Towards a textbook on ontology-guided conceptual modeling

Henderik A. Proper[1], Bas van Gils[2,3]

[1]Business Informatics Group, TU Wien, Vienna, Austria
[2]Antwerp Management School, Antwerp, Belgium
[3]Strategy Alliance, Amersfoort, The Netherlands

Abstract

This paper provides the outline of a planned textbook on conceptual modeling. The target audience of the planned textbook are learners of conceptual modeling in general, and bachelor-level students in particular. Our goal with this paper is to seek feedback regarding the setup of the planned textbook, as well as further validate the need for such a textbook. Next to discussing the planned content, and the underlying line of reasoning, of said textbook we will address both the motivation for its development, as well as some of our fundamental underlying assumptions.

Keywords

Textbook, Conceptual Modeling, Ontology-Guided Modeling

1. Introduction

The authors of this paper (and proposed textbook) have been involved in a broad range of modeling related teaching engagements for different groups of learners. This ranges from full-time bachelor and master students to practitioners. It also involved students with diverse professional backgrounds across different programmes (e.g. business administration, information management, or computer science). Content-wise, the teaching engagements pertained to e.g. ArchiMate [1, 2], BPMN [3], DEMO [4], and e3Value [5], as well as fact-based modeling [6].

Across these teaching engagements, we see an underlying need to learn *how to model*, which we consider to be a skill in itself, independent of the modeling language/framework that is used. In line with this, and after checking the available books/literature, we think there is a need for a textbook on learning conceptual modeling in general. The phrase "in general" in the latter sentence, refers to the fact that we specifically consider conceptual modeling as having a broader role to play [7] than a database-design-only one (as we often find it being 'framed' into). The primary target audience of the planned textbook are initial learners of conceptual modeling, and bachelor-level students in particular. At the same time, we also keep in the back of our minds that professionals may also need to (re)learn this important skill.

Proceedings of the 17th International Workshop on Value Modelling and Business Ontologies (VMBO 2024), February 26–27, 2024, 's Hertogenbosch, The Netherlands
Henderik.Proper@tuwien.ac.at (H. A. Proper); Bas.vanGils@ams.ac.be (B. v. Gils)
http://www.erikproper.eu (H. A. Proper); https://strategy-alliance.com/en/team/bas-van-gils (B. v. Gils)
0000-0002-7318-2496 (H. A. Proper); 0000-0003-2434-0547 (B. v. Gils)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

In line with [8, 9], we suggest to start by teaching students to create conceptual models that capture the (core) concepts and relationships in a domain, and to then teach them to specialize such models towards more purpose/domain specific modeling languages. Some of our own research also indicates that *grounding* e.g. UML models [10, 11], System Dynamics models [12, 13], Process models [14, 15], or ArchiMate models [9, 16], on more generic conceptual models involving the core domain concepts and their relations, is helpful in better underpinning the models, as well as supporting the process of learning how to model [8].

Existing (text)books, such as [17, 6, 18, 4], do touch on some of these elements. However, they generally remain 'framed into' conceptual modeling for database design only, while also not providing an explicit procedure for the creation of conceptual models. One notable exception regarding the latter is Halpin's textbook [6] on conceptual *database design*, which does provide an explicit procedure. The modeling procedure as suggested for the textbook, will certainly build on Halpin's work.

We also posit that following such a grounding and specialization process will enable us to make it more explicit to learners what the role of their ontological commitment (and the meta-model underlying the used modeling language in particular) is in 'painting' their picture of the world when they, as modelers, create a model of a domain. As such, beyond the presently suggested textbook, we are also considering potential follow-up textbooks to specialize the general modeling skills towards specific (classes of) modeling languages, such as (business) process modeling and enterprise architecture modeling.

Our goal with this paper is to seek feedback from the VMBO community regarding the setup of this textbook, as well as further validate and discuss the need for such a textbook. Next to discussing the planned content, and the underlying line of reasoning, we will also highlight our fundamental underlying assumptions regarding modeling. In line with this, the remainder of this paper is structured as follows. Section 2 provides an exploration of some (but not all) of the underlying assumptions and foundational perspectives from which we will structure the textbook. The actual planned structure is presented in section 3.

