</td>
</tr>
<tr>
<td></td>
<td>See also [DT13a, DT10b, MSTW09a, DT12, ADT12a, ADT12b, DT15, TD16], [MSTQ09, MSTW09b, MSTW11],</td>
</tr>
<tr>
<td></td>
<td>[MSTW12].</td>
</tr>
<tr>
<td>19.</td>
<td>B. Thalheim and M. Tropmann-Frick. Wherefore models are used and accepted? The model functions as a quality instrument in utilisation scenarios [TTF16c]</td>
</tr>
<tr>
<td></td>
<td>This paper continues research quality [JT10], architecture [JT11, NT14, TELT14], and privacy [AFT09].</td>
</tr>
<tr>
<td>20.</td>
<td>B. Thalheim. Model-based engineering for database system development [Tha17b]</td>
</tr>
<tr>
<td></td>
<td>See our approaches to development of methodologies for modelling, especially database modelling and the codesign framework in the 90ies and 00s.</td>
</tr>
<tr>
<td>21.</td>
<td>B. Thalheim. General and specific model notions [Tha17a]</td>
</tr>
<tr>
<td>22.</td>
<td>B. Thalheim. Normal models and their modelling matrix [Tha18b]</td>
</tr>
<tr>
<td>23.</td>
<td>Y. Kropp and B. Thalheim. Data mining design and systematic modelling [KT17b]</td>
</tr>
<tr>
<td></td>
<td>See the previous work on data mining design [BT12, JP10, ST12].</td>
</tr>
<tr>
<td>24.</td>
<td>A. Dahanayake and B. Thalheim. The rigor cycle of conceptual modelling [DT18]</td>
</tr>
<tr>
<td></td>
<td>See also our contributions to design science foundations of conceptual modelling [DT10b, DT10a, DT11, DT13b].</td>
</tr>
</tbody>
</table>


The First Collection

available through research gate and academia

Table of Content of the First Collection

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

   Preliminary version: Lecture Notes in Computer Science 5833, [Tha09d] [.89ex]


4. The Science and Art of Conceptual Modelling. TLDKS VI, LNCS 7600, [Tha12c] .................................................. 94-122
   Preliminary version: Lecture Notes in Computer Science 6860, [Tha11c]


6. The Definition of the (Conceptual) Model. EJC’13, [Tha13a, Tha14] .......................................................... 135-148

7. Das Modell des Modelles. EWE’15 [Tha15] .......................................................... 149-151

8. Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung.

   The main book on the Kiel approach to models: [TN15].


10. A Conceptual Model for Services. CMS’15@ER’15 [TD15] .......................................................... 164-174


References


B. Thalheim and A. Dahanayake. Comprehending a service by informative models.


B. Thalheim. The theory of conceptual models, the theory of conceptual modelling and foundations of conceptual modelling.


Abstract

Models, modeling languages, modeling frameworks and their background have dominated research on information systems engineering for last four decades. Models are mainly used as mediators between the application world and the implementation or system world. Modelling is still conducted as the work of an artisan and workmanship. While a general notion of the model and of the conceptual model has already been developed, the modelling process is not investigated so well.

Modelling has to be based on principles and a general theory of modelling activities. One of the lacunas is still a proper understanding of adequacy of models, adequacy of modelling and deployment methods, and a theory of adequacy. We will concentrate on the first issue.

Keywords: model notion; model adequacy; analogy; focus/truncation/abstraction; purposeful; well-formed model; model dependability

1 Models, Modelling Activities, Systematic Modelling

Models are principle and central instruments in mathematics, data analysis, modern computer engineering (CE), in teaching any kind of computer technology, and also modern computer science (CS). They are built, applied, revised and manufactured in many CE&CS sub-disciplines in a large variety of application cases with different purposes and context for different communities of practice. CE&CS expressively use the conception of model for daily work. Modelling is one of their four central paradigms beside structures (in the small and large), evolution or transformation (in the small and large), and collaboration (based on communication, cooperation, and coordination). It is now well understood that models are something different from theories. They are often intuitive, visualisable, and ideally capture the essence of an understanding within some community of practice and some context. At the same time, they are limited in scope, context and the applicability. Models have been considered to be somewhere in the middle between the perception and understanding of the state of affairs (world, situations, data etc.) and theories (concepts and conceptions, statements, beliefs, etc.) since they may describe certain aspects of a situation and may represent parts of a theory. Models should thus be considered to be the third dimension of science [2, 50, 52]. Other disciplines (see for instance [50]) have developed a different understanding of the notion of model, of the function of models in scientific research and of the purpose of the model. Models are often considered to be artifacts where also virtual models are considered beside real one. Models might also be mental models and thought concepts. Models are used as instruments in utilisation scenarios. They function in these scenarios.

2 The Notion of the Model

There is however a general notion of a model and of a conception of the model:

A model is a well-formed, adequate, and dependable instrument that represents origins. (see [8, 45, 47])

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

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

1The title of the book [4] has inspired this observation.
A well-formed instrument is adequate for a collection of origins if it is analogous to the origins to be represented according to some analogy criterion, it is more focused (e.g. simpler, truncated, more abstract or reduced) than the origins being modelled, and it sufficiently satisfies its purpose.

Well-formedness enables an instrument to be justified by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability or validation, and by stability and plasticity within a collection of origins.

The instrument is sufficient by its quality characterisation for internal quality, external quality and quality in use or through quality characteristics (see [40]) such as correctness, generality, usefulness, comprehensibility, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

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

3 Adequacy as a Generalisation of Mapping, Truncation, and Pragmatic Properties

Following H. Stachowiak (see, for instance, [33, 34]), a model is often defined in a phenomenalistic way based on three properties:

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

2. **Truncation (reduction)** property: the model lacks some of the ascriptions made to the origin.

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

We observe however that these properties do not qualify a representation as a model. The mapping and truncation properties are far too strict and need further investigation. A model must not be a mapping from some origin. Homomorphism is a nice property but far too strict in most applications. We might use representations that are not images of mappings such as a Turing machine, a system architecture, or development strategies. Furthermore, we might use representations that are not reducts of origins such as (conceptual) information system models for the variety of viewpoints users of databases might have. Truncation (or abstraction) considers a model to be an Aristotelian one by abstraction by disregarding the irrelevant. The relevance criterion is based on the purpose (or goal or function) of a model. So, truncation is far too fuzzy. Models are developed by a community of practice for utilisation by a community of practice and in a context. The utilisation depends on the intentions of users and their context. So, we observe that the utilisation of models determines (a) the kind of model, (b) the governing purposes or goals of utilisation of the model, (c) the properties of a model, (d) the amplification a model provides with extensions, (e) the idealisation by scoping the model to the ideal state of affairs, (f) the divergence by deliberately diverging from reality in order to simplify salient properties of interest, and (g) the added value of a model. The seven additional statements are combined in the mission a model has. The mission clarifies how the model functions well within its intended scenarios of usage according to its capacity and potential. The mission must be coherent with the context, the determination or specific basis of conduct or utilisation of the model, and must be acceptable for the users or – more concrete – the community of practice. Therefore, the mission clarifies the functions (and anti-functions or forbidden ones), purposes and goals of the utilisation, the potential and the capacity of the model.

4 An Agenda: Towards Adequacy of Modelling Methods

The theory of modelling is still struggling with a number of research challenges (see [40]): Adjustable selection of principles depending on modelling goals; model suites with explicit model association; development of a language culture; models 2.0; explicit treatment of model value; coexistence of theory, languages, and tools; adequate representation variants of models; compiler development for models; model families and variants. These challenges are the background behind the consternation that has been summarised at Modellierung 208 by W. Hesse (see also [11, 12]): ... but they do not know what they do ..., Babylonian language confusion and muddle; “it’s not a bug, it’s a feature” and other statements for de-facto-standards and lobbyists; why I should cope with what was the state of art yesterday; each day a new wheel, new buzzwords without any sense, and a new trend; without consideration of the value of the model; competition is a feature, inhomogeneity; Laokoon forever; dreams about a sound mathematical foundation; take but don’t think - take it only without critics; academia in the ivory tower without executable models; where is the Ariadne thread through.

This consternation and the challenges can be summarised by a research agenda, e.g. with the following problems:
Can be develop a simple notion of adequateness that still covers the approaches we are used in our subdiscipline?

Do we need this broad coverage for models? Or is there any specific treatment of dependability for subdisciplines or specific deployment scenarios?

Which modelling methods are purposeful within which setting?

Which model deployment methods are properly supporting the function of a model within a utilisation scenario?

How does the given notion of model match with other understandings and approaches to modelling in computer science and engineering?

What is the background of modelling, especially the basis that can be changed depending on the function that a model plays in some utilisation scenario?

Language matters, enables, restricts and biases (see [54]). What is the role of languages in modelling?

Which modelling context results in which modelling approach?

What is the difference between the modelling process that is performed in daily practice and systematic and well-founded modelling?

Are we really modelling reality or are we only modelling our perception and our agreement about reality?

What is the influence of the modeller’s community and schools of thought?

5 The Storyline for this Keynote

In this keynote we discuss mainly the first element of the research agenda: adequateness of models, modelling methods, and modelling as a systematic activity. So far, the adequateness notion is far too fuzzy and too wide. The keynote is based on a large body of knowledge developed on models, modelling activities, and systematic modelling. The basis of our understanding of adequacy and dependability is the case study in the Kiel compendium of models, modelling activities and systematic modelling (see [50]). This MMM approach to modelling has been investigated for models 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 introduction is based on a discussion of adequacy for two modelling methods widely used in our area. The specific utility of models follow the line given in [19, 20]. We are going to introduce a general and formal notion of adequacy. Since adequacy cannot be separated from dependability we have also to investigate it for the two modelling methods. Finally, the keynote ends with a collection of open problems on adequacy of modelling methods.

References


For details and classical database design books we refer to [1, 5, 17, 21, 22, 26, 31, 37, 38].

For details on language theory we refer to [3, 7, 18, 27, 28, 36, 43, 56].

For details of design science research we refer to [13, 15, 30, 55].

Formalisation also includes approaches to a general theory of modelling such as [9, 10, 16, 23, 24, 25, 29, 32, 35, 57].

For details of our work we refer to [2, 6, 8, 14, 39, 41, 42, 43, 44, 45, 46, 48, 49, 51, 52, 53].

Joint Proc. Workshops at Modellierung 2018, CEUR 2060, 11-18


Conceptual Model Notions - A Matter of Controversy
Conceptual Modelling and its Lacunas

Bernhard Thalheim
Christian-Albrechts University at Kiel, Department of Computer Science, D-24098 Kiel

December 10, 2017

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 center of research in computer engineering and computer science since its beginnings. It is an old subdiscipline 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 [39]). A commonly acceptable statement for a general model notion is the following one: 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 fulfills in utilisation scenarios. The function determines the purposes and goals.

CS-conceptual modelling is often related back to the introduction of the entity-relationship model(ing 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”. 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.

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. (see [25])

Community description: Conceptual modeling is about describing the semantics of software applications at a high level of abstraction. Specifically, conceptual modelers (1) describe structure models in terms of entities, relationships, and constraints; (2) describe behavior or functional models in terms of states, transitions among states, and actions performed in states and transitions; and (3) describe interactions and user interfaces in terms of messages sent and received 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 [12])

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 [2, 10]8)

Instance-integrating conceptual modelling: A conceptual model consists of a conceptual schema 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 [4])

8And 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.

Some research challenges in conceptual modeling: Provide the right set of modeling constructs at the right level of abstraction to enable successfully communication among clients, analysts, and application programmers. Formalize conceptual-modeling 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 modeling serve as analysis and development tools for exotic applications such as: modeling the computational features of DNA-level life to improve human genome understanding, annotating text conceptually in order to successfully communication among clients, analysts, and application programmers. Develop a theory of conceptual models and conceptual modeling and establish a formal foundation of conceptual modeling.

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.

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 [7])

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 [8])

Enterprise modelling and conceptual modelling: A conceptual model 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 [3])

Holistic view: In most cases, a model is also a conceptual model. (see [28])

Conceptual models as a result of an activity: We use the name of conceptual modeling 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 [26])

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 [31])

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 [17])

Semiotics viewpoint: Conceptual modeling is about describing syntax, and semantics (potentially also pragmatics) of software applications at a high level of abstraction. (see [11])

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 [18])

8The 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.
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 [24]) 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 [38])

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:

**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 aren’t 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, refineable, and translatable artifacts in software processes.

Conceptual modelling is supported by a large variety of tools, e.g. (see [21]). 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 a consistent, easy-to-use and -revise, seamless, and industry-quality tools is still on the agenda. Information and software systems become eco-systems. Modelling eco-systems are not yet properly addressed.

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 a risk of making illogical statements. Modelling must thus be based on methodologies.

2.2 Myths of (Conceptual) Modelling

Modelling and especially conceptual modelling is not yet well understood and misinterpreted in a variety of ways. It has brought a good number of myths similar to those known for software development (see [1]):

1. *Modelling is mainly for documentation.* The introduction of the conceptual modelling for database systems has been motivated by 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 is 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 [34]).

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 [43]). 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 [23]) uses models as a kernel for development of systems. Conceptual modelling ist 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 [16]).

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 commonsense, consensuses, and contexts.

A normal models (called ‘lumped’ model in [45]) 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 [45] 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, commonsense) 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.

Constructiveness. 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.

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.

2.4 Problems and Challenges

Conceptual modelling techniques suffer from a number of weaknesses. These weaknesses 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 [40, 41]) who gave a critical insight into software specification and arrive with the following general weaknesses for conceptual modelling of information and database systems:
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 the 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 deployment of information systems.

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 be developed as well 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\textsuperscript{11} or more generally some mental model\textsuperscript{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.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
version & adequate & dependable & origin & function & scenario & concepts \\
\hline
reality, world & reflection, truncation & formal, reflection & world & describe & communication, understanding & n.g. \\
\hline
community & abstraction, mapping & semantic invariance & software application & describe & construction & n.g. \\
\hline
conceptual database & mapping, homomorphy & n.g. & data, operations & describe & construction, documentation & reality concepts \\
\hline
system & abstraction & reflection, qualitative assumptions & system, objects & n.g. & construction & n.g. \\
\hline
representation & system, abstraction & n.g. & relationships & system & describe & representation, system concepts \\
\hline
enterprise & mapping, abstraction & faithful & domain & purpose-determined & understanding & concept space \\
\hline
result of activity & mapping, abstraction & n.g. & system knowledge & describe & acquisition, elicitation & domain knowledge \\
\hline
purpose-oriented & abstraction & viable, fit & any & elicitate & n.g. & n.g. \\
\hline
documentation & summary, abstraction & n.g. & data system & represent, survey & strategy development & n.g. \\
\hline
semiotics & syntax abstraction & semantics, pragmatics & software application & describe & representation & n.g. \\
\hline
document understand & mapping & closeness & application & understand by users & design & n.g. \\
\hline
conceptualise & formal representation & semantics & any & describe & representation & concept(ion) space \\
\hline
ad-hoc & selective mapping & n.g. & domain & consider problem & solving & n.g. \\
\hline
\end{tabular}
\end{table}

\textbf{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?

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

\textbf{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?

\textbf{Research question 4.} What is the role of the \textit{community of practice} in conceptual modelling? Which kind of model supports which community in which context?

\textbf{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?

\textbf{Research question 6.} Since models must not be conceptual models (see models in [39]), we might ask whether there exists a set of characteristics or criteria that separate

---

\textsuperscript{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 [30]).

\textsuperscript{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 [13, 19]). 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.
a conceptual model from a model that is not conceptual. What are 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 [39]), 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 fulfill many different functions. Instruments are partially independent or autonomous of the thing they operate on. Models are however special instruments. They are used with a specific intention within a utilisation scenario. The quality of a model becomes apparent in the context of this scenario.

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 conceptual database model is system construction. 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 limited to representation of static aspects of systems. They can also be used for 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 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 [33]), a speech act consists of uttering elements, referring and predicating, requesting activities, and causing an effect. Whether at all and which language is going to be used is a matter of controversy too.

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 not to model 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 a 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

---

\[13\] Notice however that the first introduction of conceptual data models has been oriented on a documentation scenario.
5. The conceptual modelling matrix. The modelling matrix combines the deep model with the 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 an 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 artifact 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 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. They represent 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 underdetermined 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 nonverbal 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 commonly using concepts. 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 [6, 14, 22, 20, 27]). 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 [32]), e.g., lexical or ontological, logical, or reflective.

