Towards a Multi-Stage Strategy to Teach Enterprise Modelling⋆

Hend*erik* A. Proper[1,3 [0000−0002−7318−2496]], Marija Bjeković[1], Bas van Gils[2], and Stijn J. B. A. Hoppenbrouwers[4]

[1] Luxembourg Institute of Science and Technology (LIST), Belval, Luxembourg
[2] Strategy Alliance, Lelystad, the Netherlands
[3] University of Luxembourg, Luxembourg
[4] HAN University of Applied Sciences, Arnhem, the Netherlands
e.proper@acm.org, marija.bjekovic@list.lu, bas.vangils@strategy-alliance.com, stijn.hoppenbrouwers@han.nl

Abstract. This paper is concerned with the teaching of enterprise modelling. Enterprise models play an increasingly important role in society. In general, such models are not created as mere "one off" artefacts. They rather have a life of their own, covering a broad range of uses (from analysis and understanding, via simulation and design, to execution and monitoring), while involving an even broader variety of stakeholders / audiences. In our view, this increased use of, and even increased dependence on, enterprise models, also makes it important to teach people how to model well.
The aim of this paper is therefore twofold. Firstly, it aims to identify key challenges in teaching enterprise modelling. Secondly, it also aims to provide the humble beginnings of a multi-stage strategy to teach enterprise modelling, meeting these challenges. Both are rooted on a theoretical perspective of modelling, as well as practical experiences. We also reflect on the need for future experimentation and theoretical underpinning of the suggested teaching strategy.

Keywords: Enterprise modelling · Teaching enterprise modelling

1 Introduction

Enterprise models play an increasingly important role in society. In general, such models are not created as mere "one off" artefacts. They rather have a life of their own, covering a broad range of uses (from analysis and understanding, via simulation and design, to execution and monitoring), while involving an even broader variety of stakeholders / audiences. In our view, this increased use of, and even increased dependence on, enterprise models, also makes it important to teach people how to model well.

In line with this, the aim of this paper is twofold. Firstly, it aims to identify some of the key challenges in teaching enterprise modelling. Secondly, it also aims to provide the humble beginnings of a multi-stage strategy to teach enterprise modelling, (at least

⋆ This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (`www.fnr.lu`), via the ValCoLa and CoBALab projects.

partially) meeting these challenges. Both will be rooted on a theoretical perspective of modelling, as well as taking on board practical experiences.

The theoretical perspective concerns a fundamental understanding of (enterprise) models, modelling, and (domain specific) modelling languages, also involving earlier work by the authors. The practical experiences involve the experiences of the authors[5] in both teaching enterprise modelling, and real-world experiences in enterprise modelling.

The remainder of this paper is structured as follows. Section 2 will discuss the aforementioned theoretical perspective on enterprise modelling, while also relating this to our experiences in teaching and modelling in practice. This results in some of the key challenges we see towards the teaching of enterprise modelling, that inspired us in designing the suggested teaching strategy. In moving towards this suggested strategy, Section 3 builds on this by introducing the concept of grounded enterprise modelling. This involves the idea of considering an enterprise model in a purpose / domain specific modelling language as being *grounded* on a conceptual model in a more generic modelling language. In doing so, we will also integrate our experiences [36, 35] in co-designing the ArchiMate enterprise (architecture) modelling language [31]. Based on these inputs, Section 4 then provides the outline of an initial multi-stage strategy to teach enterprise modelling. Before concluding, Section 5 reflects on the need for future experimentation with, and theoretical underpinning of, the suggested teaching strategy.

2 A Fundamental View on Enterprise Modelling

When discussing strategies on teaching enterprise modelling, it is important to first establish our fundamental view on conceptual modelling, and enterprise modelling in particular.

2.1 Models and Modelling

We understand *models* as essentially being means of communication about some domain of interest, and the *process of modelling* as a communication-driven process led by a pragmatic focus [23]. This view is inspired by different related research tackling the fundamental modelling aspects such as [54, 47, 19, 55, 56], as well as our own earlier work [27, 30, 36]. In line with this, we consider a model to be: "*an artefact acknowledged by an observer as representing some domain for a particular purpose*" [8].

The *observer* in this definition refers to the group of people involving both the model creators as well as the model's audience. On one extreme, it can refer to the entire society, while on the other extreme, it can refer to an individual. Though it may not be the general rule, in an enterprise modelling context it is very often the case that model creators are at the same time its audience.