2. Preliminaries

2.1. Defining the notion of conceptual (domain) model

Without a sound understanding of what a model is, and what the act of modeling entails, it will be difficult to teach modeling. This seems trivial, but our experience is that this is far from the case. As such, the proposed textbook will need to start with a reflection of the concepts of *domain model* and *conceptual (domain) model*. In line with [19], we understand a *domain model* to be: "*A* ***social artifact*** *that is* ***understood****, and* ***acknowledged****, by a* ***collective human agent*** *to* ***represent*** *an* ***abstraction*** *of some* ***domain*** *for a particular* ***cognitive purpose***." With *domain*, we refer to 'anything' that one can speak and/or reflect about; i.e. the domain of interest. As such, *domain* simply refers to 'that what is being modeled' (below we will return in more detail regarding to the notion of the domain that is being modeled). Furthermore, the domain could be something that already exists in the 'real world', something that is desired to exist in the future, something imagined, or even something that is brought about by the existence of the model itself. Based on this, and in line with [19], we define a *conceptual model* to be: "*A model of a* *domain, where the purpose of the model is dominated by the ambition to remain as-true-as-possible to the original domain understanding.*"

2.2. Conceptual models as linguistic artifacts

Requiring a model to be an artifact that needs to be understood by human agents, immediately puts models in the realm of language. Therefore, an important theoretical foundation of domain models is the semiotic triangle by Ogden and Richards [20], as depicted in fig. 1. The tenet of the

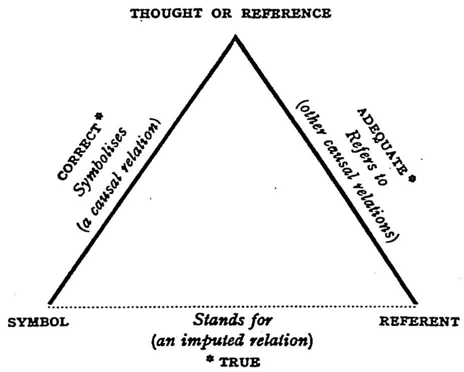

Figure 1: Ogden and Richard's semiotic triangle [20]

semiotic triangle is that when we use *symbols* (including models) to speak about 'something' (a *referent*), then these symbols represent (*symbolize*) our thoughts (*thought or reference*) about that something (*referent*). The *thought or reference* is the meaning we have assigned to the *symbols*. The *referent* can be anything, in an existing world, or in a desired/imagined world. A domain to be modeled generally involves a complex of (related) referents, which results in a complex of corresponding concepts/thoughts.

2.3. Verbalization-based approach

The link to the semiotic triangle, and the general communication-oriented stance, is also the reason for using a verbalization-based approach as the starting point to learn modeling. In doing so, we follow the so-called 'telephone heuristic', as coined by the fact-based modeling approaches [21, 22, 6], as if one has to explain to someone else what a domain is about through a telephone; or better yet, via a 'chat' session. In that context, it involves the formulation of elementary facts about the domain being modeled, while being as explicit as possible. At this point it should be noted that, as we will discuss below, there is a need to be more nuanced about the use of the term 'fact'. As such, we will actually prefer to speak about *elementary propositions*.

From a practitioner's perspective, providing such explicit verbalizations of one's understanding of a domain to be modeled may sound as a laborious task. However, we think that from a *learning* perspective, explicitly articulating one's domain understanding in terms of elementary propositions provides a good starting point. An interesting analogy in this regards is the use of swim-paddles when training to swim[1]. Such paddles enable swimmers to improve their swimming stroke. At the same time, during an actual match, swimmers would not use them.

Additionally, as also argued in [6], the example elementary facts also remain useful for e.g. validation purposes. Seeing not only the (abstract) domain model, but also example *instances* on which they are based will help stakeholders to (a) understand models better, and (b) have confidence in their correctness and usefulness.

2.4. Domain-aware information systems – Bringing models to life

In general, the 'boxes and lines' based models that we tend to produce in our domain are rather 'passive' in the sense that they just 'sit on paper'. Unlike programming, where the program that is being developed can already be run during its development. Therefore, even though we argue that it is important to make it clear that conceptual modeling should not be 'framed' into conceptual design of databases only, we do suggest to include exercises where models are translated to actual running software. Indeed, resulting in (different forms of) information systems.

However, in doing the latter, we also plan to make a distinction between (traditional) information systems where the conceptual models are only used as a design artifact, and *domain aware* information systems (DAIS), where the conceptual models are an explicit artifact in the information system. An example of such domain aware information systems are process-aware information systems which do not just 'log' the execution of processes, but pro-actively monitor/direct their execution based on the process models.