We distinguish two different meanings of the word ‘concept’ (see [42]):

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 [42]) 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 simultaneously have different descriptions. Competing description may differently represent the same concept depending on 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 [24, 35]), concepts are defined in a more sophisticated form as a tree-structured structural expression.

\[
\text{SpecOrderedTree(StructuralTreeExpression} \\
\quad \text{(DefinitionItem, Modality(Sufficiency, Necessity), Fuzziness, Importance, Rigidity, Relevance, GraduationWithinExpression, Category))).}
\]

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. Conceptual14 modelling uses concepts which are systems of explanation. Semantification (see [9]) improves comprehensibility of models and explicit reasoning on elements used in models. It is based on name spaces or ontologies that 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 also may 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 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 then 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 [29]). Typical mandatory principles are usefulness, flexibility, sim-
plicity, realizability, and rationality. Optional conceptualisation principles are perceptability, error-proneness, and parsimony.

The principle of conceptualisation is considered to be one of the seven fundamental principles for conceptual modelling (see [15]). 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 [36]). Conceptualisation can be considered to be complete 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.

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., 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 is 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 ⊒ Model ⊕ Concepts

Conceptual modelling is not yet a common method in science (see [31]). Systems can be built 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 concepts 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 refifiable 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, the reduce the 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 [5]):

1. Thou shalt choose an appropriate notation.
2. Thou shalt formalise but not overformalise.
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

Abstract

Data are considered to be the oil of the 21\textsuperscript{th} century. They are also a rich source for many sciences, especially those that use observational data for development of an understanding behind the data. They are used to gain an insight into the discipline based on observations. This insight may result in a quantitative theory offer. The main target is however a theory that explains the data. We develop a model-backed approach to theory development based on quantitative theory offers. Models are becoming the mediator between quantitative and qualitative theories. Models can be systematically developed based on a layering approach.

1 Introduction

1.1 From Empiric Sciences to Data Science

Data science is considered to be a new stage of scientific research. Data science is based on analysis of data resources. The analysis asks the right questions with efficient processing algorithms, machine learning and cognitive computing techniques, refined statistical models, and innovative visions of how to more effectively extract the relevant data assets and scrutinise them fast with more sophisticated results. It goes beyond empirical sciences, theory-oriented research, and computational research [12]. Disciplines often use a combination of empirical research that mainly describes natural phenomena, of theory-oriented research that develops concept worlds, of computational research that simulates complex phenomena and of data exploration research. Thus, Figure 1 distinguishes four stages of sciences according to [12].

Data science discovers pattern and generates insights in data sources or data proxies. It is based on raw data and build these insights based on knowledge from the scientific discipline and application domain. It provides models, recommendations, and potential theories on how to interpret the data. It is based on a process of organising data for analysis including data proliferation, data collection organisation, cleaning, application of tools, and analysis. It may consider huge data collections as well as small data sets. The proxy data are compiled and may become ‘smart’ quantitative data for quantitative research. Data science is essentially the ‘science’ that is turning data proxies into narrative and into quantitative data. It thus develops an understanding of the data itself.

In our case, we investigate rather thin data sets. The picture is however similar to the one with very large data sets.

1.2 Data Science: From Proxy-Based Investigation to Theories

Explorative and investigative theory development (e.g. [1, 18, 20]) starts with an investigation of data sources and develop some proxy-based observation concepts and a theory offer. A theory offer is a scientific, explicit and systematic discussion of foundations and methods, with critical reflection, and a system of assured conceptions providing a holistic understanding. A theory offer is understood as the underpinning of technology and science similar to architecture theory [23] and the approaches by Vitruvius [32] and L.B. Alberti [2]. A (scientific) theory is a “systematic ideational structure of broad scope, conceived by the human imagination, that encompasses a family of empirical (experiential) laws regarding regularities existing in objects and events, both observed and posited. A scientific theory is a structure suggested by these laws and is devised to explain them in a scientifically rational manner. In attempting to explain things and events, the scientist employs (1) careful observation or experiments, (2)
reports of regularities, and (3) systematic explanatory schemes (theories).” We Figure 6 displays this

![Diagram showing the relationship between proxy-based research, quantitative research, and qualitative theory-oriented research.](image)

Figure 2: The current state-of-the-art in the data science situation. We start with some data, e.g. proxy data. We \( g_f^x \)-derive proxy concepts (or concepts) and form some proposals for \( h_c^x \)-formation of a proposal of a potential explaining theory, i.e. a theory request (or theory offer). Proxy sources can be aggregated and \( (f^c_x) \)-condensed and thus become quantitative sources which are the basis for \( (g^x_f) \)-formation of quantitative concepts. These qualitative concepts are \( (h^c_f) \)-embedded into theory offers (or, resp., theory requests for proxy requests) and are the basis for a theory offer that serves as an explanation for the theory request. Quantitative concepts can be \( (F^c_x) \)-mapped back to proxy concepts. Proxy-based research and quantitative research is well-integrated if the diagram is commuting, i.e. \( F^c_x(g_f^x(sources)) = g^x_f(sources) \).

Theories can be build on the basis of theoretical concepts which are supported by sources. Quantitative concepts should be associated with qualitative concepts. The association can only be developed in the case when the association among the data has been clarified. So far, the explanations that can be generated are mainly developed for explaining the observations made on the basis of proxies. We arrive therefore with the following problem:

**Research challenge:** How can we close the gap between quantitative theory offers and qualitative theories within the setting of data science?

### 1.3 A Typical Data Science Application

Investigative modelling at CRC 1266 [1, 16] aims at exploring and explaining transformations in societies as “processes leading to a substantial and enduring re-organisation” [1] of any or all aspects of the human social, cultural, economic, and environmental relations. Proxies are observations for main concepts in Figure 3. These main concepts need however a quantitative underpinning and a number of theoretical concepts.

![Theoretical concepts to be investigated in the CRC 1266](image)

**Figure 3: Theoretical concepts to be investigated in the CRC 1266**

#### 1.4 The Storyline of the Paper

We develop an approach to data science based on models. The ladder in Figure 1 is thus supported by models in the form depicted in Figure 4.

![The four kinds of models in scientific research](image)

**Figure 4: The four kinds of models in scientific research**

Models are instruments that function in utilisation scenarios. One of these scenarios might be the development of a theory for a theory offer. We will show in the sequel how this approach can be systematically applied to development of mediating models that close the gap in Figure 6. We start with a notion of model in Section 2. Six research questions are developed which are answered in Sections 3 and 4. Next we develop a model construction approach in Section 3. Finally, we apply this approach to data science and use models as mediators in Section 4.

### 2 Models

Models are widely used in life, technology and sciences. Their development is still a mastership of an artisan and not yet systematically guided and managed. The main advantage of model-based reasoning is based on two properties of models: they are focused on the issue under consideration and are thus far simpler than the application world and they are reliable instruments since both the problem and the solution to the problem can be expressed by means of the model due to its dependability. Models must be sufficiently comprehensive for the representation of the domain under consideration, efficient for the project.

---

1. We restrict the mindmap to main concepts and do not display the full concept network. For details see the website of...
solution computation of problems, accurate at least within the scope, and must function within an application scenario.

**Research question 1**: Can models be used for resolving the gap between theory and practice in quantitative research and theories in qualitative research?

Consider for instance the CRC 1266 application: Transformation is considered in this context as a phenomenon that requires detailed description of features and hence quantitative data are necessary for descriptions by empirical models and simulations. Models mediate between quantitative theories and qualitative theories. Models are applied in hypothetical and investigative scenarios, should support causal reasoning as well as network-oriented reasoning, and are developed in an empiric framework.

### 2.1 The Notion of Model

Let us first briefly repeat our approach to the notion of model:

A **model** is a well-formed, adequate, and dependable instrument that represents origins and that functions in utilisation scenarios. [10, 27, 28]

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

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

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

Well-formedness enables an instrument to be justified by an empirical corroboration according to its objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability or validation, and by stability and plasticity within a collection of origins.

The instrument is sufficient by its quality characterisation for internal quality, external quality and quality in use or through quality characteristics [26] such as correctness, generality, usefulness, comprehensibility, parsimony, robustness, novelty etc. Sufficiency is typically combined with some assurance evaluation (tolerance, modality, confidence, and restrictions).

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

### 2.2 Functions of Models

Models are used as instruments in certain utilisation scenarios such as communication, reflection, understanding, negotiation, explanation, exploration, learning, introspection, theory development, documentation, illustration, analysis, construction, description, and prescription. They have to fulfill a number of specific functions in these scenarios. Typical functions of models as instruments in scenarios are (a) cognition, (b) explanation and demonstration, (c) indication, (d) variation and optimisation, (e) projection and construction, (f) control, (g) substitution, and (h) experimentation [31].

### 2.3 Model = Normal Model × Deep Model

A model consists of a normal model that is combined with some deep model similar to the visible (or exterior) and invisible parts of an iceberg [16, 29, 30]. The deep model reflects the intentions of the problem world, (β) the accepted understanding within the community of practice, (γ) the context of the application domain, (ε) the background that is commonly accepted in the problem and application domain, and (δ) the general restrictions to the origins that might be considered. The deep model allows to derive a part of the justification and adequacy of a model.

The normal model reflects the collection of origins that are currently under consideration. Both the deep and the normal model are dependent on the functions that a model should play in application scenarios. Development of models is often restricted to development of a normal model under the assumption that the deep model is given by the modelling method, the context, the community of practice, and the function that the model has to play in a given scenario. The modelling methods also determined the methods that are used for model development. It might also include the utilisation methods.

**Research question 2**: Can we separate the deep model from the normal model in such a way that the model can be composed of the deep model and of the normal model?

If the answer to this question is positive then we might try to consider the model as an enhancement of the deep model. In this case, the development of a model can be layered.

**Research question 3**: How can be development of a model layered into the development of a deep model followed by the development of the normal model?

We may now ask us whether this approach is universal. The answer will be negative if the notion of model also includes models with intractable deep models, e.g. for metaphors, parables, or physical representations. We might however concentrate on models in sciences and technology.
2.4 Model Suites

Models may be given as a holistic instrument that combines all aspects into one model. The approach is often too challenging. A simpler approach is the consideration of a model as a model suite (or model ensemble) [8, 25] that consists of a coherent collection of models which are representing different points of view and attention. It is extended by an explicit association or collaboration schema among the models, controllers that maintain consistency or coherence of the model suite, application schemata for explicit maintenance and evolution of the model suite, and tracers for the establishment of the coherence.

Research question 4: Exists there a systematic approach to model development that is based on a co-development of normal models and deep models? Which additional models should be integrated into the model suite?

2.5 Generic and Specific Models

Model development does not start from scratch. We often start with generic models. A generic model [16] is a model which broadly satisfies the purpose and broadly functions in the given utilisation scenario. It is later tailored to suit the particular purpose and function. Generic models can be calibrated to specific models through a process of data or situation calibration, refinement, concretisation, context enhancement, or instantiation.

Research question 5: Can we develop normal models starting with a generic model and are they still integratable with the deep model?

If the answer is positive then generic normal models can be calibrated to specific normal models through a process of data or situation calibration, refinement, concretisation, context enhancement, or instantiation.

2.6 Data Mining as a Success Story

In [16], we developed the V-model to data mining based on a separation of the data mining process into the domain perspective with its domain world of users from a community of practice, the modelling perspective with a model world, and the data perspective in a data world. Users are interested in solution of certain problems an application world, share the context and also the scientific and technological background. The classical data model mining process uses these perspectives for a stepwise development of a model that allows to solve their problems, e.g. (1) by modelling the problem and the issues under consideration, (2) by preparing the data world for development and enhancement of models, (3) by applying data mining algorithms for pattern detection and model development, and (4) by using the model for development of some solution for the problems and thus augmenting the application domain world. The model development process itself can be understood as a multi-iterative guided procedure that has its flow of activities.

This approach extend the classical CRISP framework [4] and other approaches to systematic data mining, e.g. [15]. Each of these approaches has its capacity and potential as well as its threats and limitations. The question is now:

Research question 6: Can we generalise a data mining setting to model development for data science in such a way that models mediate between theory offers and theories?

Data analysis and model suite development currently inherit success stories in a similar application. These success stories follow some kind of a meta-pattern and result in a specific data mining process as an example of a modelling method or modelling mould. Data mining starts with exploring and understanding the data mining project, its data, and a general setting of principles of modelling. After the project and the nature of the data is understood, data are preprocessed and prepared for the application of algorithms. Next pattern within the data are investigated. This pattern analysis results in clusters, maps, association or collaboration schema among the models, controllers that maintain consistency or coherence of the model suite, application schemata for explicit maintenance and evolution of the model suite, and tracers for the establishment of the coherence.
3.1 Separating Modelling into Layers

Model suite development results in a number of models: deep, generic, specific, and normal models. Since any model has its deep elements we may start with development of this deep model. In many cases, we might use reference model or generic models (or tactical model frames like those we use in data mining and analysis). Let us investigate whether a layered approach can be applied within a five-layered separation of concern and aspects. The separation into layers generalises the approaches used in mathematics, e.g. separation by Craig’s interpolation theorems.

Classical modelling often intentionally presupposes the initialisation and intrinsic layers and assumes that these layers cannot be reconsidered and specifically changed according to the functions. We loose, however, the understanding of the model and cannot understand why the model is dependent without an understanding of these layers.

(I) The Initialisation Layer

The W*H specification pattern [9]. can be applied to model initialisation as well as includes then the following set of statements:

- a plan, function, and purpose dimension (model as a conception: ‘wherefore’, ‘why’, ‘to what place or end’, ‘for when’, ‘for which reason’) within a scenario in which the model is going to be used as an instrument,
- a user or CoP dimension (‘who’, ‘by whom’, ‘to whom’, ‘whichever’) that describes the task portfolio in the CoP and profile of users including beliefs, desires and intentions,
- an application and a problem dimension (‘in what particular or respect’, ‘from which’, ‘for what’, ‘where’, ‘whence’), and
- additionally, the added value dimension (evaluation).

The initialisation layer may also be enhanced by a contrast space for user-related separation of a model and a relevance space that is dependent on the user [11]. The contrast and relevance spaces as a form of mind-setting also define what is not of interest.

(II) The Enabling Setup Layer

The intrinsic setup defines the opportunity space and the infrastructure for the model. The results will be from one side a deep model and from the other side a modelling framework or modelling mould that guides and govern next activities. We define the context and the most of the background (the grounding (paradigms, postulates, restrictions, theories, culture, foundations) and the basis (assumptions, concept world, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense)) of the model. The context, extrinsic, and strategic dimension answers question like ‘at or towards which’, ‘where about’, ‘to what place or situation’, and ‘when’.

Additionally, we decide which methodology and environment seem to be the most effective and purposeful. The development and deployment dimension (‘how’, ‘whence’, ‘what in’, ‘what out’, ‘where’) defines the modelling methodology, i.e. the modelling mould.

(III) The Extrinsic Source Reflection Layer

Strategem We separate the deep model elements from elements of the normal model. According to the model function, the normal model represents extrinsic elements of potential origins based on their content and thus answers questions such as ‘what’, ‘with which’, and ‘by means of which’. It reflects the extrinsic theory essentials that are necessarily to be represented, e.g. conceptions or pre-conceptions from the theory that is underpinning the application. The normal model can be build from scratch (‘greenfield’ modelling). It is more usual based on the experience gained so far. The latter case thus starts with a generic or reference model that might incorporate parameters. The extrinsic source reflection layer can be understood as a tactical layer.

(IV) The Operational Customisation Layer

Generic or general normal models are adjusted to those that a best fitted to those origins that are considered for the application. The operational customisation layer is sometimes holistically handled with extrinsic reflection. Inverse modelling is the general case however. It instantiates parameters, adapts the normal model to those origins (or data sources) that are really under consideration, prepares the model for the special use and to the special - most appropriate solution, and integrates the deep model with the normal model. The normal model is typically pruned in order to become simpler (Solomonoff and Occam principled) more deviation- or error-prone. The (normal) model might be enhanced by concepts and thus become a conceptual model.

(V) The Delivery and Product Layer

The final result of the modelling process is a model suite that is adequate for origins, properly justified, and sufficient. We cannot expect that one singleton model is the best instrument for all members of the community of practice. A sophisticated model that integrates deep and specific normal models is delivered to some members. An informative model that is derived from this model can be better for other CoP members. Models delivered in the finalisation space are often enhanced by additional annotations, e.g. relating the model to the demands for members of the CoP by answering the ‘with’, ‘by which’, ‘by whom’, ‘to whom’, ‘whichever’, ‘what in’, and ‘what
out’ questions. At the delivery and product layer we thus generate a number of associated models.