Similarly to [19], we define *domain* as any "part" or "aspect" of the world *considered relevant by the observer*. The notions of *world* and *domain* are construed in the

[5] All authors have, next to their work in research, also worked in industry, doing different assignments involving modelling, and / or have been teaching conceptual modelling to students and / or practitioners.

constructivist sense, allowing for actual, past, future and possible worlds. This emphasis is in particular relevant when considering domains outside of physically observable objects, which is typical for enterprise modelling.

The *purpose* of a model is often considered as the main discriminant of the added value of a model [54, 47, 55]. We understand *purpose* as aggregating two interrelated dimensions: (1) the *domain* that the model (should) pertain to, and (2) the intended *usage* of the model by its intended *audience*. The purpose thus provides the basis for identifying required qualities of the specific model [15, 12] (whereby the qualities may be defined in terms of e.g. Krogstie's SEQUAL framework [33, 32]).

The purpose of a model does not only define requirements on the scoping of the represented domain, but also on the actual representation in relation to its intended use and audience. In practice, we observe that when the purpose of a model is not explicitly considered and / or not made clear in the modelling process, modellers also lack clear criteria to scope the domain / model. Especially novice modellers, then run the risk of getting "out of control". To ensure one remains focussed on the purpose of the model, it seems relevant to teach learners about agile principles [5], in particular when applied to modelling [2]. This leads to a first challenge in teaching learners how to model:

Challenge 1: *Learners should become aware of the (guiding) role of a model's purpose.*

By stating that a model is an *artefact*, we have chosen to exclude *conceptions* [19], or so-called mental models, from the scope of our definition. Conceptions are *abstractions* of the world under consideration, adopted from a certain perspective, and indeed share this property with models. However, a conception resides in the mind of a person holding it, and as such is not directly accessible to another human being. To communicate the conception, it has to be externalised. While conceptions reside in mental space, models are necessarily *represented* in physical / digital space.

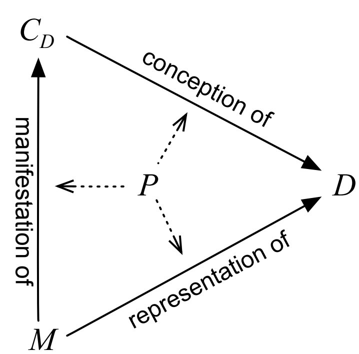

Fig. 1. Fundamental understanding of modelling

The resulting situation is depicted in Figure 1. Given a purpose P, an observer will have a *conception* C_D (in their mind) of the modelled domain D, while some model M is intended to be a *representation* of the domain D, and as such should be the digital / physical *manifestation* of the conception. The purpose P influences the conception of the observer, as well as the needed *representation* and *manifestation*.

Figure 2 refines this, by making explicit that the observer not only has a conception C_D of the modelled domain in their mind, but also of the created model C_M, as well as the purpose C_P. This is an important point, as it underlines the fact that while modelling, multiple observers need to align their conceptions of the domain being modelling, the purpose for which the model is (to be) created, and the actual model itself. This is of particular relevance in the context of collaborative modelling [4, 25, 49, 41].

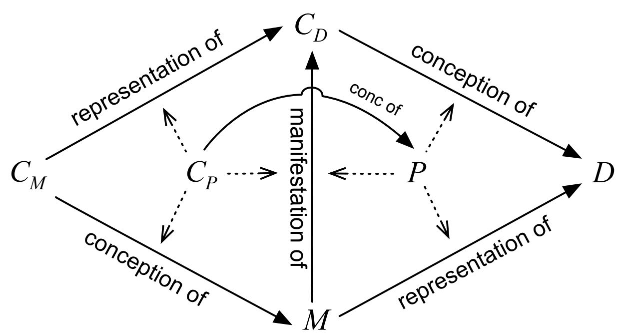

Fig. 2. Conceptions in modelling

Building on the above definition of models, we define a conceptual model to be: *a model where its purpose involves a need to capture knowledge about the represented domain*. In other words, a model answering a need to understand and / or articulate the workings and / or structure of a domain. Such a model needs to reflect human cognition in that it concerns concepts, their relationships, and relevant properties, which makes it a *concept*ual model. An *enterprise model* can be now be defined as a *conceptual model that represent some part and / or aspect of an organisation / enterprise*.

2.2 The Role of a Modelling Language

With this understanding of enterprise models in place, we can turn our attention to modelling languages. As defined in [7, 6], we regard a modelling language as having a *linguistic function* and a *representational function*.