2.5. Awareness of one's ontological commitment

As mentioned before, we also aim to make learners aware of their/the-used ontological commitment (meta-model or conceptualization), in relation to the modeling purpose. In [9, 23], this thinking is 'boot strapped' by discussing how a modeler mentally 'paints a picture of the world' in terms of available kinds of elements. This process is illustrated in fig. 2. On the left, we see a situation where the viewer/modeler can only paint a picture of the world in terms of 'elements'. Having only elements to paint this picture is not useful, as one cannot express relations between the elements. On the right, the viewer/modeler can paint this picture in terms of 'entities' and 'relations'[2]. This then, allows the viewer/modeler to capture entities and their relations. This, however, does not allow one to speak about classes and types of things that may have common properties and rules. So, a next step, in the context of this illustration, would be to introduce a type-instance distinction as a special kind of relation.

This process of a step-wise refinement of the ontological commitment, is actually also used in clarifying the anatomy [16] of the ArchiMate language as illustrated in fig. 3.

[1] https://www.yourswimlog.com/ultimate-guide-swim-paddles/
[2] In retrospect, erroneously referred to as relation*ships* in this diagram

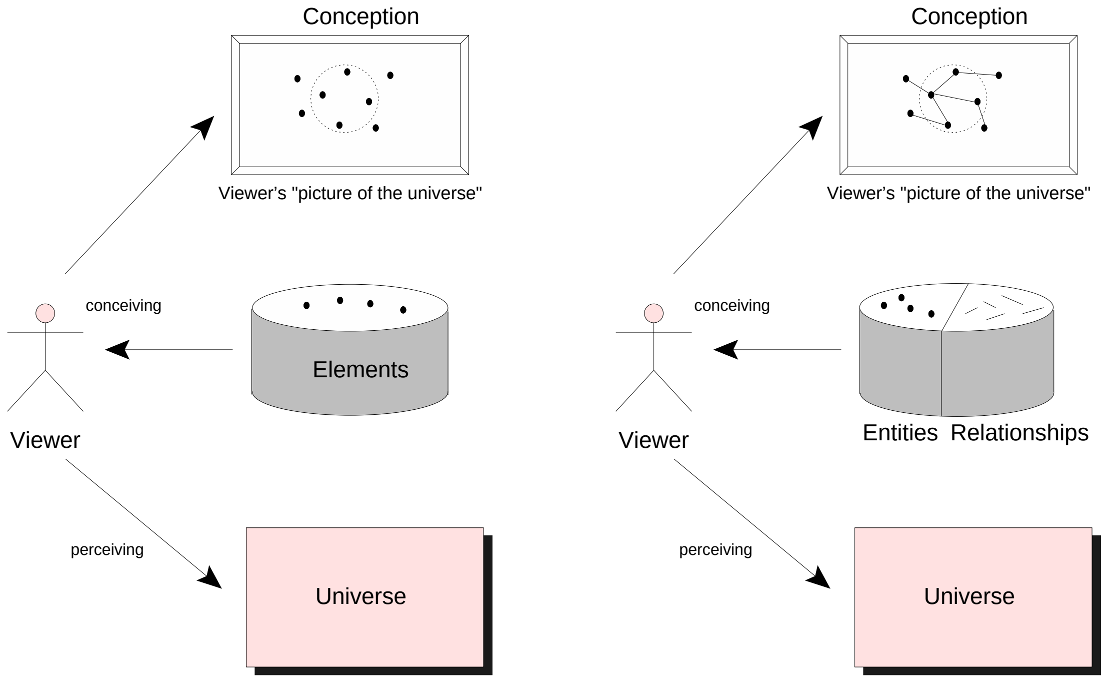

Figure 2: Mentally 'painting' a picture of the observed domain [9, 23].

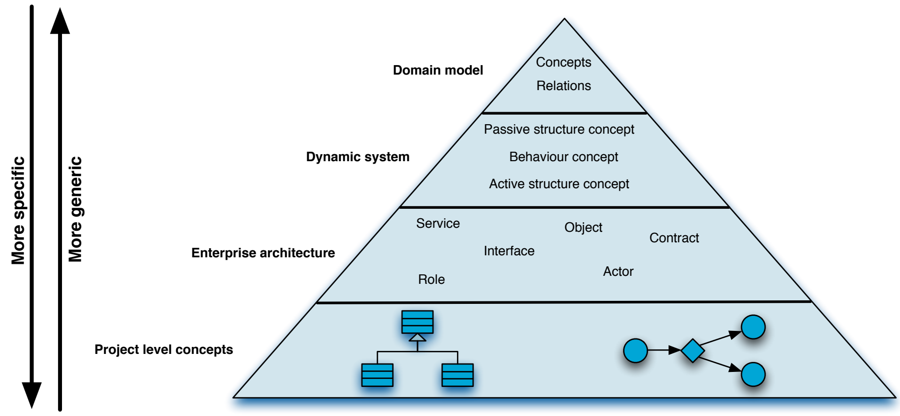

Figure 3: The concept hierarchy of the ArchiMate language [16]

2.6. Focus on the core skills

The proposed textbook will focus on the core skills involved in conceptual modeling. Basically involving the conscious, and controlled, creation and externalization (i.e. capturing as a social artifact) of an abstraction of a domain.