### 3.2 Systematic Model Development

We combine now the five layers in Figure 5. At the left side we represent the issues for the model. The right side displays the activities and methods for the development. The corners of the octagon represent the starting and final stages as well as sources and enablers of the intermediate stages. We restrict the picture to the layered model development process.

### 4 Models as Mediating Instruments

Model-backed reasoning is thus some kind of revisable reasoning depending on the stages of knowledge. Modelling as a process starting with suites of generic models and revisable refinement according to data on hand. It should support handling of uncertainties and incompleteness of any kind and must thus make use of an integrated data management. Therefore, model-backed reasoning is properly based on layered model development.

#### 4.1 Towards Models as Mediators between Theory Offers and Theories

Models can be used to render the theory offer. At the same time models may also render a theory. We claim that these two views can be integrated. The model functions thus as mediator [17]. The rendering procedures are however different.

We envision that this integration can be based on the mappings in Figure 6. Models can be understood as being composed of model concepts that are supported by data sources. We can now distinguish $f_{c,q}^m$-mappings at the same level, $g_{c,q}^m$-mappings between sources and concept, $G_{c,t}^m$-mappings from concepts to supporting sources, and $h_{c,q}^m$-embedding mappings from concepts to theory offers, models, or theories. Quantitative concepts are indicators or general quantitative properties. Model concepts are already abstractions from those quantitative concepts. Theoretical concepts in Figure 3 are elements of a theory that is currently under development. The research task is the harmonisation of the two mappings $f_{c,q}^m$ and $f_{c,t}^m$. This harmonisation can be based on the mappings for supporting resources $f_{s,m}^q$ and $f_{s,t}$ if some commuting diagram properties are valid for model concepts and the model.

![Figure 5: The layered model development framework](image)

![Figure 6: Models as integrating and mediating instrument for data science](image)
a model has however also a feedback turn both on the theory offer and the theory. The model is then at the same time an instrument, a mediator, a companion, a middle, and a medium [5]. The model thus becomes an investigation instrument.

4.2 Evidence-Based Reasoning in Data Science

Let us finally discuss an obstacle of quantitative research that results in some obstinacy of models. Theory and model development are in both cases evidence-based due to the way how they are derived from proxies. The Observation-C(aims/Hypotheses)-E(vidence)-R(asoning) pattern [7] starts with some observations and detection of hypotheses about these observations. Hypotheses are transformed into claims and research questions that form a research agenda. Evidence are then either systematically elicited from data, from previous investigations, or from the belief and knowledge space. Reasoning should then connect evidences to the claims. The results is some kind of Bayesian formulas representing the claim with the evidence. Evidence-based reasoning combines therefore inductive and abductive reasoning. It is enhanced by Occam’s razor approaches [19] that allow to finalise the model development. It can be combined with Solomonoff induction [24] that enhances a result (1) by conduction of experiments that will test the claims and (2) by provisionally accepting the claim if the experiment confirms the claims. The reasoning schema follows the induction/abduction-retrospection-observation-concepts-theory-offer pattern. It can be combined with Epicurus’ principle of keeping multiple explanations that allowing consideration of several models and theories as long as they are consistent with the observations.

OCER pattern are the basis for evidence-based proxy reasoning (e.g. in the CRC 1266 [1]) that follow positive evidences. Evidence-based reasoning is based on the following principles:

1. Models represent only acceptable possibilities (each model captures a distinct set of possibilities to which the current description refers) which are consistent with the premises and the knowledge gained so far what makes them intrinsically uncertain because they mirror only some properties they represent.

2. Models are proxy-driven (the structure of the model corresponds to the proxies it represents. They might also include abstractions such as negation.

3. Models represent only what has been observed and not what is false in contrast to fully explicit models (that represent too what is false).

4. The more proxies that are considered, and the richer those models are, the more accurate the world view is.

5. We use pragmatic reasoning schemata (e.g. A causes B; B prevents C; therefore, A prevents C).

Evidence-based reasoning thus makes a difference between deterministic conclusions (A cause B to occur: given A then B occurs) and ordered sets of possibilities (A enables B to occur: given A then it is possible for B to occur).

5 Conclusion

Models are one of the main instruments in science and technology. They support reasoning in various forms, e.g. by systematic revisable modelling based on data and as an associated collection of models;

This paper develops an approach for development of fully fledged models (a) with extrinsic parts similar to usual (normal) models and (b) with intrinsic parts which are typically hidden in the modelling approach, in the background and context of the model, and in the intentions behind the model. While making this explicit, we are able to use a model as a problem description and to compute the solution of the problem under consideration directly from the model. The paper presents the first part of this solution. The development of corresponding tools is the topic of a forthcoming paper.

The presented layered approach should not be applied as a 1-2-3-4-5 waterfall sequence of activities. Rather, model development and model utilisation use an evolutionary approach that returns to previous steps whenever sufficiency characteristics of models become problematic within the application domain. The layers can however be considered as phases of development. We notice that our layered approach also supports model revision and model evolution. It can also be used for model migration and model reengineering. The layered approach seems to be combinable with modelling cultures, e.g. those that can be observed for our first case study [14]. Figure 5 represents the ‘greenfield’ or glassbox development with model development from scratch. A similar picture can be given for ‘brownfield’ model development, i.e. model redevelopment, revision, and migration. Blackbox model representation uses only the first and the last layers.

The layered approach is based on a separation of concern within an initialisation layer, within an intrinsic and this implicit setup layer, within an extrinsic and thus explicit source reflection layer, within an operational customisation layer, and finally with a model delivery layer.

We conjecture that a similar layered approach can be developed for model utilisation. The layers will be different but oriented on the usage of a model as an instrument. It can also follow the solution story initialise-prepare-investigate-do-deliver that is used for layered model development.
The main result of the paper is a systematic approach that closes the gap between theory offers and theories in data science. The approach is based on the mediating function of models between theories and theory offers.

References
Modelling Cultures

Hannu JAAKKOLA a,1 and Bernhard THALHEIM b,2

a Tampere University of Technology, P.O.Box 300, FI-28101 Pori, Finland
b Christian-Albrechts-University Kiel, Computer Science Institute, 24098 Kiel, Germany

Abstract. Modelling is an essential part of information systems development. Models are used for communication between interest groups and inside development teams. Models are also used for transferring baseline artefacts between development phases. Models are mainly developed by humans, which represent certain cultures - national, enterprise, professional, team, project etc. Because of that we claim that models, as well as many other information systems related artefacts are culture dependent. The models are born in certain context and these must be also interpreted by taking the original context into account. In our earlier studies we have analysed the effect of culture in information systems development: culture related aspects in general level, in information search and interaction and in web information systems. These studies acts as a basement for this paper having focus in modelling. Because of that we shortly answer to the question "How cultures differ from each other". This reviews and synthesis is generally accepted frameworks for cultural analysis. In addition we shortly open the results of our earlier studies. Because modelling is a human activity, as well as information systems are used by humans, we feel important to build a model of humans in information systems development and use context. The findings of culture analysis are transferred to modelling practices via our framework that defines model as an instrument transferring elements of its development context to the models - we discuss about the roles of normal models, deep models and modelling matrix. Finally we will concentrate in the problems of cross-cultural modelling using selected national cultures as an example.

Keywords. culture-dependence of modelling; deep model, modelling matrix; multi-cultural system development;

1. Cultural Differences

1.1. The Layered and Dimensions Approaches

G. Hofstede [12] defines culture adapting the layered structure of Maslov pyramid (Figure 1). He uses the term “mental program” to describe the characteristics of each layer. The lowest level - operating system - is common for all humans. The second layer - collective program - is learned and remains same in a collective group of people and indicates the culture. The “onion model” on the right side of the Figure 1 describes the main elements of the culture. Values are the core of the culture. Rituals are collective

1hannu.jaakkola@tut.fi
2thalheim@is.informatik.uni-kiel.de
activities that are essential in a culture and indicate the membership of the group. Heroes are highly prized examples in a culture and indicate positive values of it. Symbols are words, gestures and objects that are common for those share the culture. Practices are manifestations of all other elements of a culture.

R. Lewis [21,22] applies the pyramid model in his culture analysis (Figure 2). As seen in the Figure the culture layers between Finland and Germany are different. Both countries in Lewis’ classification (discussed later in this paper) are close to each other and belong to the group of linear-active and data-oriented cultures. In spite of that, the values and core beliefs - the core of the culture - are different.

In our paper [17] we have introduced three methods to be used in recognizing cultural differences cultural differences: the 6D model of Hofstede, Lewis “triangle model” and Hall’s high/low context culture model. In addition to these there are several other ones; most of these overlap with the former ones and do not provide additional value to the analysis. Figure 4 illustrates the classification principles of Hofstede’s model.

The model of Hofstede basis on the analysis of six cultural dimensions [11,12]:

- Power Distance (PDI): the extent to which power differences are accepted.
Figure 3. The Hofstede 6D model [11]

- Individualism / Collectivism (IDV): the extent to which a society emphasises the individual or the group.
- Masculinity / Femininity (MAS): refers to the general values in the society - hard/soft values.
- Uncertainty avoidance (UAI): refers to the extent that individuals in a culture are comfortable (or uncomfortable) with unstructured situations.
- Long-term / Short term orientation (LTO): refers to the extent to which the delayed gratification of material, social, and emotional needs are accepted.
- Indulgence / Restraint (IVR): acceptance of enjoying life and having fun vs. controlling the life by strict social norms.

The country comparison tool[^3] provides access to the database collected by Hofstede during the decades. The data covers culture values of most countries in the world. The left side of Figure 4 indicates the similarity of Finland and Germany. Meaningful difference is in LTO and MAS values, slight difference in IDG. The German work to reach results, which are more long-range than the Finns (LTO). They appreciate material values more than Finns (MAS) and live in the atmosphere, which is more puritan than in Finland (IVR).

1.2. Interaction and Collaboration

The Lewis’ model [21,22] focuses in analyzing interaction and collaboration activities of people. The nationalities locate in the corners and sides of a triangle. The corners represent the basic stereotypes: linear-active, multi-active and reactive. Linear-active cultures are data oriented (decisions are based on facts and official sources) and can be described by terms cool, factual and decisive planners. Reactive cultures are “listeners” - they base their behavior in more rich base of information sources (oral information from social networks, family, friends, ...) than people in data oriented cultures do. In communication they are not active members of the interaction; listening and reacting is typical to them in dialogues. Terms courteous, amiable, accommodating, compromiser and good listener describe people in reactive cultures. They are also usually members of collective cultures in Hofstede’s classification. Multi-active cultures are dialogue oriented. Like

people in reactive cultures they use rich set of information sources, but especially prefer oral information. Multi-active characteristic means doing several things at once, being extrovert and being active member of dialogues. Terms warm, emotional, loquacious and impulsive describes them.

1.3. Context Dependence

The Hall’s [9] model divides cultures according to the importance of context recognition in communication. Context means the extent of “wordless” communication included in the messages. High importance of context in communication indicates the importance of the membership in a collective group – i.e. collectivistic group culture in Hofstede’s classification. In high context cultures, the meaning of the message relates to the context in which it is presented. The group members know a variety of details included in the message without explicit messaging. The low context cultures are opposite. These cultures prefer punctual and clear messaging. All information is clearly included in the message and need for knowing the context is minimal. Low importance of context indicates highly individualistic society in Hofstede’s classification. Finland, as well as all Scandinavian countries, and Germany belong to the category of low context cultures; Japan and Arab counties instead are typical high context cultures. Somewhere in the middle of the continuum are USA and England, in which it is also typical to use words and sentences with hidden meaning.

1.4. The Storyline of the Paper

In this paper we start with an investigation whether cultures have an impact on models, i.e. on model development and model utilisation. Models can, for instance, be used as a means for communication. A hypothesis could be that models are a stability kernel among different people. The opposite hypothesis states that models are culture dependent. Section 2 discusses the relationship between modelling and cultures. Section 3 introduces a general model notion and a separation between the normal model and the deep model. We illustrate the cultural dependence for different kinds of schemata. Section 4

Figure 4. The Lewis’ triangle model (modified from [21])
investigates the cultural differences for modelling styles. It is shown that the models developed in different cultures might also be different although the application is the same. Culture thus determines how models are developed and how models are used.

2. Cultures Influence Human Modelling Activities

2.1. Modelling is Different Worldwide

While zapping through textbooks from different countries on database analysis, design, and development we observe that the same topics and the same application tasks result in rather different database schemata. So, what causes these differences? What styles are preferred where? Under which circumstances one model is better than the other? Which detailedness is the best? Shall we concentrate on typical structures only? How exceptional cases are handled?

We observe also different modelling pattern and styles beside language differences (ER-like, UML, ORM, NIAM, IDEF, etc.). Students who got first lectures in object-orientation develop completely different schemata than those who got introduced to functional or procedural paradigms. Some companies like Ploenzke or SAP have a completely different way of representing the same application.

Moreover, the same language paradigm is often modified and extended. For instance, there are more than 50 extensions of the entity-relationship approach. Most of them are actually incompatible. Many modelling languages exist in a large variety of dialects what makes knowledge transfer and communication difficult. One reason might be that the ways of thinking, of modelling, of controlling, of working, and of supporting are different in different communities and thus result in different environments and thus in different cultures. Another reason might be the insufficiency of a language. In this case, we can use language pluralism and develop model suites [29].

Modelling might also follow different paradigms and postulates. It might use different not combinable theories. Modelling is biased by the developers and their educational and professional background [33]. It is typically laden\(^4\) by concepts that are to be represented, by its community, by the context into which the model is set, and by the way of utilising a model. If we compare these factors influencing modelling with the culture notion then we realise that all these factors are culture-driven.

So, we may conclude that organisation, professional, educational, and finally national cultures influence the outlook, the content, the adequacy and dependability of a model. It is not only the behaviour of people that is governed by the cultures but also the development and utilisation of tools that is governed by the culture. Models are instruments that are used in utilisation scenarios. Communication is one of the main scenarios. Models are used similar to utterances in natural languages in this scenario.

2.2. Culture Sensitivity in Information Systems Development

We have handled the topics related to information systems (IS) development in multicultural context in several papers. The papers handle information systems development from different points of view. Culture related aspects affect in both the development and

\(^4\)This concept has been considered in detail by H. Kangassalo (and J. Palomäki) in the EJC’15 keynote “Definitional conceptual schema - The core for thinking, learning, and communication” at June 11, 2015.
the use of IS. The development work is made in multi-cultural distributed teams, in which it is important to recognize the dynamics of the team in decisions related organizing the work, leading the team and managing the development project. Transfer towards cloud based ecosystems and web information systems (WIS) makes recognition of the end-user base more difficult. In requirements engineering phase we have more and more often “faceless” clients from different cultures and from different parts of the world that must be served by the WIS [15].

Our earlier studies cover general aspects in IS development [14], information and query-answer related aspects [17] and web information systems design related aspects including database design and conceptual modelling [16]. These papers provide a “handbook” type list of findings to support IS development in multi-cultural context. Our analysis applies the interpretations of human behavior using Hofstede’s dimensions and Lewis analysis. It also acts as an evidence to the applicability of culture analysis and stereotyping methods to guide IS development for multi-cultural context. The realization of the findings is included in the requirements engineering phase, which transfers them to non-functional requirements in the requirements specification of the IS.

We have approached the topic via Hofstede’s and Lewis’ models. Hall’s model provides some new aspects to the analysis, which are worth of more studies. Low context cultures are tended to demand exact communication. Our hypothesis is that IS development in such cultures indicate clear and unambiguous user interface, whereas high context cultures are tended to accept some ambiguities and complexity in it. Low context indicates linearity, high context multi-activity.

### 2.3. Information Systems Modelling and Culture

In IS projects models are means for communication - transferring duties and work items trough the life cycle of the IS and supporting interaction between the interest groups. We defined modelling to be a kind of solution to the problems of communication. Modelling languages are culture independent unlike natural languages. However, our hypothesis is that the use of them and the structure of the models indicates culture of its user. In IS development models transfer system related knowledge between interest groups. Because most of the modelling techniques used in practical work are semi-formal, the lacking exactness opens door for misunderstandings. In addition the sender’s and receiver’s ability to interpret the model may vary; one of the reasons is culture. Interpretation of the models is also context sensitive (i.e. in different contexts the interpretation may vary). The model itself is a construction of concepts and individuals according to their internal concept handling mechanism interpret it. In our paper [13] we introduced a hypothesis that also this mechanism is culture dependent - that what a Finn finds in a (conceptual) model would be different to the findings of a German or Japanese. In the same paper we have listed problems related to communication and collaboration in multi-cultural context: (1) behavioral patterns of people are different, (2) concept creation and handling is different, (3) language of communication is different, (4) communication includes opportunity to serious misunderstandings and (5) transformations (transferring the message from one language to other) may change the meaning of the message. All these problems fit to IS modelling, too.
2.4. Modelling of a Human Being, Team Dynamics and Organization Culture

In culture adaptable information systems development context there have been efforts to model its user. In adaptable IS the system includes a model of the user. If this “user model” is equal to the real behavior of the user, the system may adapt its operations according to the expectations of different users. In culture adaptable IS this model includes culture related factors.