The *linguistic function* refers to the ability of a modelling language to frame the discourse about a domain and shaping the observer's conception of a domain [44]. In this regard, a modelling language should provide a *linguistic structure*, involving a specific classification of concepts to be used in the discourse about the world (the embodied *world view*, or *Weltanschauung*). This linguistic structure will differ between e.g. a modelling language for value modelling and one for process modelling.

The *representational function* refers to the ability of the language to express the conceived domain in a purposeful model. This generally involves a *representation system* involving both an abstract and a concrete syntax of the modelling language.

As discussed above, the purpose of a model is often considered as the main discriminant of the added value of a model [54, 47, 55]. This also entails that if a model, in line with its purpose, needs to be represented in some modelling language, then there has to be an alignment between this purpose, and both the *linguistic structure* and the *representation system* of the chosen modelling language. For example, when the purpose of a given model is to provide senior management with insights into the value exchanges between partners in a business network, then the linguistic function should allow for the expression of concepts such as value, value exchange, and partners. At the same time, the representational function should allow for a representation of a model that is suitable towards the target audience (e.g. senior management).

When learning a modelling language, learners have to master both functions of the language. This means, they have to learn both the *linguistic structure* and the *representation function*. In addition, learners need to learn to judge, for a given modelling language, the aptness of these functions to a modelling purpose at hand.

It is important to acknowledge that the *linguistic structure*, being its essential world view (*Weltanschauung*), may not only limit the freedom of what can be expressed in a model. It may even limit, or at least influence, the way in which modellers observe the domain. This may lead to situations where a modelling language may "feel unnatural", in the sense that the linguistic structure puts to much restriction on a modeller's "freedom of expression". At an anecdotical level, this corresponds to the *hammer* and *nail* paradigm. At a more fundamental level, it corresponds to the notion of linguistic relativity [57][6], which states that the structure of a language determines, or greatly influences, the modes of thought and behaviour characteristic of the culture / context in which it is spoken. As underlying challenges for teaching modelling, we see:

Challenge 2: *Make modellers aware of the role of the modelling language, and its possible costs and benefits towards the purpose of a given model.*

In our experience, learners of an enterprise modelling languages tend to struggle, in parallel, with both the complexities of the (targeted) enterprise modelling language, and getting to grips with the modelling problems that they are asked to solve as part of the learning. This entails figuring out what the main elements in the domain to be modelled are, and then trying to convert those insights into a model conform the modelling language that is used [18]. As such, it seems relevant to distinguish between: (1) learning to conceptualise a domain, in line with a given purpose for the model, and (2) learning how to represent this conceptualisation in terms of the (linguistic structure and representation system of the) target enterprise modelling language.

Challenge 3: *Separation of concerns in learning how to conceptualise a domain, and learning how to represent this in the target modelling language.*

We also find that at the start of the learning process, computer-based tooling tends to get in the way of the learning process. This suggests the need to make a distinction between learning to model, in the given enterprise modelling language, and the use of a supporting modelling tool.

Challenge 4: *Separation of concerns in learning to model using the target modelling language, and the use of an associated modelling tool*

We certainly do not claim that the above challenges are all challenges facing the learners of enterprise modelling. First of all, they certainly do not include the challenges of e.g. collaborative modelling [4, 25, 49, 41], or the challenges of eliciting knowledge

[6] More colloquially also known as the Sapir-Whorf hypothesis.

from domain experts and / or stakeholders. However, we do see the above challenges as being at the core of the basic skills needed for (enterprise) modelling.

3 Grounded Enterprise Modelling

In this Section, we introduce the notion of *grounded enterprise modelling*. We suggest this notion as a way to meet challenge 3, i.e. the need to separate: (1) learning to conceptualise a domain, and (2) learning how to represent this conceptualisation in the target enterprise modelling language. It also will, in our view, *help* meet challenge 2 on making modellers aware of the role of the modelling language, as well as challenge 1 regarding the awareness of the purpose of a model.

Inspired by (1) earlier experiences with the need to better manage domain concepts during software and / or information system development [43, 28, 9] (2) work on explicitly identifying the need to introduce modelling concepts into a modelling language [30, 44], as well as (3) the way in which the ArchiMate language was designed in terms of a series of layers with increasingly more specific modelling concepts [35, 36], we developed the idea to use generic conceptual models to ground other, more specific, models on top of a semantically rich understanding of the domain in terms of a fact-based model [42, 46]. In developing this approach, we also conducted some initial experiments in grounding enterprise models, involving (1) activity models [16, 14], (2) system dynamics models [58, 59], and (3) architecture principles [10].