In real-world situations, the *purpose* of a model, especially in terms of scoping, level of detail, intended use, etc, are important and will have a major impact on the *requirements* one puts on the model, and ultimately the tasks involved in creating the model. In this textbook, however, the focus is on learning the core skills involved in the act of modeling. In the aforementioned (potential) follow up textbooks on e.g. process modeling and enterprise architecture, we will be able to more explicitly add the *purpose* dimension.

3. Suggested outline of the textbook

The suggested textbook would start with a discussion of the preliminaries. After that we suggest to follow four stages in *teaching* (and doing) modeling: (1) explore a domain and operationalize the models, (2) ground the resulting understanding on a foundational ontology, (3) manage (in line with the model's purpose) the complexity of the resulting model, and (4) use the model as a base for more purpose/domain specific models (e.g. BPMN, DEMO, and ArchiMate models.).

The modeling process as followed in specific real world contexts may follow a different (e.g. more cyclic) order. However, at the same time, it is important to refer back to the earlier referenced analogy regarding the use of 'swim-paddles' when improving one's swimming stroke. As such, we do posit that the suggested order is a good order to follow when 'still' learning how to model. Furthermore, we argue that the four stages provide a natural progression in priorities (explore the domain, grounding of the model, manage complexity, and specialize the model towards specific purposes).

3.1. Motivation for conceptual modeling

Before embarking on the journey to learn conceptual modeling, it makes sense to create some awareness of the things that could go wrong when conceptual models are left implicit or badly structured.

This could involve the discussion of some examples and cases, but can also involve some exercises of what may happen if the conceptual model is e.g. left implicit.

3.2. Preliminaries

We would start the journey on learning how to model with a a general positioning (and philosophic orientation) of the key notions as used in the book.

Learners are likely to already have a basic, if only at an informal level, understanding of what a model is. We, therefore, plan to start with a reflection on the definition of domain model, and conceptual (domain) model in particular. In line with the discussion in section 2, we would also use the semiotic triangle (fig. 1) to better position the creation, and use of, conceptual models as being linguistic acts, while also stressing the inter-subjectivity in which models are embedded.

Finally, we would lay the foundation of the learners' awareness of their ontological commitments, by which they will 'paint their pictures of the world' in line with fig. 2.

We also plan to reiterate some of the consequences of the philosophic orientation in the later chapters of the textbook, to make things more concrete along the way. For instance, by clarifying that they are using an increasingly more specific ontological commitment by which they 'paint their pictures of the world' (see fig. 2).

Exercises for learners would involve:

- Basic questions regarding the semiotic triangle.
- Basic questions regarding the notion of model.
- Basic questions regarding the role of ontological commitments.

3.3. Stage 1 – Creating an initial conceptual model

In learning to create the initial conceptual model, students will need to first follow a series of steps. The planned text book would discuss each of these steps.

3.3.1. Verbalize domain abstractions

In this step, students would need to verbalize, using the aforementioned telephone/chat heuristic, formulate *elementary propositions* about the domain they are to model. The *elementary* in this context refers to the need to ensure that each proposition is formulated in such a way that it cannot be split further without loosing information. For instance, the proposition "John smokes and drives the car with license plate 925 DJX" is not elementary as it can (normally) be split into "John smokes" and "John drives the car with license plate 925 DJX". In this context, fact-based modeling approaches actually speak about elementary facts. However, since we suggest to treat the identification of truth-makers as a specific step in the modeling process, we prefer to first speak about *elementary propositions* before we can speak about *elementary facts*.

Exercises for learners would involve:

- Identify and reflect on the specific scope of the domain to be modeled and the purpose for which the model is created.
- Identify if a given example verbalization is elementary or not.
- Given a text describing a domain and/or set of tables reporting on sample instances from a domain, express this in terms of elementary propositions.