One of the best-known model is MOCCA environment developed by K. Reinecke [27]. MOCCA is an application that can adapt ten different aspects of its UI with 39366 combination possibilities altogether. MOCCA acts also as an example of the technical implementation of the flexible user interface in information systems design. The user model takes into account the cultural background of the user including user’s cultural adaptation because of the influence of foreign cultures. The user profile basis on the following parameter: MAS, UAI, PDI, LTO, IDV, year of birth, political orientation, social structure, religion, education level, familiarity to certain form of education, computer literacy and gender. External dependencies cover nationality of the person, his/her mother’s and father’s nationality and language skills (mother tongue, foreign languages). Dynamics of the model basis on the former length of stay under the effect of the foreign culture.

M. Phaedra and M. Permanand [26] have introduced a student model that takes into account his/her demographic factor values. The person (student) has simple arguments: identification, age and gender. The dimensions of the model fall into five categories that describe particular contextual categories: geographical aspects, religion, ethnic background, education level (including school - note the importance of school as a root of an important source of information in dialogue oriented and reactive cultures), and particular physical environment settings and terrains. External properties cover - as in MOCCA - parent data including their occupation (social group) and native language. The model does not include any aspects creating dynamics, if changes in the parameter values not counted. The model neither includes any cultural factors derived from stereotype models.

G. Dafoulas and L. Macaulay [6] have modelled the dynamics of multi-cultural virtual software development teams. The model lists a variety of factors that to take into account in management of the team and organizing the work in it. They emphasize that each individual is a member of multiple cultures (Cultural profile category): one or more national/ethnic cultures, one or more professional cultures, a functional culture, a corporate culture, and a team culture, among others combined to individual (personal) characteristics. They have seen, especially in multi-cultural distributed teamwork the importance of professional and functional culture: “software professionals worldwide belong the computing subculture, which is stronger than any other culture”. A Russian software engineer (professional culture) would be more similar to an American peer than to a Russian marketing manager (functional culture). The model of cultural dimensions in virtual software teams does not specify the properties of an individual (professional) but a roadmap to manage the team. Human resources category includes PDI, UAI, IDV, time difference between members, trust level between team members, concept of space (Lewis) and material power (goods that create or indicate power). The required skills interact with human resources. The required skills category covers communication skills, participation activity, leadership, conflict resolution, problem solving, decision-making, goal setting and motivation. Team development category covers the improvement of required skills by taking into account first the team profile (diversity level), the role profile
and finally task profile (requirements); the improvement is a continuous iterative process. Although this model belongs more into the category of “management and leadership models” it points out important aspects that indicate personal properties to be included in the model of a software engineer.

Our paper [14] includes a simple user model structured as a mind map. This model indicates the important factors of an individual to be taken into account in developing adaptive information systems. The personal properties category of the model covers personality profile (nine general factors and three dialogue preference related factors), work profile (six factors) and education profile (three factors). The portfolio category includes task related parameters, user involvement description, type of collaboration and restrictions to take into account.

E.G. Blanchard et al. [5] use very similar approach in their conference paper related to intercultural communication. They have found a remarkable (literature based) evidence which shows that the way people interpret and react to their environment significantly differs from one culture to another and that wide range of human activities and situations influenced by culture. In spite of that, the human-related technologies have not accounted for culture. Western context dominates in design and solutions, which are tested and validated on Western samples. In their paper Blanchard et al. (2013) introduce a simplified conceptual model of intercultural communication. Cultural elements concept class in the model covers cognitive cultural elements and cultural non-verbal communication (body language) related aspects. Non-cultural (innate) elements concept class includes behavioral primitives (gestures, postures, facial expressions) and some innate non-verbal communication elements. Additional concept classes cover the role of context, culture and cultural group, enculturated individual aspects, cultural group cohesion and a variety of descriptors.

B.S. Parumasur [25] handles the problems related to organizational development (OD). The paper states that American and European consultants have developed most of the OD practices. Because of that, cultures collide in different cultures. Contextualized and customized approach is needed: The skills readiness acquired at school varies (abstract thinking, team skills, entrepreneurship, technical, language, ...), motivation factors vary between cultures (emergent/mature), gap to the welfare plays an important role. In all change and improvement processes gap between local values and proposed interventions must be recognized. However, the evolution of the political and economic climate changes the values rapidly; globalisation leads to adoption of foreign influence (see [27]). The paper concludes to a model, which indicates organisation’s readiness to changes; the model applies Hofstede’s 6D model in the following way. PDI: High PDI indicates acceptance of social and economic gaps, acceptance of inequality, acceptance of centralization, valuation and respect to authorities and hierarchical relationships. In high PDI cultures close supervision is needed. PDI indicates also suitability of participative / non participative decision making. Large PDI is associated with collectivism and (lower national wealth), small to individualism (and greater national wealth). UAI: High UAI indicates resistance to changes. Combined with high PDI it reflects the responsibility of an organization instead of individuals. UAI links to formalization, the need for formal rules and specialization. IDV: IDV indicates the acceptance of person level benefits (salary differentiation based on productivity) and the importance of interpersonal relationships. High IDV leads to a need to explain every act in terms of self-interest. IDV relates also to the directness / informality of feedback (performance improvement vs. de-
stroying the harmony in certain conflict situations) and the aim to avoid face losing (in a group. MAS: Masculinity relates to career advancement and salary vs. social aspects of work. It indicates also differentiation of gender roles. Relatively high MAS and Weak UAI justifies the high level of achievement motivation.

2.5. Revisiting Cultural Studies

The different approaches to dimensions of cultures have been combined, systematised, and generalised in [17]. The result is a 11-dimensional Kiviat graph in Figure 5. The dimensions could be used to derive guidelines for web information system development what has been illustrated by the dimension values for Finland, Germany, and Japan.

![The Kiviat graph of the cultural dimensions of people](image)

**Figure 5.** The Kiviat graph of the cultural dimensions of people

The graph can be extended by dimensions from other culture models. Instead we can use this combination also for derivation of other properties. The models by Hall, Hofstede, and Lewis cannot be solely considered. Additionally, combined properties cannot be derived. For instance, the triangle model [21] does not allow to reason on the cultural distance. The cultural distance is classically the differences of cultural values and is expressed as a function of differences in values of some of these dimensions, e.g. Euclidian or Mahalanobis distances. The triangle model also mixes three dimensions in Figure 5: the kind of being active, the kind of reacting on the partner and the way how tasks are performed. If we compare the distances between German and Japan people from one side and between Japan and European Russian people then first one is small in the triangle Lewis model whereas it is larger in Figure 6. The distances between Northern German and Japan people and between Japan and European Russian people in Figure 6 match far better with observations in normal life.
So, the eleven dimensions in Figure 5 provide a better means for supporting also cross cultures. They can be easily extended by Victor’s model [23]. Since some of the dimensions are important for some aspects and not relevant for others, we can use views for these aspects. For instance, if we concentrate on the communication and interaction level then the Lewis triangle or the three-dimensional characterisation together with high and low contexts should be taken into account. Models are also developed for communication scenario. Therefore, we can abstract from Hofstede’s approach in this case. If we consider however the modelling activities then Hofstede’s dimensions become more central.

3. Models and Modelling

3.1. A Model is an Adequate and Dependable Instrument

Modelling is a topic that has already been in the center of research in computer engineering since its beginnings. It is an old subdiscipline 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. A model is a well-formed, adequate, and dependable instrument that represents origins [31,32]. Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its community of practice within some context and correspond to the functions that a model fulfills in utilisation scenarios.

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

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

3.2. Database Modelling and Cultures

We already developed a number of stereotypes for database schemata [15]:

(a) strictly hierarchical (ER-like) database schemata,
(b) schemata with local viewpoints that reflect the needs of some stakeholders (local-as-view approach),
(c) variants of XML-schemata, Bachman diagrams,
(d) sets of local database schemata with the requirement that the corresponding database schemata is simply the union of the set (global-as-view based on local viewpoints),
(e) sets of personalised views based on local database schemata with some kind of coherence constraint among all views (rigid global-as-view) etc.

These schema stereotypes can directly be associated with stereotypes as shown in table 1.
Table 1. Cultural stereotypes, kinds of database schemata that are potentially preferred, and potentially useful database schema stereotypes [15]

<table>
<thead>
<tr>
<th>Cultural stereotype</th>
<th>Preferences</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Power Distance</td>
<td>completely specified and well-formed, easy to understand and persistent database schema</td>
<td>(a)</td>
</tr>
<tr>
<td>Low Power Distance</td>
<td>freely configurable database schema that is adaptable to current needs and preferences</td>
<td>(d)</td>
</tr>
<tr>
<td>Individualism</td>
<td>my own database schema according to my and only my preferences (work profile, education profile, personality profile, security profile)</td>
<td>(e)</td>
</tr>
<tr>
<td>Collectivism</td>
<td>commonly agreed database schema reflecting all elements within a group according to the collaboration style</td>
<td>(b)</td>
</tr>
<tr>
<td>Masculinity</td>
<td>restriction to essential elements and only those, strict structuring schema with additional and optional elements, with exploration opportunities, personalised schemata</td>
<td>(a)</td>
</tr>
<tr>
<td>Femininity</td>
<td>complete schema with all elements, hierarchical structuring, more linear, well-scoped sub-schemata with simple reference to main schema</td>
<td>(d), (e)</td>
</tr>
<tr>
<td>Uncertainty avoid-</td>
<td>complete schema with all elements, hierarchical structuring, more linear, well-scoped sub-schemata with simple reference to main schema</td>
<td>(a), (d)</td>
</tr>
<tr>
<td>Uncertainty tolerance</td>
<td>extensible schema, flexible schema style, web-like schemata</td>
<td>(c), (e)</td>
</tr>
<tr>
<td>Long-term culture</td>
<td>all potential elements are reflected as well as all viewpoints, focused (oil stain) schemata</td>
<td>(a), (b)</td>
</tr>
<tr>
<td>Short term culture</td>
<td>handy schemata depending on current use and smooth integration of them, decomposable schemata</td>
<td>(e)</td>
</tr>
<tr>
<td>Indulgence</td>
<td>schema with a central part containing all necessary elements and further elements that might of use in future puritanical schemata without any non-essential elements</td>
<td>(e), (c)</td>
</tr>
<tr>
<td>Restraint</td>
<td>puritanical schemata without any non-essential elements</td>
<td>(a)</td>
</tr>
<tr>
<td>Linear-active culture</td>
<td>schemata with step-wise exploration of all its aspects</td>
<td>(b)</td>
</tr>
<tr>
<td>Multi-active culture</td>
<td>different variants of the global schema for parallel integrated work</td>
<td>(d), (c)</td>
</tr>
<tr>
<td>Reactive culture</td>
<td>completely fledged schemata with all details and views for later work</td>
<td>(d)</td>
</tr>
</tbody>
</table>

3.3. The Normal Model, the Deep Model, and the Modelling Matrix

Model development is typically based on an explicit and rather quick description of the ‘surface’ or normal model and on the mostly unconditional acceptance of a deep model [18]. The latter one directs the modelling process and the surface or normal model. Modelling itself is often understood as development and design of the normal model. The deep model is taken for granted and accepted for a number of normal models.

The deep model can be understood as the common basis for a number of models. It consists of the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, common-sense) of modelling. It uses a collection of undisputable elements of the background as grounding and additionally a disputable and adjustable basis which is commonly accepted in the given context by the community of practice. Education on modelling starts, for instance, directly with the deep model. In this case, the deep model has to be accepted and is thus hidden and latent.

This separation into normal model and deep model provides a means to distinguish two different logical theories behind: entailment or logical consequence for normal models and semantic presupposition for deep models. The pragmatic presupposition additionally consider the relation between a model developer or user and the appropriateness
of a model in a context. Inferences become then context- and scenario-dependent. Models are thus also evaluated based on their added value and not mainly evaluated based on their validity or correctness\(^5\).

A (modelling) matrix is something within or from which something else originates, develops, or takes from. The matrix is assumed to be correct for normal models. It consists of the deep model and the modelling scenarios. The modelling agenda is derived from the modelling scenario and the utilization scenarios. The modelling scenario and the deep model serve as a part of the definitional frame within a model development process. They define also the capacity and potential of a model whenever it is utilized. Deep models and the modelling matrix also define some frame for adequacy and dependability. This frame is enhanced for specific normal models. It is then used

### 3.4. Why Conceptual Modelling is (Not) Acceptable

A conceptual model is an adequate and dependable artifact or instrument that

- is enhanced by concepts from a concept(ion) space,
- is formulated in a language that allows well-structured formulations,
- is based on mental/perception/situation models with their embedded concept(ion)s, and
- is oriented on a matrix that is commonly accepted.

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 [35]. Conceptual modelling is either the activity of developing a conceptual model or the systematic and coherent collection of approaches to model, to utilise models, etc.

Conceptual modelling is not in the center of development activities in all countries. Observing the history of the ER-conferences on conceptual modelling for three decades, we discover that it is still a central and attracting topic in Europe with a movement from North to South over three decades, did not change in Middle East, lost its attraction in Northern America and partially also Southern America, and has not been a central issue in the rest of Asia. So, one might ask why this attention and changes happened. One answer could be the loosing interest and importance in this approach. Another answer could be however that development is based on rather different styles in different countries, i.e. is culture-dependent. A third answer might be that models are latent and not explicitly stated what is also culture-dependent.

### 4. Cultures in Modelling

#### 4.1. Models, Languages, and the Background

P.P. Chen [3] made the observation that the entity-relationship modelling language follows specific construction rules of the Old Egyptian and the Chinese language. This modelling language can only represent simple English sentences. Later, [10] could show that the extended ER modelling language HERM [28] covers the main categories in the English language. So, languages enable and hinder modelling.

\(^5\)“All models are wrong. But some of them are useful.” (often cited as a phrase by G.E.P. Box [2])
The background and especially the grounding are often incorporated into the deep model that is not explicitly communicated. For instance, the grounding for information system models includes DBMS and CE paradigms and postulates, set semantics, database theory, DBMS solution layering, DBMS technology (theory and culture), graphics and diagrammed canonical representation, ER canon, data-first-methods-second paradigm, database-approach-as-guide, etc. The basis of the model house in Figure 7 includes also a number of specific assumptions and commonly accepted practices. For instance, database modelling can be based on specific extended ER language, hidden basic types, views as derived (algebra) expressions, concept fields, extended ER thought style, parametric generic concept field, Indo-European utterance composition, extended ER development methods, extended ER heuristic rules, transformation techniques to other deep models, and extended ER tools. Additional assumptions are Salami slice tactics, the believe that functionality comes later, a rigid separation into firstness of syntax and secondness of semantics, visualisation, well-formedness (including lazy normalisation), extended ER pattern, reuse of experienced solutions (as exemplars), and flat two-dimensional schema representation. Global-as-design is commonly accepted in the database community. The global schema is the main result. Views are then defined on top of the schema by algebraic expressions in order to cope with user viewpoints. Global-as-design has its limitations. The combination with local-as-design, e.g. for BPMN diagram suites, becomes rather difficult.

The language as an essential part of the basis, the grounding and also the other choices in the basis are acceptable in one community of practice and might be completely unacceptable for others. So, the deep model and partially also the matrix are part of the cultural setting. It is often claimed that the organisation and education cultures rule this setting. The other dimensions of culture [14] are, however, not less important.

4.2. Models as a Sufficient and Necessary Means of Communication

Communication or exchange of data/knowledge/information is one of the main scenarios where models function as a content that is communicated. Models support learning, description, prescription, prognosis scenarios as well. Communication involves several partners with their own background and culture and is based on a relationship between these partners. Each of these partners also interprets the model in a specific way based on hidden background and the specific treatment of the four directives, i.e. presupposes a specific conditional framework against which the model makes sense. The explicit part of the model is the normal model. The implicit or pragmatic part is the deep model. The matrix of the model combines the deep model and the specific ways of model usage according to the considered scenarios. We shall see in the sequel that the pragmatic part is interwoven with the culture.