3.3.2. Identify typing

In principle, this step follows the flow as suggested in most fact-based modeling approaches. Given a set of elementary propositions, one can look for types and their (example) extensions. Symmetric to what happens in database design, this also involves defining how we can refer to/uniquely identify individuals (i.e. the counterpart of the *information principle* and the *closed world assumption*). In the context of conceptual database design [6] examples would be Person John fueled Car 925 DJX, Person Paula fueled Car 254 AZI, and Car LH 62 GV fueled by Person Jim. This would lead to the object types Person, John, and Car, and finally the proposition type Person ... fueled Car ....

Exercises for learners would involve:

- For a given set of elementary proposition, identify the proposition types.
- For a given domain description (and example instances), execute the modeling steps as discussed so-far.

3.3.3. Clarify how we may refer to individual instances

The next step would be to make explicit how we may refer to the individual *things* in the domain. In many situations this involves some 'string' or 'number'. For instance, a person name, a social number, a license plate number, a bank account number, etc.

However, in a general sense, one cannot always assume to have an 'identifier' to refer to specific things. For instance, we may observe Person Erik Proper took a photo of **some** Elephant, as well as Person Hans van Wijck took a photo of **some** Elephant. with as the underlying proposition type Person ... took a photo of Elephant ... In these verbalizations of the instances, the "**some**" in "**some** Elephant" acknowledges the fact that the observed situation involved a specific elephant, but that we do not have an identifier for said elephant.

When using conceptual models in a database design context, such situations are frowned upon, as one would like to know which elephant to enable the storage of the fact in a database. In such case, one is forced to either introduce such an identification, or treat the observation as a unary proposition type: Person Hans van Wijck took a photo of an Elephant and/or switch to a more generic interpretation: Person Hans van Wijck took a photo of an Animal of Kind elephant. In this case, the resulting model is not a conceptual model of the *full* domain, but rather, but a conceptual model of the way the domain will be *reported on* in the database.

Exercises for learners would involve:

- For a given domain description, given example instances, identified object types and proposition types, identify the way we would/could refer to the instances.
- Even though we want learners to understand the applicability of conceptual modeling to be broader than database design, it does make sense to let learners 'play' with some database exercises.

  This should certainly involve SQL related exercises, but could also include e.g. Prolog or Datalog related exercises.

  Even more, for more advanced groups of students, one could use a (knowledge) graph database related exercises to also operationalize the *domain aware* aspect of information systems, connecting the (graph representation of the) conceptual model to its instances.
- For a given domain description (and example instances), execute the modeling steps as discussed so-far.

3.3.4. Create a graphical representation, and conduct a population check

Using the object and proposition types identified so-far, students can create a first graphical representation of the conceptual model. We suggest to use an ORM-alike notation [6] that allows for the inclusion of example instances (possibly involving *anonymized* instances as in **some** Elephant) in the graphical representation, to enable validation of the structure of the model as well as possible rules on the domain. Adding, and reflecting on, the example instances enables a population check to see if the 'fabric' of the present version of conceptual model accommodates the example instances.

Exercises for learners would involve:

- For a set of object types and proposition types, and a set of example instances, create a graphical representation, and reflect on the correctness (e.g. elementarity) of the resulting model.
- For a given domain description (and example instances), execute the modeling steps as discussed so-far.

3.3.5. From propositions to facts

The next step is to have students reflect on truth makers. For each of the object types the learners need to reflect on the criteria that determine the truth-of-existence of its instances. In the case of proposition types, this involves the identification of criteria that make its instances true or not. These truth-makers need to be added to the conceptual model. This reflection may also lead to refinements of the model as it stands so-far.

Exercises for learners would involve:

- For a domain description, and an associated set of object types and proposition types, determine the truth makers.
- For a given domain description (and example instances), execute the modeling steps as discussed so-far.

3.3.6. Adding domain rules

This step involves the identification of rules. In terms of [24] this can involve both alethic rules and deontic rules. Alethic rules (including derivation rules) involve necessities, which cannot, even in principle, be violated by, typically because of some physical or logical law. Deontic rules involve obligations, which may be violated, even though they ought not.

An example of an alethic rule is: **no** person is a parent of **themselves**. Two examples of a deontic rule are: **each** person who teaches at a University **ought to have** a Doctorate, and **each** person who drives a car **must** have a valid drivers' license.

This introduction of such domain rules, will involve both graphical representation of rules, as well as SBVR-based [25] rules in a semi-natural language format.

Reflecting on such rules, may also lead to further refinement of the conceptual model. For instance, leading to the nuance of a person having a drivers' license, versus having a *valid* driver's license.

Exercises for learners would involve:

- For a domain description, an associated set of object types and proposition types, as well as a textual description of alethic/deontic rules, formulate these rules in graphical/SBVR-based format.

3.3.7. Allowing for – instance level – changes in the domain

The modeling steps so-far have ignored the question of possible changes in the domain that is being modeled. Propositions/facts may change over time. The goal of this step is to add a time dimension to the conceptual model, also requiring a temporal extension to the elementary propositions, with associated refinements. For example, the earlier example of John and Paula fueling (their) cars, could now be refined to the following 'log book' [26][3] that reports on events that happen:

> Person John fueling Car 925 DJX STARTED AT 22.03.2023 10:51:20
> Person John fueling Car 925 DJX ENDED AT 22.03.2023 10:53:29
> Person Paula fueling Car 254 AZI STARTED AT 22.03.2023 10:52:12
> Person Paula fueling Car 254 AZI ENDED AT 22.03.2023 10:55:29

Adding a temporal ordering in which events occur, also enables the introduction of history related domain rules. For instance: When a person fuels some car at some point in time, then that person should own that car at that same point in time.

Exercises for learners would involve:

- Given a log of facts that occurred in a domain, show the conceptual domain model and its 'population' at various points in time and reflect on the model evolution.
- Express some history related rules in an SBVR like format.

3.4. Stage 2 – Ontological grounding

The use of (foundational) ontologies allows modelers (and learners) to develop semantically sounder models. Such ontological grounding would entail 'stereotyping' the elements from the conceptual domain model in terms of foundational classes. As an example, the *cars* that have been used as an example several times in this paper could be tagged as being *kinds* of things (in the OntoUML sense).

By attempting to link the model elements to a conceptually sound ontology (e.g. UFO [27]) we do more than simply enrich the model; it also offers benefits in terms of creating a conceptual model that is sound (conforming to ontological commitment that fits with the rules specific by the ontological model). For example, we could discover that our model has a 'specialization' that, according to the ontological constraints, is not allowed. It could also be that we have defined a concepts (performer, which is a person, and band, which is a group of persons/people) and failed to identify the fact that they may belong to a mixin-type (performer-artist) that allows us to reason about both types of performers in a consistent manner.

At the end of this stage, we can also reflect on the possible iterative process flow during actual modeling assignments. When developing the initial model there is, of course, no objection to *already* provide the grounding on a foundational ontology 'as one goes along'. As long as this 'grounding' (depth first) does not hamper modelers in 'exploring' (breadth first) their understanding of the domain. As such, during stage 1, the priority has to remain on *exploring* the domain, while in stage 2 this shifts to *grounding* the resulting domain model on a foundational ontology.

Exercises for learners would involve:

[3] This involves actually an interesting modeling approach to reflect on in relation to the 'process logs' as presently used in the context of process mining.

- Given a conceptual domain model at a certain point in time, take four related model elements and ‘tag’ them with their ontological class. Explain your choices.
- Given a log of facts that occurred in a domain and the associated model evolution, reflect on the notion of (p)endurants.

3.5. Stage 3 – Managing complexity of domains

Novice modelers tend to struggle with conceptual modeling for several reasons. In teaching different modeling approaches, we have noticed that *the number of model elements* as offered by a modeling language may cause major problems in learning the basic skills of modeling. Furthermore, if the domain scope is large – meaning that modeling it would lead to many concepts and relationships – it can also be daunting to oversee what is going on. Simple rules such as ‘start small’ are not really helpful in such situations as they offer little practical guidance.

There are several strategies that we will include in this part of the proposed book. Producing a definite list would certainly be part of future. Nevertheless, some options include: *a*) strategies for (functional) decomposition of a domain with a high level of complexity, allowing modelers to tackle one part of the domain at a time, *b*) strategies to start with a very coarse-grained model and step-wise refinement to tackle complex domains, and

Exercises for learners would involve:

- Applying different techniques to manage the complexity of a (conceptual model of a) domain.

3.6. Stage 4 – Linkage to more specific modeling languages

In this last stage, we will show the learners how to use the resulting conceptual models as a base to develop (and ground) more specific models, based on earlier work in the context of e.g. UML [10, 11], System Dynamics [12, 13], Process Modeling [14, 15], and ArchiMate [9, 16].