Models for communication must follow felicity resp. appropriateness conditions, i.e. conditions on well-formedness. Models and especially representation models must be developed on the principles of visual communication, of visual cognition and of visual design [15,24,30]. The culture of modelling is based on a clear and well-defined design, on visual features, on ordering, effect, and delivery, and on familiarity within a user community.

The meaning of models is typically combining four parts: (1) the literal model meaning ("what is said"), (2) the conveyed model developer meaning, (3) the model user meaning, and (4) and the implicated meaning ("what is implicated"). The implicated
meaning might be conventional or non-conventional. Non-conventionality of models includes what is the implicated content within the model and what has been left aside (non-conversationally). The first one can be general or particular. These different kinds of model content influence the model informativeness. The first part is triggered by the meaning of the model constructs and the model design as a statement. The second, third and fourth meanings are human related and thus depend on the culture of the people involved. The model itself should have a holistic interpretation.

Models in communication scenarios have to follow general principles and a set of rules called maxims of model communication. They are, in general, communication implicatures from [7,8].

**Cooperative principle:** Make your model such as is required, at the stage at which it occurs, by the accepted purpose of the model within the communication scenarios in which it is are deployed. This general principle has several sub-principles called “Maximes of Model Communication”

- **Maxim of quantity:** Make the model as informative as is required for the current purpose of the model do not make the model more informative than is required.
- **Maxim of quality:** Try to make model as valid as possible do not incorporate aspect that are invalid. All constructs need an adequate evidence.
- **Maxim of relation/relevance:** The model and all its elements must be relevant.
- **Maxim of manner:** The model should be parsimonious and perspicuous, i.e. economic and well-formed. Any obscurity of expression and ambiguity are avoided.

**Implicated maxim of efficiency:** The maxim of quantity requires that a model should be sufficient for an understanding by the model user (I(nformation)-principle) [20]. From the other side it requires that the model should contain all necessary elements for an understanding by the model user (R(elevance)-principle). The model represents as much as the modeller can and must. The M(odality)-principle assumes that non-normal, non-stereotypical situations by the model that contrast to normal situations are given in an explicit and understandable form. The P(recision)-principle requires that a model is only at a precision level according to purposefulness. The B(revity)-principle prefers smaller models over longer, complex ones even though it has to be interpreted in a vague way. In some cases, vague models might serve better its function.

These maxims are explicitly stated by the sufficiency characteristics which allow to evaluate the quality in use, the external quality and the internal quality. Based on the modelling style we are able to reason on negation. A typical, however, often impractical approach with the strongest interpretation is the closed-world assumption in modelling that allows to conclude about the meaning of missing parts in the model. This assumption follows [4] (“Dire et ne pas dire”). The maxim of efficiency is often based on ‘hidden’ sub-models (called in the sequel ‘deep model’) which are taken for granted within a context and background by a community of practice.

We observe that these maxims are accepted in different cultures in a different way. So, the pragmatics of models depends on the culture. Moreover the deep model is governed by this pragmatics. The principles cannot be satisfied at the same time. Which principle is preferred also depends on the community of practice and thus on their culture imprinting.
4.3. Cross-Culture in Modelling

The adequacy of models has been handled in a strict or flexible way. Some model notions require a mapping property as a strict form of analogy. At the same time truncation or abstraction is required instead of focus. Also well-formedness is often taken more tolerant. Purposefulness is however commonly accepted. A similar observation can be made for dependability of models which is often only implicitly assumed. All model notions analysed in [34] use an implicit deep model that is undisputable. A rather surprising difference is the explicit statement on quality characteristics which have to be satisfied. At the first glance it seems that the list is random.

Let us, however, analyse the German database or information system books which are often used for teaching and papers and books from US where the first are published in the ER conferences since 1992. We observe that there are common properties applicable to both. There are also properties than can be only observed for one side. some properties are out of scope or out of style although they are important for information systems.

To make these different style more clear we shall use Lewis’ horizons of communication [21] in Figure 8. There are general properties that are commonly accepted by the two communities. There are also properties that are out of style or out of scope. There are also typical German and US properties that can only be observed for one of the communities. For instance, the US approach to development is often based on an 80:20 principle, i.e. the schema is left open for further development. The normal case is mainly considered. The opposite is observable for the German style. The schema must be complete. Whatever is not explicitly stated in the schema is not relevant in the application.

Therefore, cross-culture projects often result in a complete mismatch although the orientation to global-as-design and the deep model are commonly accepted. In oder to come to a common solution, the principles to modelling must be agreed in advance. This agreement may start with an agreement of the maxims (of quantity, quality, relevance, manner) and on the R-, I-, M-, B-principles. According to [1], the choice of the language is influenced by effectiveness (cost-effectiveness, representation effectiveness), infrastructure (especially tools), resource availability, knowledge capitalisation, and - what she calls - political factors. The latter are cultural factors.

4.4. Deep Models are Governed by Culture

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, commonsense) of modelling.

Let us consider information systems development: The grounding includes DBMS and Computer Engineering paradigms and postulates, set semantics, database theory,

---

6These observations only cover partially the specific styles and should be extended with other material as well. Since we are interested in the general culture-dependence of modelling and not in a complete empirical study we restrict ourselves.

7A similar observation has been made by M. Bjeković [1] for selection of enterprise modelling languages. She investigated the role of the purpose in modelling, the choice of modelling languages, and the factors for preferring one language above the other.
DBMS solution layering, DBMS technology (theory and culture), graphics and diagrammed canonical representation, ER canon, data-first-methods-second paradigm, and DBS guiding question. The basis can be built on specific extended ER language, on hidden basic types, on views as derived (algebra) expressions, on concept fields, on a specific extended ER thought style, on parametric generic concept field, on Indo-European utterance composition, on extended ER development methods, on extended ER heuristic rules, on transformation techniques to other deep models, and on ER tools. Typical commonsense and common practices that are applied are: global-as-design, Salami slice, functionality comes later, rigid separation into firstness syntax and secondness semantics, visualisation, well-formedness (lazy normalisation), ER pattern, experienced solutions (as exemplars), and an orientation on flat schemata. Deep adequacy uses a specific analogy, specific focus, specific purpose. Deep dependability is based on arguments from the origins, on coherence inherited, on a rigid stability, and on sufficiency on the basis of extended ER quality criteria. The deep model is extended by the four directives: (i) Perception and situation models with lexicology and lexicography (e.g. an ontology as cut-out in the concept fields); (ii) a specific communication-oriented profile; (iii) the context typical for current IT or Business Informatics; (iv) the ER community of practice. The deep model provides the interpretation and the make of the normal model.

We realise that all components of the deep model are governed by the culture of the community of practice. This culture must be accepted and is the basis for a smooth communication within this community. The culture includes the acceptance of several principles: (1) the community uses a common vocabulary (*Helsinki principle*); (2) the ‘what’ and ‘why’ of modelling is agreed (*principled universe of discourse, environment*, ...)
and information system); (3) an individual can have more than one viewpoint, one for each subject in which he is interested or has to deal with (searchlight principle); (4) all relevant general static and dynamic aspects, i.e., all rules, laws, etc., of the universe of discourse should be described in the conceptual schema (100% principle); (5) 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. (conceptualisation principle); (6) the conceptual schema for an information system in practice can be perceived as being built up like some sort of onion the inner layer of the onion being formed by the minimal conceptual schema based on the fundamentals of logic, the extensions representing the layers of the onion (onion principle); (7) development is concentrated on the what about what with paying attention to the how with what we do it (e.g. conceptual level, external level, internal level) (x-level architecture principle).

4.5. Model Matrices are Driven by Culture

According to [19], a disciplinary matrix consists of (I) symbolic generalizations as formal or readily formalisable components or laws or law schemata, (II) beliefs in particular heuristic and ontological models or analogies supplying the group with preferred or permissible analogies and metaphors, (III) values shared by the community of practice as an integral part and supporting the choice between incompatible ways of practicing their discipline, and (IV) exemplars for concrete problem solutions similar to Polya’s theory for puzzle-solving (see also Wittgenstein ‘Game’ [36]). Additionally we consider (V) a guiding question as a principal concern or scientific interest that motivates the development of a theory, and (VI) techniques as the methods an developer uses to persuade the members of the community of practice to his point of view. So, the modelling matrix includes the deep model ((I),(II),(III)) which already culture-governed and additionally.

The modelling matrix is a specific disciplinary matrix and consists of the deep model and the modelling scenarios with specific stereotypes. So it governs the development of the ‘rest’ of the model development and model utilisation. The agenda is derived from the modelling scenario and the utilisation scenarios. The modelling matrix thus provides also a specific understanding of adequacy and dependability of models.

We may now derive specific modelling matrices for information system models — mainly for the development of the normal model. The matrix is assumed to be correct for normal models. Normal modelling involves showing how systems and their models can be fitted into the elements the matrix provides. Most of this work is detail-oriented. The matrix itself is thus driven by the culture accepted by the community of practice within the given context.

5. Conclusion

The aim of this paper was to analyse the culture sensitivity of modelling and models. We see modelling as a human activity and because of that it is as culture sensitive as human behavior in general. Worldwide we use same modelling techniques and tools, which for their part unify modelling practices and models. There are also several studies that criticize the use of tools and practices developed in powerful cultures in foreign
culture context. The kernel of the criticism is that these tools and techniques transfer the elements of the origin to the culture where these are used. Big gap can be seen between Western and Eastern cultures, as well as between mature (welfare) and emergent (more poor) cultures.

In our paper we have approached the topic from the direction of culture analysis in general level and applying the results of our findings in (cross-cultural) modelling context. A modelling framework — “the model house” — is used as a basic structure. The role of normal and deep models, as well as the role of modelling matrix are parts of this framework. Culture dependent aspects in conceptual modelling provide and comparison of modelling styles of two cultures are used as applications.

Our conclusion is that we found a lot of culture sensitive aspects in modelling. Models include a lot of “wordless” information that have source in modelling languages. In this context we want to make analogy to Hall’s high and low context cultures discussed in section 1 of this paper. Modelling languages — because of the semi-formal character — leave a lot of gaps to exact specification. This gap includes always some amount of culture related aspects. In addition normal information system requirements specification includes a lot of non-functional features that are defined by still less formal language, like natural language. These features are culture sensitive and also in most cases impossible to test by normal testing practices and verification; instead of tests human validation is used - again one culture sensitive step more.

References


Models and their Foundational Framework

Bernhard Thalheim
Christian-Albrechts University at Kiel, Department of Computer Science, D-24098 Kiel

Abstract

The term model is mainly used in two meanings which are considered to be different: a model of a problem domain as a conceptualisation; a model of a set of formulas as an interpretation in which every formula within this set is true. A general theory of models has not yet been developed. H. Stachowiak proposes a phenomenal approach and ‘defines’ models by their properties of mapping, truncation and pragmatics. Meanwhile, a notion of the model has been developed. At the same time, it seems that there are rather different understandings of model in sciences and especially Mathematical Logics. Sciences treat models as reflections of origins. Mathematical logics considers models as an instantiation in which a set of statements is valid. So, mathematical model theory is often considered to be a completely different approach to modelling. We realise however that mathematical model theory is only a specific kind of modelling. We show that the treatment of models in logics and in sciences can be embedded into a more general framework. So, the theory of models is based on a separation of concern or orientation.

1 Introduction

Modelling is a topic that has implicitly been in the center of research in science and engineering since its beginnings. It has been considered as a side issue for long time. During the last 40 years it has gained more attention and becomes nowadays a subdiscipline in many disciplines. The compendium [TN15] introduces models in agriculture, archeology, arts, biology, chemistry, computer science, economics, electrotechnics, environmental sciences, farming, geosciences, historical sciences, languages, mathematics, medicine, ocean sciences, pedagogical science, philosophy, physics, political sciences, sociology, and sports. The models used in these disciplines are instruments used in certain scenarios. So, essentially it is an old subdiscipline of most natural sciences with a history of more than 2,500 years [Mul16]. It is often restricted to Mathematics and mathematical models what is however to much limiting the focus and the scope.

The modelling method is a specific science method that uses models as instruments with certain intention or goal, e.g. for solving a task. The model represents or deputes origins. The model is used instead of the origin due to its properties, esp. adequacy and dependability. The modelling method thus consists (i) of the development of ‘good’ models, (ii) of the utilisation of the model according to the goal, (iii) of the compilation of the experience gained through model utilisation according to the goal, and finally (iv) of generalisation of the experience back to the origins. So, a model must be well-build for this goal, must be enhanced by methods that support its successful deployment, and must support to draw conclusions to the world of its origins.

1.1 A Model is an Adequate and Dependable Instrument

A model is a well-formed, adequate, and dependable instrument that represents origins [Tha14, Tha17a].

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

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

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

Figure 1: The model as an instrument that is adequate and dependable for its driving directives (origins, profile (functions, purposes, goals), community of practice, context) within its background (grounding, basis) and that properly functions in utilisation scenarios as a deputy of its origins

Figure 1 represents a model of the model. The development and utilisation methods form the enabling aspects of the modelling method. Driving directives are (1) origins to be represented by the model, (2) purposes or goals or functions of models, (3) the community of its users and developers, i.e. the community of practice, and (4) the context into which the model is embedded. Models function as instruments in application or utilisation scenario. Typical functions of models are (a) cognition, (b) explanation and demonstration, (c) indication, (d) variation and optimisation, (e) projection and construction, (f) control, (g) substitution, and (h) experimentation. A model is not built on its own. It has an undisputable grounding that has to be accepted. The basis of the model - similar to the cellar - can however be disputed. Grounding and basis form the background of a model. We observe that the background is often given only in an implicit form. The same kind of concealment can also be observed for the utilisation scenario which are implicitly given by sample and generalisable case studies for the utilisation frame.

The model is not simply an image of its origins. The mapping property [Kas03, Mah09, Mah15, Sta73] might be too restrictive for models. Instead, we use analogy. Models can also be material artifacts. A model can be a model of another model. Models might follow different structuring and behaviour than the origins. Usefulness and utility according to goals govern the selection of a model instead of quality characteristics such as validity. Finally, a model comes with its background. It cannot be properly understood and used if the background is concealed. Let us distinguish the concepts of goal, of purpose, and function in the sequel. The goal of a model is in general the association between a current state and the target state that is accepted by stakeholders or – more general – by members of a community. The purpose enhances the goal by means that allow to reach the target state, e.g. methods for model development and utilisation. The function extends the purpose by practices or – more systematically – by
scenarios in which the model is used. A typical scenario is the modelling method and its specific forms.

1.2 Models in Science and Daily Life versus Models in Mathematical Logics

Models in sciences and model theory in mathematical logic are often considered to be completely different issues [Bal16]. This point of view is correct as long there is no consolidated understanding of a notion of a model. Models in model theory are instantiations of a set formulas. This set of formulas is satisfied by a model according to a logical definition frame. The model is a structure that is defined with the same signature as the set of formulas.

So, we might come to the conclusion that there are at least three different understandings of the model. We will oppose this conclusion in the sequel. It is only true for the Fuzzy or phenomenalistic view.

Models in science typically follow the modelling methods. They may be composed of a number of models and be based on other models. A model must not be true. It should however be coherent to some extent within its discipline.

The origin in science is not limited to material origins. The origin itself can be virtual or be again a model, e.g. a mental model. So, the modelling methods may also be iteratively applied.

Models often used in daily life. One kind are metaphors or parables. The typical kind is, however, a pattern for explanation, negotiation, and communication. Models carry a meaning. It is often debated whether a fashion model or a diagram or a visualisation can be considered as a specific kind of a model.

The modelling method presented so far is associated with its origins. We might however also use models for construction of other origins or models. In this case, the model is not generalised but used as a blueprint for another artifact. So, we observe that the modelling method must be extended.

1.3 Models and their Utilisation Scenarios

Models are used in various scenarios, e.g. communication, system construction, perception, analysis, forecasting, documentation, system modernisation and optimisation, control, management, and simulation. Let us in the sequel concentrate on the first three scenarios.

The extended modelling method is embedded into a more general form of activities, i.e. scenarios. The model itself is used as an instrument in a scenario or a bundle of scenarios which we call usage spectrum. It has a function or a number of functions in these scenarios. This functioning must be effectively supported by utilisation methods and is used by members of a community of practice in most cases. For instance, models of situations/states/data are often used for structuring, description, prescription, hypothetic investigation, and analysis. So, we observe that the function (or simpler the purpose or the goal) of the model is determined by the concrete way how a model is used.