4. Conclusion

In this paper we provided the outline of a planned textbook on conceptual modeling, targeting learners of conceptual modeling in general, and bachelor-level students in particular. As stated in the introduction, our goal with this paper is to seek feedback regarding the setup of this textbook, as well as further validate the need for such a textbook.

References

[1] M. M. Lankhorst, F. Arbab, S. F. Bekius, M. Bonsangue, H. Bosma, J. Campschroer, M. J. Cuvelier, P. Fennema, L. Groenewegen, S. J. B. A. Hoppenbrouwers, M.-E. Iacob, W. P. M. Janssen, H. Jonkers, D. Krukkert, P. Penders, H. A. Proper, R. J. Slagter, A. W. Stam, M. W. A. Steen, R. J. Wieringa, F. S. de Boer, H. W. L. ter Doest, R. van Buuren, P. A. T. van Eck, D. van Leeuwen, L. van der Torre, G. E. Veldhuijzen van Zanten, Enterprise Architecture

at Work – Modelling, Communication and Analysis, The Enterprise Engineering Series, 3rd ed., Springer, Heidelberg, Germany, 2013. doi:10.1007/978-3-642-29651-2.

[2] M.-E. Iacob, H. Jonkers, M. M. Lankhorst, H. A. Proper, D. A. C. Quartel, ArchiMate 2.1 Specification, The Open Group, 2013.

[3] OMG, Business Process Modeling Notation (BPMN) Specification, Version 1.1, Technical Report formal/2008-01-17, Object Management Group, Needham, Massachusetts, 2008.

[4] J. L. G. Dietz, H. B. F. Mulder, Enterprise Ontology - A Human-Centric Approach to Understanding the Essence of Organisation, The Enterprise Engineering Series, Springer, Cham, 2020. doi:10.1007/978-3-030-38854-6.

[5] J. Gordijn, H. Akkermans, Value Based Requirements Engineering: Exploring Innovative e-Commerce Ideas, Requirements Engineering Journal 8 (2003) 114–134. doi:10.1007/s00766-003-0169-x.

[6] T. A. Halpin, Information Modeling and Relational Databases, from Conceptual Analysis to Logical Design, Morgan Kaufmann, San Mateo, California, 2001.

[7] H. A. Proper, G. Guizzardi, Modeling for Enterprises; Let's Go to RoME ViA RiME, in: T. Clark, S. Zschaler, B. Barn, K. Sandkuhl (Eds.), Proceedings of the Forum at Practice of Enterprise Modeling 2022 (PoEM-Forum 2022) Co-Located with PoEM 2022, London, United Kingdom, November 23-25, 2022, volume 3327, CEUR-WS.org, 2022, pp. 4–15. URL: https://ceur-ws.org/Vol-3327/paper02.pdf.

[8] D. Aveiro, G. Guizzardi, S. Guerreiro, W. Guédria (Eds.), Advances in Enterprise Engineering XII – 8th Enterprise Engineering Working Conference, EEWC 2018, Luxembourg, May 28 – June 1, 2018, Proceedings, volume 334 of *Lecture Notes in Business Information Processing*, Springer, Heidelberg, Germany, 2018. doi:10.1007/978-3-030-06097-8.

[9] H. A. Proper, Foundations of Work-Systems Modeling, Lecture Notes, Radboud University, Nijmegen, The Netherlands, Nijmegen, The Netherlands, 2006. URL: https://www.erikproper.eu/publications/proper2006worksystemsmodeling.pdf. doi:10.48550/ARXIV.2401.16221.

[10] A. I. Bleeker, H. A. Proper, S. J. B. A. Hoppenbrouwers, The Role of Concept Management in System Development – a Practical and a Theoretical Perspective, in: J. Grabis, A. Persson, J. Stirna (Eds.), Forum Proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, Faculty of Computer Science and Information Technology, Riga, Latvia, 2004, pp. 73–82.

[11] H. A. Proper, A. I. Bleeker, S. J. B. A. Hoppenbrouwers, Object-Role Modeling as a Domain Modeling Approach, in: J. Grundspenkis, M. Kirikova (Eds.), CAiSE'04 Workshops in Connection with the 16th Conference on Advanced Information Systems Engineering, Riga, Latvia, 7-11 June, 2004, Knowledge and Model Driven Information Systems Engineering for Networked Organisations, Proceedings, Vol. 1, volume 3, Faculty of Computer Science and Information Technology, Riga, Latvia, 2004, pp. 317–328.