A model might be oriented towards this community of practice. It can however also represent the scenarios themselves. It might represent the context of these scenarios, e.g. the scientific or engineering background, the relation to time and space, the application area insight, and the knowledge accepted by the community. It might also be oriented to representation of either a situation and state under consideration or a evolutionary change process.

The different orientations is the basis to distinguish the six concerns for models: community of practice, background/knowledge/context, application scenario and stories of model utilisation with their specific frames, situation/state/data, dynamics/evolution/change/operations, and models as representations and instruments.

Figure 2 shows the relation between the concerns and the functions a model might have2.

1.4 The Storyline of the Paper

A general theory of models, of modelling activities and of systematic modelling has not yet been developed although modelling has already attracted a large body of knowledge and research3. The notion of the model is not yet commonly accepted. Instead we know a large variety of rather different notions. Model development activities have been a concern in engineering. The process of model development has not yet attracted a lot of research. Model deployment also needs a deeper investigation. The model is mainly used as an instrument in certain application scenarios and must thus function in these scenarios. So, a model is a medium.

2Modified and revised from [Tha17c].

3It is not our purpose to develop a bibliography of model research. Instead we refer to bibliographies in [TN15] and the more than 5,000 entries in R. Müller’s website, e.g. [Mül16].
We have already introduced the general notion of a model as a starting point. The next step could be the development of a general theory of modelling. It is often claimed that modelling is rather different in science and engineering. So, we might conclude that there is no general theory of modelling. This paper is going to show that there is a general theory of modelling. We start with a case study in Section 2. These lessons gained in this case study are a starting point for a general theory of models, of modelling activities, and of systematic modelling. In Section 3 first elements of this theory are developed.

2 Models in Everyday Life and Sciences: A Case Study

Analysing model notions we realise that there are at least four different approaches:

1. The general phenomenalistic definition uses properties such as mapping, truncation and pragmatic properties for the association between origins and models. Most research on models starts with this approach.

2. The axiomatic definition follows frames used in Mathematical Logics and defines models as exemplifications of formal systems and formal theories. Models thus depute and represent a certain part of reality.

3. The mapping-based definition is based on a direct homomorphic mapping between origin and model. We might have another mapping between model and implemented system that is a realisation of the model.

4. The construction-oriented definition defines a model as being a result of a modelling process by some community of practice.

There is a fifth approach to models which simply uses artifacts as models without any definition, e.g. in human communication and also in sciences\(^4\). The definition given above follows, however, the mathematical way of defining things through definitional extensions.

Models are used as (a) perception models reflecting someone’s understanding, (b) mental models that combine various perception models and that make use of cognitive structures and operations in common use, (c) domain-situation models representing a commonly accepted understanding of a state of affairs within some application

\(^4\)One of the prominent definition is given by John von Neumann [vN55]: “The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work - that is correctly to describe phenomena from a reasonably wide area. Furthermore, it must satisfy certain esthetic criteria - that is, in relation to how much it describes, it must be rather simple. I think it is worthwhile insisting on these vague terms - for instance, on the use of the word rather. One cannot tell exactly how “simple” simple is. Some of these theories that we have adopted, some of the models with which we are very happy and of which we are proud would probably not impress someone exposed to them for the first time as being particularly simple.”
domain, (d) experimentation models that guide experimentation, (e) formal model based on some kind of formalism, (f) mathematical models that are expressed in some mathematical language and based on some mathematical methods, (g) conceptual models which combine models with some concept and conception space, (h) computational models that are based on some (semi-)algorithm , (i) informative models that used to inform potential users about origins, (j) inspiration models that provide an intuitive understanding of something, (k) physical models that use some physical instrument, (l) visualisation models that provide a visualisation, (m) representation models that represent things like other models, (n) diagrammatic models that are based on some diagram language with some kind of semantics, (o) exploration models for property discovery, (p) prototype models that represent a class of similar items, (q) mould models that are used for production of artefacts, (r) heuristic models that are based on some Fuzzy, probability, plausibility etc. relationship, etc. Although this categorisation provides an entry point for a discussion of model properties, the phenomenon of being a model can be properly investigated. Each category is rather broad and combines many different aspects at the same time. We already introduced a general notion of model. In this Section we will investigate whether the general definition covers all these kind of models for science and also daily life and whether it can be supported by a holistic treatment of models.

2.1 Models in Mathematical Logics

Let us consider only one kind of logics: classical Mathematical Logic based on first-order or higher-order predicate logics. Similar observations can be drawn for other mathematical logics as well. Mathematical logic has a long tradition of model research. Model theory became its branch and has a deep theoretical foundation. The main language is the first-order predicate logic. This language is applied in a rigid form [ST08] that became a canonical form of Mathematical Logics: It uses a canonical way of associating syntactic and semantic types. Additionally, the semantic type and the syntactic type have the same signature. The expressions of syntactic types are inductively constructed starting with some basic expressions of certain construct by application of expressions of some other construct. For instance, we may start with truth values and variables. Terms and formulas are then based on these basic expressions. The context is not considered. The world of potential structures is typically not restricted. The rigidity however allowed to gain a number of good properties. For this reason, first-order predicate logics became a first-class fundament for Computer Science.

In general, a model in Mathematical Logics is defined through its relationship to a set of formulas. These formulas are valid in the model. Additionally, axioms and rules of the first order predicate logics are valid in the model since they are valid in any structure of given signature. Models are thus instantiations (or exemplifications) for a set of statements. The theory of deduction is the main basis for reasoning. Therefore, the five concerns in Figure 2 have the specific peculiarity shown in Figure 3.

Figure 3: Models in logics for investigation of situations and expressible properties: axioms and rules form the context world; admissible states are characterised by a set of formulas; models are instances of potential systems that obey the system
The special side of the approach of Mathematical Logics to modelling is the consideration of the set of all potential models together with a given instantiation. This approach is however also taken into consideration for other model kinds as we shall in the sequel.

A model might become then an exemplar or prototype for a given theory. It can represent this theory and thus allows to reason on the given theory. It can be thus a final or an initial model (see the theory of abstract data types [Rei84, Wec92]) where the first one is the best and most detailed representation of the given theory and allows to reason on all potential negative statements as well.

We notice that classically the community of practice is not considered. Also, dynamics is not an issue. There is not really defined any reasoning frame beside the calculus itself. We are free to choose Hilbert style or Gentzen style or any other derivation style for reasoning.

A specific decision within mathematical logics is the invariance of the signature, i.e. models as structures and logical languages for theory statements share the same signature. Therefore, there is a tight mapping between terms and formulas and the properties that can be stated on the model.

This specific mapping property has also been used for the phenomenal characterisation of models as structures that a based on a mapping from the origin to the model, e.g. [Bal82, Sta73, Ste66, Ste93]. We also observe that the truncation or abstraction property is a specific property of logical models.

2.2 Mathematical Models

Mathematical models are considered to be the most prominent kind of model. A mathematical representation of another ‘donor’ or origin model is based on the mathematical language. The mathematical model is used for solving of problems that have been formulated for the origin model. The association between the mathematical model and the origin model must be problem invariant. Solution faithfulness is often not given explicitly required, i.e. the solution obtained for the mathematical model must be faithful for the origin model. Mathematical modelling presumes the existence of this origin model. So, (1) it starts with an application analysis and a formulation of the problem to be considered in the application area. Next, (2) this formulation is transformed to the origin model which allows to describe the problem. (3) This origin model is then mapped to a mathematical model. (4) The fourth phase is the development of a solution of the problem within the mathematical model. (5) The solution is verified and will be validated for faithfulness within the origin model. Finally, (6) the solution is examined for its reflection in the application area. If the solution is not of the required quality then the phases are repeated. This 6-phase circular frame [GKBF13, Pol45] is a commonly accepted scenario for mathematical modelling.

We observe that the mathematical model is similar to the logical reasoning frame. Main quality requirements satisfaction of the problem solving purpose, adequacy of the mathematical model, robustness against minor changes,
and potential and capacity for problem solution. The community of practice should not influence the model properties. It may influence on the selection of various representation models. The situation and its dynamics determines the appropriateness of the mathematical language. The mathematical model is determined by some mathematical method that has shown useful in the past.

Our model notion extends the model discussion by H. Hertz [Her84, vDGOG09]. He postulates that some artefact is a model due to its analogy to origins, its dependence within an application context, its purposefulness, its correctness, its simplicity, and its potentially only implicit given background. Models have thus a validity area.

Mathematical models are specific formal models. They are based on a formalisation that can be mapped to some mathematical language. The mapping from the formal model to the mathematical model should preserve the problem, i.e. it is invariant for the problem. The mapping should additionally also allow to associate the mathematical solution to the problem with a correct or better faithful solution in the formal model and for the origins, i.e. the model is solution-faithful [BT15]. The mathematical language has not only a capacity and potential. It also restricts and biases the solution space. The calculus used for the derivation of the model is any mathematical and not restricted to logical reasoning.

2.3 Science Models

All sciences widely use models. Typical main purposes are explanation, exploration, hypothesis and theory development, and learning. Models are mediators, explainers, shortcuts, etc. We can consider models as the third dimension of sciences [BFA+16, TD16, TTF16]3. Following [Gra07], sciences may combine empirical research that mainly describes natural phenomena, theory-oriented research that develops concept worlds, computational research that simulates complex phenomena and data exploration research that unifies theory, experiment, and simulation. Models are an essential instrument in all four kinds of research. Their function, however, is different as illustrated in Figure 5 [BFA+16].

<table>
<thead>
<tr>
<th>Data exploration research:</th>
<th>Models as starting and intermediate point for hypotheses; investigation, pattern detection, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational research:</td>
<td>Models as mediator or starting point in inverse modeling, ...</td>
</tr>
<tr>
<td>Theory-oriented research:</td>
<td>Models for representation, exploration, explanation, reasoning, understanding, comprehension, learning, ...</td>
</tr>
<tr>
<td>Empirical research:</td>
<td>Models for visualization, communication, and investigation of phenomena, experiments, observations, ...</td>
</tr>
</tbody>
</table>

Figure 5: Some model functions according to the kind of scientific research

Empiric research also uses a canonical modelling mould. Beside an ad-hoc mould we might use a sophisticated one: (1) define a research question (based, for instance, on the rhetoric frame (who, what, when, where, why, in what way, by what means (Hermagoras of Temnos, also Augustine’s De Rhetorica) or W*H framework [DT15]), (2) consider threats to the research, (3) choose a research model (e.g. positivistic), (4) develop an approach how facts become theories, (5) create a generic meta-model (with some level of abstraction, with independent and dependent parameters and indicators), (6) define analysis approaches (qualitative or quantitative), (7) define the research program and agenda as a specific research process, (8) select the research method, (9) analyse the capacity and potential of quantitative data, (10) design the experiment, (11) design the case study, and (12) design the outcomes survey.

The empirical research approach often combines qualitative and quantitative approaches. The quantitative approach is often oriented on observable data whereas the qualitative approach orients towards theory, on concepts and conceptions, and on a characterisation of the situations of interest. The quantitative theories are often ‘phenotypical’ approaches contrary to the ‘genotypical’ approaches used in qualitative approaches. A typical approach is used in the collaborative research centre 1266 6. It uses additionally an investigative reasoning approach. Figure 6 shows the differences between genotypical and phenotypical models. We use a planar representation of the three dimensions: (1) the composition dimension with sources, concepts, and theories; (2) the kind dimension with qualitative and quantitative reasoning, and (3) the model dimension that allows to concentrate on certain aspects of the first dimensions depending which function, purpose and goal the model should satisfy. A typical specific

3The title of the book [CH04] has inspired this observation.

6Scales of Transformation – Human-Environmental Interaction in Prehistoric and Archaic Societies: https://www.sfb1266.uni-kiel.de/en

Studia Metodologiczne, 2019, Poznan
treatment of concepts is applied in modelling. Since models orient on certain aspects and represent also combined representations, concepts used in models are often not directly derived from concepts in the theory. Additionally, we should distinguish between quantitative, investigative, and qualitative models. The model kind in Figure 6 uses investigative reasoning and lends some elements from quantitative and qualitative theories beside the theory offering that are used for investigative reasoning. The quantitative theory should also be reflected in the qualitative theory.

![Figure 6: Models for investigative and quantitative reasoning in empirical research](image)

A qualitative theory uses a concept or conception space that represents situations of interest (may be based on some mapping $g^c$). The situation can be observed and characterised by sources (may be based on some $f^c$ mapping). Empirical research in sciences often differentiates between an investigative reasoning and quantitative reasoning. Both use phenotypical observations on proxies. Quantitative approaches aggregate and combine the source data and thus allow to reason on correlation, dependencies, time and spatial relationships. The first two reasoning approaches should be based on a commuting diagram, i.e. we assume $F^c(\sigma^c(f^c(situating)))) = g^c(situation)$ for any situation considered.

Evidence-based proxy modelling and reasoning treats models in a different way.

(a) Models represent only acceptable possibilities. Each model captures a distinct set of possibilities to which the current description refers) which are consistent with the premises and the knowledge gained so far what makes them intrinsically uncertain because they mirror only some properties they represent.

(b) Models are proxy-driven. The structure of the model corresponds to the proxies it represents.

(c) Models represent only what has been observed and not what is false in each possibility in contrast to fully explicit models (also representing what is false).

(d) The more proxies that are considered, and the richer those models are, the more accurate the world view is.

(e) Additionally, we use pragmatic reasoning schemata, e.g. $A$ causes $B$; $B$ prevents $C$; therefore, $A$ prevents $C$. The model themselves illustrate then concepts. Therefore, sources support concepts and conceptions what inverts the mapping ($G^c$ instead of $g^c$).

Let us now consider the theory-oriented research. The frame for empirical research is similar to communication frames in Subsection 2.5. We neglect inverse modelling [Men89] although it is an important approach to science and it has been reconsidered and generalised under various other names, e.g. [ASG13, Noa09, SV05, BST06, TT13]. Data science approaches have been considered in [KT17].

So, we arrive with the hexagon in Figure 7. Models function as instruments within the science. They are vehicles for investigation, for analysis, for discovery of alternatives, for prognosis, for exploration, for explanation, for intellectual absorption, for learning, for understanding, for scoped and focussed comprehension, for representation of certain aspects, for discussion with partners within their background, for quick illustration, etc. They are supported by various kinds of reasoning. It seems that this variety is rather broad. If we however orient our investigation...
on the scenarios then we discover that the model utilisation scenarios determine the function of the model. At the same time, the background with the grounding and basis strikes through. Models are biased by their foundations, by their development and utilisation methods, their communities of interest, and their context. A specific context is the school of thought [Bab03, Fle11]. The concept space determines what could the content and the scope of a model. The MMM compendium [TN15] illustrates that models, the approach for to model, and modelling share a good number of common approaches.

2.4 Conceptual Models

Conceptual models are widely used in Computer Science and more specifically in Computer Engineering. In Computer Science and Computer Engineering, one main scenario is (1) the model-based construction of systems beside (2) the explanation and exploration of an application, (3) description of structure and behaviour of systems, and the (4) prognosis of system properties. Model-based construction might include conceptualisation. The application scenarios mainly follows the description-prescription frame. The model is used as a description of its origin and as a prescription of the system to be constructed. The notion of conceptual model is not commonly agreed however7. In a nutshell, a conceptual model is an language-determined enhancement of a model by concepts from a concept(ion) space.

The conceptual modelling method uses a canonical style of model development and utilisation. Models are instruments in perception and utilisation scenarios. They function is explicitly defined, e.g. models for design and synthesis. The scenario can incorporate a decision point that stops after understanding the perception and domain-situation models or that designs and synthesises the conceptual model after a preparation phase. The last stage support then evaluation and acceptance of the model.

So, Figure 8 displays the more specific way of conceptual modelling for information systems. The IS community with its actors \{a\} shares an IT orientation. It might however be in conflict with the business users. They reason in a different way and are often using a local-as-viewpoint approach. The global-as-design approach might not provide an appropriate support. The model development and utilisation becomes canonical after the choice of the enabling language and the modelling method. The origin models such as the perception and domain-situation models follow the style accepted in these communities. The global-as-design approach must then provide appropriate aggregations and derivations for support of local viewpoints. The community also shares the assumption of strict

7We know almost threescore different notions what shows the wider controversy about this notion[Tha18a]. E.g., Wikiquote (see [Wik17]) lists almost 40 notions. 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)
separation of specification into syntax and semantics with the firstness paradigm [KL13, Pei98] for structures and the secondness [Cas55] of functions and views. The model to be developed inherits all the paradigms, assumptions, biases, conceptualisations, cultures, background theories, etc.

A typical example for conceptual modelling is entity-relationship modelling. [Tha18b] observed a large number of paradigms, postulates, specific modelling cultures, commonsense, practices, and assumptions such as global-as-design (with derivation of local viewpoints), Salami slice typing (for homogenisation of object structure within a class), set semantics (instead of multi-set semantics that is used for implementation), uniqueness of names within a schema, hidden implementation assumptions, specific styles for model composition one must follow, well-formedness conditions, etc. Some approaches add also requirements such as strict binarisation of all relationship types.

The notion of conceptualisation, conceptual models, and concepts are far older than considered in Computer Science. The earliest contribution to models and their conceptualisations we are aware of is pre-socratic philosophy [Leb14].

2.5 Models for Communication and Human Interaction

Human communication heavily uses models. They are often not called models. Some models might be metaphors or prototypes. Other models might be incomplete or not really coherent or consistent. They are however used for exchange of opinions among users. Models function in communication scenario as a medium. The communication itself determines the role and thus the function and therefore the purpose of the model. Models represent in this case a common understanding of the communication partners. They are biased by these partners. Communication is based on some common understanding about the topic that is under consideration. Partner have already agreed on some background. They use this agreement within their communication. This agreement is based a common reflection and some common model. This model is taken for granted and not further discussed in communication. So, partners agree on some background or deep model. Typically, deep models [KT17, Tha17b] are not explicitly communicated. We need however an understanding of a theory of deep model and return to it in the next Section. The model is used for a shared understanding, for sense making, for reflection, for derivation of open issues, and for negotiation.

The hexagon in Figure 9 shows the differences between models in Mathematical Logics or sciences and communication model. The main difference is the explicit community dependence of such models. Each of the partners or agents a has some understanding of the world. This understanding is the main ingredient of a personal model that we call below perception model. The perception model also reflects the setting of the agent, especially the orientation and the priming. The communication might also be based on some common understanding, i.e. on a situation model. The situation model represents the common world view, shared knowledge and beliefs, and shared opinion.
The modelling methods is governed by communication and human interaction. So, we might base the frame on the dialogue and interaction frame. Models play a different role. They are used for common understanding. Typical specific models for human interaction are metaphors [Lak87].

Our second case shows the differences and also commonalities between Mathematical Logics and human interaction. The model must suffice all hidden agreements within the community of practice, the context, and the specific scope and focus taken by the agents. Therefore, the logics becomes now more advanced. Mathematical Logic as the opposite is oriented on general laws and thus not oriented on one model but rather on a family of models.

### 2.6 Lessons Learned with the Case Studies

We may now summarise the experience we gained:

- We realise by these case studies that there exists a common framework to models, to the activities of modelling and to modelling as a systematics reflection, for development of models, and for utilisation of models.
- Models are used to represent certain issues. They are more focused and must serve its purpose. The purpose and the focus determine which kind of adequacy is appropriate.
- Models do not exist on their own. They represent something in the world. The world under consideration depends again on the modelling frame. In most cases, mental models and perception or situation models are the origins which are reflected by the model.
- The justification must be given in a way that can be accepted by its community of practice. Models are developed by some members of this community and are utilised by some – may be other – members of this community of practice. So, models must be satisfying. Therefore, we need an explicit understanding of the sufficiency and thus quality of the given model.
- Models are composed of models that reflect their background and of models that represent specific states and situations within from one side and specific dynamics.
- Models are used as instruments in certain scenarios. They have a number of specific functions in these scenarios.
- Models are typically multi-models, i.e. an association of models which are reflecting specific sides of the same issue depending on the viewpoint that is actually considered. Since such models must be coherent we may bundle them within a model suites [DT10, Tha10].
• Model development and model utilisation typically follow canonical stories. An example is mathematical modelling that consists of a six-step procedure. Similar procedures can be observed for most sciences that start with a research question, initialise a certain research agenda or problem-solving program or schedule, adapt elements to be used to this program, and then solve a problem. Solution-faithfulness is assumed as a hidden quality characteristic beyond the problem invariance. Modelling is typically based on some specific method or methodology, e.g., the mathematical method. These methods are a mould for the modelling process itself, e.g., a pattern, template, stereotype, work-holding attachment, and an appliance. The method itself follows a macro-model.

• Modelling is still a big challenge to science and has a lot of lacunas. The biggest lacunas seem to be the missing support for combined model-based reasoning. Conceptual modelling uses a specific kind of layered model-based reasoning with changing reasoning methods depending on the stage of model development and model utilisation, e.g., in greenfield development of conceptual models: settlement of the context and the method, transfer of mentalistic concepts to codified ones with a concept expression language, transfer of domain-situation models to raw conceptual models, language-backed negotiation and agreement on a number of conceptual models that allow reflection of different viewpoints, maturation of these conceptual models, and proper documentation. The reasoning method changes according to the stages. The integration of all these reasoning methods into a holistic one is not required.

3 Towards a General Theory of Models

3.1 Deep Models and the Modelling Matrix

The context and methodology layer determines the set-up of the model. It is often taken for granted and as given. It makes modelling more economical and also more reliable. A number of quality characteristics can be thus satisfied without any further consideration. Model development is typically based on an explicit and rather quick description of the ‘surface’ or normal model and on the mostly unconditional acceptance of this set-up. In reality, this setting becomes an essential part of the model. We call it deep model [Tha18b]. It directs the modelling process and the surface or normal model. Modelling itself is often understood as development and design of the normal model. This approach has the advantage that the deep model can be used as a basis for many models.

The set-up is the modelling matrix behind the model. It consists of the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense) of modelling. It uses a collection of undisputable elements of the background as grounding and additionally a disputable and adjustable basis which is commonly accepted in the given context by the community of practice.

The modelling matrix is often given as a stereotype one should follow while developing the normal model. Adequacy and dependability of is partially already defined by such stereotypes. The stereotype of a modelling process is based on a general modelling situation.

Stereotypes determine the model kind, the background and way of modelling activities. They persuade the activities of modelling.

3.2 The Five Concerns and a General Approach to Modelling

The case studies led us to the conclusion that there is a common three-layer setting in modelling:

(1) **Community and scenario setting:** The community governs the function that a model has to serve according to their issues and scenario.

**Community of practice and application cases:** The community of practice has its needs and desires. It faces a number of application cases. The application case consists of tasks that should be accomplished. These tasks form the community portfolio. The application cases can be solved by a model, i.e., the model functions as an instrument. Community members determine which model functions best. The community agrees on the issues for modelling.
(2) **Guiding settings:** The deep model and the matrix is commonly agreed according to the setting in the first layer.

**Context:** Modelling has its implicit and sometimes also its explicit context. Knowledge and disciplinary schools of thought and understanding are considered to be fixed. In a similar form, the background is fixed. This context forms the deep model that underpins the entire modelling process. A typical element of the deep model is the school of thought.

**Modelling methodology and application mould:** Modelling follows typically practices that are accepted within the community of practice. These practices are often stereotyped. The methods that are used for model development

(3) **Origins and targets:** Members of the community form their personal perception models and share their domain-situation model that characterises states and dynamics in the application domain that is of interest. These models are the origins on which the normal model is formed as an extension of the deep model.

The final result is a model that combines the normal and the deep models. The representation of the final model must not show all details of the deep model.

![Diagram](image)

Figure 10: The five concerns for models as a kernel for a theory of models and of modelling

This general setting takes us back to the rhetorical frame and its generalisation to the W*H specification framework [DT15]: In our case, the model (“what”) incorporates the meaning of parties (semantical space; “who”) during a discourse (‘when’) within some application with some purpose (“why”) based on some modelling language.

We thus distinguish between five grounding and driving perspectives to models:

**Community perspective:** The community has intentionally set-up its application cases, its interests, its desires and its portfolio. The community communicates, knows languages, explains, recognizes, accept the grounding behind the models, has been introduced to the basis and is common with it. Models are used by, developed by and for, and gain a surplus value for a community of practice. They may have a different shape, form, and value for community members. They must, however, be acceptable for its community. Typical specialisations of this concern are ‘by whom’, ‘to whom’, ‘whichever’, and ‘worthiness’.

**Purpose, function, goal perspective:** Models and model development serve a certain purpose in some utilisation scenarios. The model has to function in these scenarios and should thus be of certain quality. At the same time it is embedded into the context and is acceptable by a community of practice with its rules and understandings.

We answer ‘why’ and ‘for which reason’ questions.

---

*It relates back to Hermagoras of Temnos or Cicero more than 2000 years ago., i.e. they are characterised through “who says what, when, where, why, in what way, by what means” (Quis, quid, quando, ubi, cur, quem ad modum, quibus adminiculis).*
**Product perspective:** Models are products that are requested, have been developed, are delivered according to the first perspective, are potentially applicable within the scenarios, and have their merits and problems. Typical purpose characteristics are answers to ‘how-to-use’, ‘why’, ‘whereto’, ‘when’, for which reason’ and ‘wherewith’ (carrier, e.g., language) questions.

**Engineering perspective:** Models are mastered within an engineering process based on some approaches to modelling activities and to utilisation of models. Modelling is a systematically performed process that uses methods, techniques, preparations, and experience already gained in former modelling processes. The modelling method is typically given in a canonical form. It guides and steers the model development and the model utilisation processes. This guidance can be derived from the scenarios in which the model functions.

**Background and context perspective:** Model development and utilisation is a systematic, well-founded process that allows one to reason on the capacity and potential of the model, to handle adequacy and dependability of models in a proper way, and the reason on the model and its origins that it represents. A modelling culture also answers the by-what-means question beside providing the background. The background is typically considered to be given and not explicitly explained. It consists of an undisputable grounding and of a disputable and adjustable basis. The context clarifies on which basis and especially on which grounding the model has been developed and must be restricted in its utilisation. Additional context characteristics are answers to questions about the ‘whereat’, ‘whereabout’, ‘whither’, and ‘when’.

### 3.3 Model-Based Reasoning

The observation depicted in Figure 6 drives us to a multi-model approach. We build models in situations, concepts and theories in dependence on their function and purpose. The same situation-concept-theory may be the basis for a variety of models. A typical multi-model approach is the consideration of models in Physics. Models should

![Figure 11: Models as specific representations of situations, concept(ion)s, and theories](image-url)

thus be considered to be the third dimension of science [BFA+16, TN15, TTF16]. Disciplines and also human communication, human interaction, and human collaboration have developed a different understanding of the notion of model, of the function of models in scientific research and communication. Models are often considered to be artifacts. In reality, they are however instruments that are used with a certain intention. Models might also be perception models that incorporate mentalistic concepts [Jac04]. Models are used in various utilisation scenarios such as construction of systems, verification, optimization, explanation, and documentation. In these scenarios they function as instruments and thus satisfy a number of properties.

Model-based reasoning [Bre10, Mag14] is enhances classical reasoning such as reasoning mathematical calculi or logical derivation. There are several kinds of reasoning that are more appropriate and widely used:

**Evidence-based modelling and reasoning** is one of the main approaches for quantitative models. Models only represent acceptable possibilities. Each model captures a distinct set of possibilities to which the current description refers. Possibilities are consistent with the premises and the knowledge gained so far what makes
them intrinsically uncertain because they mirror only some properties they represent. In investigative and quantitative modelling, models can be proxy-driven where the the structure of the model corresponds to the proxies it represents. They might also include abstractions such as negation which must be then stratified. Propositional evidence-based reasoning is based on monotone functions and specific interpretations for logical connectives. Models represent in this case only what has been observed and not what is false in each possibility what is different from fully explicit models which represent what is false. The more proxies are considered the richer those models are, the more accurate the world view is. Evidence-based modelling and reasoning uses pragmatic reasoning schemata, e.g. A causes B; B prevents C; therefore, A prevents C. The calculus may use several implication forms, e.g. deterministic conclusions (A cause B to occur: given A then B occurs) and ordered sets of possibilities (A enables B to occur: given A then it is possible for B to occur).

**Hypothetical and investigative modelling** considers different assumptions in order to see what follows from them, i.e. reasons about alternative possible worlds (i.e. states of the world), regardless of their resemblance to the actual world. Potential assumptions with their possible world conclusions and assertions are supported by a number of hypotheses (allowing to derive them). It is often combined with abductive reasoning. Evidence against hypothesis is performed by testing its logical consequences, i.e. exploring different alternative solutions in parallel to determine which approach or series of steps best solves a particular problem.

**Causal reasoning and modelling** is a specific variant of inductive reasoning and justification-backed truth maintenance with assertions (beliefs, background) and justifications within some context (current beliefs, justifications, arguments). It establishes the presence of causal relationships among events based on methods of agreement, difference, concomitant variation, and residues. It uses assumptions and thus avoids inconsistent sets (‘nogood’ environment). The environment consists of a set of assumptions, premises, assumed statements, and derived statements for the world view. Justifications (e.g. data-supported) represent cause. Hypotheses are not derived from evidence but are added to evidence. They direct the search for evidence. They are tested by modus tollens \((H \rightarrow I) \land \neg I \Rightarrow \neg H\).

**Network reasoning** uses models that are expressed as networks. Nodes carry justification (arguments) and status (in, out, believed, relevant, necessary, ...). Edges, hyperedges, or directed edges have an antecedent (support nodes) and conclusions. They may also be non-monotonic and enable backtracking for dependencies (causality, chronological, space, etc.). Labels also express the degree of consistency and believability. Queries can be expressed as subgraphs and are evaluated by query embedding into the network.

Model-based reasoning is an interactive and iterative process that helps to digest a theory and to develop the theory. Therefore, model-based reasoning integrates many reasoning approaches, e.g. deduction, induction, abduction, Solomonoff induction, non-monotonic reasoning, and restrospective reasoning. Model refinement might also be based on inverse modelling approaches. Facets of the last one are inductive learning, data mining, data analysis, generic modelling, universal applications, systematic modelling, and pattern-based reasoning.

### 3.4 Towards Powerful Methodological Moulds

The hexagon picture and the consideration of the variety of different (reasoning) techniques might lead to the impression that a general treatment of models and a methodological support is infeasible. Sciences and humans have however developed their specific approaches and overcome the challenges of this complexity. We will illustrate resolution of complexity by two methods: Layered treatment and generic modelling. Both approaches are based on the separation of a model into a deep or core model and a normal model. A typical example of a methodology is the mathematical modelling method [BT15, GKBF13, vDGOG09, Pol45] (see Subsection 2.2). The CRISP cycle (data selection according to generic model, data preprocessing, data transformation, data mining, model development, interpretation/evaluation) [BBHK10] and classical investigation cycles (define issues and functions of the model, hypothetically predict model properties, experiment, (re)define model, apply and validate the model against the situation) are typical methodologies. Similar methodologies are known for data mining [Jan17], data analysis [BBHK10], and systematic mathematical problem solving [Pod01]. They use a variety of reasoning techniques and layer their application of these techniques according to the stage that is currently under consideration. These modelling methods and methodologies are used similar to moulds that are commonly used in manufacturing.
Data mining [Jan17], inverse modelling [RSS+10], and generic modelling [TTFZ14] start with a generic model. A set of associated models (called model suite) is the result of a modelling process. We may develop a singleton model or a model suite. Figure 12 displays a variant that starts with an initialisation and setting of the modelling process. The initialisation is based on the issues that are important for the community of practice, the tasks that are on the agenda, and the injection of the context. The community of practice aims at completion of tasks from its portfolio and is bound by profiles of their members what also includes beliefs and desires shared in this community. At the same time, the methodology for modelling is already chosen. That means, the upper dimensions in Figure 10 governs the entire modelling process. A similar approach can be declared for model redevelopment model evolution instead of model development from scratch (greenfield modelling). The result of the first layer is a deep model and a matrix.

The second layer or stage uses some kind of most general and refinable model as the initial model. A generic model [BST06, TF16] is a general model which can be used for the function within a given utilisation scenario and which is not optimally adapted to some specific origin collection. It is tailored in next steps to suit the particular purpose and function. It generally represents many origins under interest, provides means to establish adequacy and dependability of the model, and establishes focus and scope of the model. Modelling is often based on some experience. This experience can be systematically collected within a number of libraries. Libraries and collections are used for collecting the most appropriate setting and model. This selection is controlled or governed by rules, restrictions, conditions, and properties. The main results of the second layer are generic models and an agenda for the next modelling steps.

The third layer sets the environment for the development of the normal model. This environment prepares model development on the basis of the generic models and under inclusion of the deep model. The section of methods might also include the selection of parts and pieces from the context, e.g. from the background and especially from theories and knowledge. The fourth layer results then in the development of a normal model that can be neatly combined with the deep model. Representation models are developed for different members of the community of practice and for different functions the model must fulfill in the utilisation scenario.