[12] F. P. Tulinayo, P. van Bommel, H. A. Proper, Grounded System Dynamics: A Procedure for Underpinning System Dynamics with a Domain Modeling Method, in: P. Johannesson, J. Krogstie, A. L. Opdahl (Eds.), The Practice of Enterprise Modeling – 4th IFIP WG 8.1 Working Conference, PoEM 2011 Oslo, Norway, November 2-3, 2011 Proceedings, volume 92 of *Lecture Notes in Business Information Processing*, Springer, Heidelberg, Germany, 2011, pp. 112–125. doi:10.1007/978-3-642-24849-8_9.

[13] F. P. Tulinayo, S. J. B. A. Hoppenbrouwers, H. A. Proper, Integrating System Dynamics with Object-Role Modeling, in: J. Stirna, A. Persson (Eds.), The Practice of Enterprise Modeling, First IFIP WG 8.1 Working Conference, PoEM 2008, Stockholm, Sweden, November 12-13, 2008. Proceedings, volume 15 of *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 77–85. doi:10.1007/978-3-540-89218-2_6.

[14] H. A. Proper, S. J. B. A. Hoppenbrouwers, T. P. van der Weide, A Fact-Oriented Approach to Activity Modeling, in: R. Meersman, Z. Tari, P. Herrero (Eds.), On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005, Agia Napa, Cyprus, October 31 – November 4, 2005, Proceedings, volume 3762 of *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 2005, pp. 666–675. doi:10.1007/11575863_86.

[15] P. van Bommel, S. J. B. A. Hoppenbrouwers, H. A. Proper, T. P. van der Weide, On the Use of Object-Role Modelling to Model Active Domains, in: [28], 2008, pp. 473–484. URL: http://ceur-ws.org/Vol-337.

[16] M. M. Lankhorst, H. A. Proper, H. Jonkers, The Anatomy of the ArchiMate Language:, International Journal of Information System Modeling and Design 1 (2010) 1–32. doi:10.4018/jismd.2010092301.

[17] R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, 2nd ed., Benjamin Cummings, Redwood City, California, 1994.

[18] A. Olivé, Conceptual Modeling of Information Systems, Springer, Heidelberg, Germany, 2007. doi:10.1007/978-3-540-39390-0.

[19] H. A. Proper, G. Guizzardi, On Domain Conceptualization, in: D. Aveiro, G. Guizzardi, R. Pergl, H. A. Proper (Eds.), Advances in Enterprise Engineering XIV – 10th Enterprise Engineering Working Conference, EEWC 2020, Bozen-Bolzano, Italy, September 28, October 19, and November 9-10, 2020, Revised Selected Papers, volume 411 of *Lecture Notes in Business Information Processing*, Springer, Heidelberg, Germany, 2021, pp. 49–69. doi:10.1007/978-3-030-74196-9_4.

[20] C. K. Ogden, I. A. Richards, The Meaning of Meaning – A Study of the Influence of Language upon Thought and of the Science of Symbolism, Magdalene College, University of Cambridge, Oxford, United Kingdom, 1923.

[21] S. Nijssen, T. A. Halpin, Conceptual Schema and Relational Database Design: A Fact Oriented Approach, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[22] G. Bakema, J. P. Zwart, H. van der Lek, Volledig Communicatiegeoriënteerde Informatiemodellering, Kluwer, Deventer, The Netherlands, 1996.

[23] H. A. Proper, Grounded Enterprise Modelling, DaVinci Series, Nijmegen Institute for Information and Computing Sciences, Radboud University, Nijmegen, The Netherlands, 2008.

[24] T. A. Halpin, Business Rule Modality, in: [28], 2008, pp. 383–394. URL: http://ceur-ws.org/Vol-337.

[25] OMG, Semantics of Business Vocabulary and Rules (SBVR), Technical Report dtc/06 – 03 – 02, Object Management Group, Needham, Massachusetts, 2006.

[26] P. van Bommel, P. J. M. Frederiks, T. P. van der Weide, Object – Oriented Modeling Based

on Logbooks, The Computer Journal 39 (1996) 793–799.
[27] G. Guizzardi, G. Wagner, J. P. A. Almeida, R. Guizzardi, Towards Ontological Foundations for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story, Applied Ontology 10 (2015) 259–271. doi:10.3233/AO-150157.
[28] T. A. Halpin, J. Krogstie, H. A. Proper (Eds.), Proceedings of the 13th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2008), Held in Conjunction with the 20th Conference on Advanced Information Systems Engineering (CAiSE 2008), Montpellier, France, volume 337, CEUR-WS.org, 2008. URL: http://ceur-ws.org/Vol-337.