This development process is often cut down to the fourth layer assuming the results of the first, second, and third
layer as already given. This kind of implicitness has often been assumed for language utterance. The government and binding approach [Cho82, BST06] made the two-step generation of sentences explicit: we intentionally prepare the deep model and then express ourselves by an explicit statement which is build similar to a combination of a normal model and of a cutout of the deep model.

4 Conclusion

A collection of modelling approaches has been presented in [TN15]. It seems that the variety of modelling approaches, the different utilisation of model, the broad span of underpinning theories, the variety of models themselves do not allow to develop a common setting for models. We often met the claim that models used in social and natural sciences, in mathematics, in logics and in daily life are so different that a common treatment cannot exist. From the first side, logicians provided a specific understanding of models that is easy and formally to handle. They inspired model research and the notion of model, e.g. [Bal16, Kas03, Mah09, Mah15, Sta73, Ste66, Ste93]. This notion has mainly been based on properties that a model should satisfy: mapping, truncation, and pragmatic properties as phenomenalistic characterisation of the notion. From the second side, models in all sciences have been used as an artifact for solution of problems, e.g. [BT15, Her84, vDGOG09, vN55]. The model notion has been enhanced by amplification, distortion, idealisation, carrier, added value, and purpose-preservation properties. From the third side, language- and concept-based foundations of models have been developed in philosophy of science and linguistics [Bifrm[−]−5, Bur15, Cas55, KL13, Lat15, Pei98]. From the fourth side, models in engineering [BFA16, LH15, TD16, TTF16] are instruments for system construction. From the sixth side, models are also instruments in human interaction. They are used as metaphors, for communication, for brief reference, for depiction, as prototype, etc. For instance, the question whether a picture or a photo is a model depends on their utilisation in some interaction scenarios. We thus may conclude that a common science and culture of modelling cannot exist.

The main claim in this paper is however that a common treatment of models in science and human interaction can be developed. We base our foundational framework on a separation of concern. This separation into five governors for models provides a common treatment of models and model utilisation. We base our framework on the observation that not all concerns are considered at the same time. So, we can use some kind of stepwise procedure for model development.

Utilisation of models as instruments in scenarios is the main driving property that distinguishes something from a model. The model functions in scenarios such as communication, reflection, understanding, negotiation, explanation, exploration, learning, introspection, theory development, documentation, illustration, analysis, construction, description, and prescription. How the model functions has been illustrated in the case of model-based reasoning. Model-based reasoning goes far beyond model methods used in classical first-order predicate logics or mathematics. We use the layering approach also for model methods since the development of a general reasoning method is far beyond the horizon.

The meta-models of modelling concerns in Figures 3, 4, 7, 8, 9, 10 support the layered modelling method in Figure 12. Instead, we could separate the layers into communities and their application scenario, into background and methodology setting, into situation and theory setting, into origin calibration, and model delivery layers.

This paper has been centred around models, theories, communities, context, methodologies, state, and dynamics at the same level of abstraction. Model-driven development and architecture [MMR17, SV05] is an orthogonal approach to this paper. It distinguishes abstraction layers for models (M1), model frames (M2) [as meta-models], model frameworks (M3) [as meta-meta-models], and model framework setting (M4). The data/information and traces/events abstraction layer (M0) underpins models. Our approach has been mainly oriented on M1. We envision that the general M0-M1-M2-M3-M4 architecture can be integrated into our approach as well.

References


Abstract. Policy development is a complex and highly dimensional process. This complexity is very difficult to comprehend due to complexity of the parameter space, multi-dependence of parameters, and the nature of process. Therefore, policy makers should be supported while considering and evaluating various alternative decisions. This paper illustrates a modeling approach for advisory and assistance in decision making for political practitioners. We describe the corresponding advisory tool supporting the interactive decision process.

Keywords: Computer-based communication tool, interactive learning between scientific models and practitioners, political decision making support

1 Introduction

Policy decision making is a complex task which comprises the understanding of possible positive or negative consequences of decisions as well as a mechanism to restore consistency of a system in the case of inappropriate decisions. Thus, even policy experts often have only a vague understanding of how policies impact on relevant outcomes. Therefore, political practitioners use simple mental models (beliefs) to understand complex impacts of policies. For this reason, a technical solution for the simulation of policy impacts can be helpful, e.g., a graph displaying the impact of parameters. Our software will work as a digital playground system with relevant decision parameters as inputs and implied outcomes (consequences of the decision) as outputs.

Nowadays, it is commonly accepted that good economic policy has to be evidence-based, i.e., rest on scientific knowledge and statistically proven evidence. However, scientific modeling is often criticized by political practitioners as a purely academic exercise that fails to provide practical tools for understanding or designing optimal real-life economic processes [5]. Accordingly, scholars promote participatory policy analysis that is characterized by an interaction between economic theory and political practice to combine the ‘objective’ knowledge derived from economic theories and empirical data with the ‘subjective’ knowledge of stakeholder organizations as political practitioners ([2], [9], [5]). Moreover, inadequate communication between scientific policy analysts and political actors is proposed to be a principal cause of the limited impact of research on policymaking. For example, the ‘utilization of knowledge school’ emphasizes the fact that policy analysts and policymakers live in two separate communities [5]. Hence, to become more efficient, the relationship between scientific experts and policy actors must be redefined.

Moreover, Stiglitz argues in his highly recognized book “Whither socialism?” [14] that the market-socialist experiences in Eastern Europe failed due to the incorrect beliefs of politicians in the Arrow-Debreu concept of real market economies as a complete set of competitive markets ([14], Chapter 11). Interestingly, Stiglitz’s explanation of the failure of the market socialism experiment highlights an interesting general point: economics must be recast as something more than a constrained maximization problem to understand and design real economies. In other words, theoretical models provide a relevant benchmark for understanding real-life economic processes but require abstract scientific models and political praxis to actually change the world. Hence, as previously discussed in [6], [12], [7], identifying effective solutions for central economic problems appears to be a problem of linking abstract economic theory with feasible political practice. Accordingly, scholars of participatory policy analysis discussed innovative tools, such as participative modeling (see [5]) (i.e., improving communication in formal models by means of interactive or man-machine simulations [for example, see [1]] or decision seminars [10]).

Beyond interesting methodological ideas and concepts for assessing the role of relevant ‘objective’ scientific knowledge it is important to better understand
and design the complex communication processes between science and political practitioners in a way that combines the knowledge of both worlds to generate advanced solutions to existing economic problems, such as the transformation to a sustainable bio-economy or reaching sustainable development goals.

In this context the paper develops a computer-based tool Policy-Lab that facilitates an interactive communication and learning between political practitioners and scientific models. Figure 1 shows a graphical presentation of positioning of scientific models’ statements (scientific world), political practitioners’ beliefs (stakeholder beliefs world) and the aspired communication between these two worlds.

![Figure 1](image-url)

### 2 Policy-Lab tool

The Policy-Lab tool has to fulfill various tasks in order to effectively support the decision-making process and facilitate the learning of stakeholders. Those tasks can be categorized as follows:

- **Input Device**: Survey policy preferences, goals and beliefs using questionnaires.
- **Report Device**: Report surveyed data back to the group. This requires the dynamic application of statistical analysis of the data.
- **Interactive Modelling Device**: Users can simulate different policies and evaluate their impact on policy goals.
- **Consensus Device**: This device provides support in finding a potential political compromise.

An integral part of those devices is the simulation of scientific models. For example, typical formulas used in such simulation that political practitioners should understand in order to make a decision in the economy area look like the following one:

$$ y = [\gamma], i \in I = \{1, ..., n_i\}; s \in S = \{1, ..., n_s\}; j \in J = \{1, ..., n_j\}; t \in T = \{0, ..., 9\} $$

$$ Be_s(\gamma) = \eta s \left( \sum_{i \in I} \mu_i y_i^{-p} \right)^{-\frac{1}{p}} $$

$$ tp_s(y) = tp_s^{max} e^{\frac{a_p y_p(y) + b_p}{e^{a_p y_p(y) + b_p} + 1}} $$

$$ wZ_j(y) = s_j^{const} \sum_{s \in S} tp_s(y) \cdot \xi_{s,j} $$

$$ Z_{ij}(y) = Z_{j,0} \cdot (1 + t \cdot wZ_j(y)) $$

In this case developing a decision is rather difficult and some supporting technical solutions are necessary.

To support the simulation of these models technical methods and frameworks are used. The methods being used during the simulation are mathematical statistical methods (Bayesian model averaging, Meta Modelling) and programming languages for statistical computing (R) and optimization problems (GAMS).

In order to make models accessible to a wide range of users, who in this case are organizations or individuals, who are interested in the construction of economic policies (in our case these are agricultural policies [8]), an intuitive visualization is required. The visualization part of the tool should work as a playground for model simulation supporting expert learning, model learning, interactive learning (expert-model-expert exchange), and learning from collective decision (voting over policies or exchange games).

Thus, the Policy-Lab tool should work as an interactive input-output playground for the models’ simulation and graphical visualization.

At the same time, the tool should process a large amount of model specific data: different kinds of input-output parameters and computational cores of the models. So an important issue during the tool development is the implementation of a suitable database structure.

The Policy-Lab tool will be implemented in the form of a web application. The tool is now in the creation phase, for this reason the main concepts of tool development, tool requirements, and its structure will be discussed further.

### 2.1 Theoretical concepts

Some theoretical concepts will be explained before the structure of the playground is going to be introduced.

1) What is a model from the tool’s perspective?

In the sense of the current tool, a model is a computable unit with defined input parameters, computational core, and computed output parameters, which can be shown in a graphical form. A special sub-type of a model is a questionnaire, that has input parameters and computational core, which adds user input to a statistical model and recalculates its output. The output of recalculations is not shown to the users directly, but can be called from another view.

2) How model data will look like?
The computational core of a model is predefined by the model scientists. It can be written in R, GAMS or in other programming language. The input and output parameters depending on the language used are language specific character values, which can be saved in a database or in an external file. These parameters should be accessible to the playground.

2.2 Playground system requirements

The creation of the simulation tool begins with the comprehension of required features. Partly this information can be derived from the existing Policy-Lab tool prototype, partly from model scientists’ requirements and user expectations.

The list of requirements for the simulation tool includes the following:

- clear and comprehendible software structure
- clear and comprehendible database structure
- scalability of the system
- maintainability of the system
- efficiency of the system
- run-time reciprocative input-output system
- user-friendliness of the system

Based on the analysis of system requirements the following issues can be defined during the development of the tool:

- How to implement interactive forms for user-input and output? Which interfaces are needed?
- How input and output parameters for the models look like and how they are saved?
- How the communication between the computational module and the web interface looks like?

2.3 Playground system structure

The simulation tool should serve as a web information system for model simulations, with interactive input-output mechanisms for users. The system should have a clear structured database, expandable for new entities, since the system will describe a varying amount of models. The system should visualize a list of models and its descriptions for users. Further the system should have views for input parameters from users and possibilities for the graphical presentation of computed output. Another integral part of the system is a computational module, where the computation of output takes place.

According to the system requirements the new system should have the following components:

- Web interface for users with possible use-cases’ definition, user management functions, presentation of views related to models, including model-input-parameters and output graphics.
- Computational module with possible integration of R and GAMS sub-modules.
- Communicational interface: beside other functions web interface and computational module should be capable of interaction with each other.
- Database for the web interface
- Database for the computational module

Web interface

Web interface is a unit that contains common login, logout, and register functions, explanatory use-cases, overview of present models, view for input parameters for the models, view for the output in graphical form. Moreover, there should be a separate view for administrators to allow user management.

Computational module

Computational module is a unit that can be connected to R or GAMS sub-modules or use some other language for computation. This module should communicate with the web interface: parse user-input-parameters, convert them to input-parameters in the format of computational language depending on the model, parse computed output back to the chosen web interface format (e.g. JSON).

Database for the web interface

Database for the web interface should contain all the information about users and their management, widgets shown in the interface, and shown model views. Furthermore, for the presentation of input and output this database should have information about input and output parameters of a model.

Diagram 1 shows a fragment of a possible ER-schema for the database:

![Diagram 1](image_url)
includes descriptions of models, their simulations and different types of simulation result parameters. Additionally, every interface page has specific widgets of different types depending on model being simulated, including charts and questionnaires.

**Database for the computational module**

In the case computation is produced in another application it needs its separate database. The database for the computation should have information about models, their computational cores, and their input-output parameters.

If the computation module does not need its own database, analogical database entities are necessary. A possible ER-schema of a computational module is shown in Diagram 2:

![Diagram 2](image)

**Communication between web interface and computational module**

Communication between these two modules is an important part of the system, the whole software structure and efficiency depends on the form of communication.

Two architectural alternatives for modules’ communication have been developed:

1) Web interface and computational modules can be placed inside of one software project, so that the division in interface and computation is only a logical notion. In this case the interface and computational parameters can be saved in the same database. The computation itself can be made, for example, with JavaScript language. In the case of JavaScript, the computation will proceed efficiently as no integration of external R and GAMS modules is needed. The communication in this case is trivial and proceeds within one application.

2) In the other case, R and GAMS modules can be stored in a separate application, if the computation needs these modules because of its complexity, as it allows to bring a modular structure to the software. In addition, the exchange of or changes in R or GAMS models are made easier, because they do not influence the execution of the web interface in a negative way. Thus, the two components are not only logically, but also physically separated from each other. The communication between prototype tool and the application where model computation takes place proceeds with HTTP-messages, containing input-output parameters for computation and information about models in JSON format.

Figure 2 illustrates, how this kind of architectural style can be implemented:

![Figure 2](image)

In the system the both ways of communication will be used, depending on the complexity of a model.

**2.4 Advantages of the system**

The described playground system has a number of advantages:

- The system is scalable and extendable, as the underlying web information system is dynamic and is built accordingly to the database contents. The expandable database allows the insertion of new visual elements and models for the simulation.
- The first architectural style for communication allows the implementation of a run-time reciprocative input-output system.
- The second architectural style for communication contributes to system’s modularity and can be approached from two different perspectives: web interface based and computation based perspective. Thus, two scientists can work simultaneously on the two components. Any changes in one of the components would not cause error or stoppage of the execution in the other component. After the adaption of communicational modules, the changes can be accepted by both components.
- The tool supports expert, model and interactive learning, moreover the learning from collective decision is implementable.
- Description of use-cases supports user-friendliness.

**3 Conclusion**

Described Policy-Lab tool facilitates political decision making by presenting an interactive playground system, that simulates a large opportunity space for policy
decisions and computes possible effects of the model simulation with the decisions made.

As a result, Policy-Lab tool for policy decision enables political practitioners to relate potential policy decisions to corresponding outcomes.

The described tool should be flexible, efficient and user-friendly, in order to be able to simulate the full complexity of the models and to assist in successful decision making.

**Related work**

There exist other systems, which work with interactive user input-output and use a large number of possible input parameters and calculations, beside the Policy-Lab tool prototype, the precursor of the current simulation tool, mentioned above.

Examples of agricultural frameworks are:


Another decision making GIS-based tool is ReSAKSS [13], it contains data on agricultural, socio-economic and bio-physical areas. This tool assists policy makers in developing agricultural policies.

Examples of other frameworks are:

Today one can find modeling tools which accept a wide range of parameters and simulate some complex process in order to understand the influence of these parameters on the system in medicine.

The Lives Saved Tool for Maternal and Child Health (LiST) [15], [11] is a modeling framework developed by the Institute for International Programs at Johns Hopkins Bloomberg School of Public Health with intention to estimate the effect of health coverage on maternal and child health. LiST models the status of health coverage under the influence of various factors (e.g. increasing of health care services and usage of nutrition interventions). In this tool users can estimate the impact of different kinds of health interventions in order to plan the strategies for the improvement of medical methods in maternal, newborn, and child health. The tool contains the data about the effect of some kinds of interventions on peoples’ health. Further, the data about maternal and newborn mortality rates, health coverage and interventions of a particular country or region, is collected. Thus, a user can simulate the usage of specific health care methods in a particular region and see the influence of this usage as graphical output.

The Multi-Criteria Analysis Decision framework is a modeling framework for decision making and priority setting, which elaborates on possibilities to create „an equitable, efficient, and sustainable health care system” [15]. All possible health interventions are ranked and compared during a multi-criterion analysis. A specific web-based framework to implement this approach was developed by the EVIDEM Collaboration [3]. The EVIDEM tool is used to provide the participants of the health care process with information and to support decision making during this process. The tool simulates different factors influencing patients’ health and produces a graphical output measuring the importance of these factors or the degree of their positive or negative impact.

**References**