Grounding enterprise models starts with the observation that enterprise models, being conceptual models, involve *concepts* and their *relations*, as well as a *typing* of these in terms of modelling constructs offered by the modelling language. Consider, as an example, the ArchiMate [31] model as shown in Figure 3. It contains, a.o., the concepts Patient, Doctor, Form, Examine and Diagnose. The icons in the boxes indicate wether a concept is a *role* (e.g. Patient), *activity* (e.g. Examine) or a passive *object* (e.g. Form). The line with the double dots is a so-called *assignment* relation. For example, Doctor and Patient are assigned to the Examine activity. The arrows correspond to triggering rules, so e.g. the Examine activity is triggered by the Register activity.

In line with challenge 3, the key idea is to separate learning how to conceptualise a domain, from learning how to represent this in the target modelling language. We propose to do this by first teaching leaners how to create a conceptual model of a domain in terms of *concepts* and *relations*, and then teaching them how to "interpret" such a conceptual model in terms of the modelling concepts offered by the target language.

Towards the first step, i.e. learning to create a conceptual (domain) model, we have found that using a fact-based modelling approach [20, 3, 39] brings four key advantages, as well as two possible disadvantages.

Firstly, fact-based modelling approaches tend to use a simple (and generic) linguistic structure involving (1) a distinction between *types* and *instances*, (2) three kinds of *objects*: *entities*, *labels* and *facts*, as well as (3) *roles* played by *objects* in *facts*. In addition, generalisation and specialisation of types is possible in terms of sub / super types. The *objects* are used to express the concepts of a domain, while *roles* represent the relationships between concepts, in particular between objects playing a role in facts. This means that initially, learners only need to work with a small set of constructs.

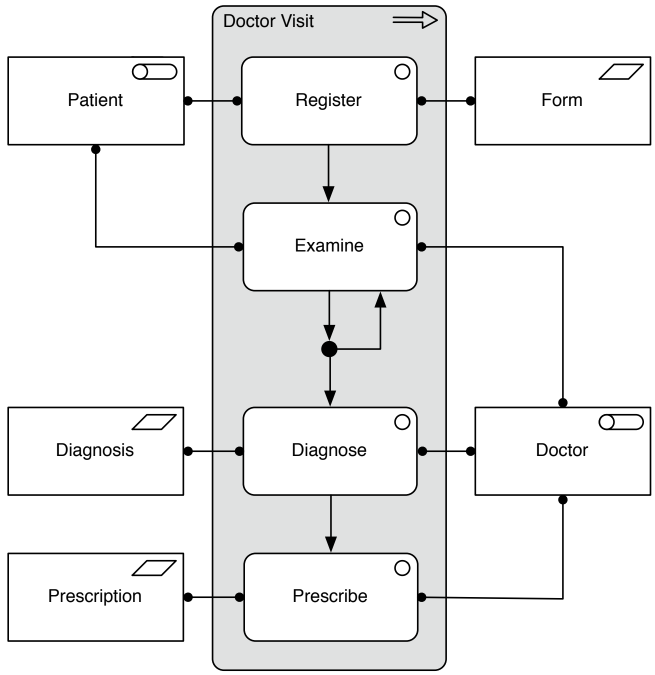

Fig. 3. Example ArchiMate model of a Doctor Visit

Secondly, some of the fact based approaches provide a detailed procedure for modelling [20, 3], which starts by verbalising examples in natural language, and then proceeds with the identification of types, and finally involves the identification of constraints / rules governing the population of the identified types. This provides learners with guidance during the conceptualisation of domains.

Thirdly, fact-based modelling approaches, with their orientation towards facts, are strongly rooted in natural language. Verbalisations in natural language of concrete facts observed in / about the domain to be modelled are used as a starting point for modelling. In our experience, this also helps learners in their efforts to master conceptualisation. Learners and practitioners indeed find the verbalisation of examples rather laborious. Nevertheless, as argued by [20], and supported by our own observations in teaching and modelling in practice, these verbalisations really bring essential conceptualisation decisions to the fore. For learners, this step is, therefore, quite important.

Fourthly, fact-based modelling approaches do not require modellers to make an immediate distinction between entities and attributes. This allows modellers, in particular learners, to explore the structure of a domain first, before having to make a decision on the relative importance between object types.

Using a fact-based modelling approach as a general conceptual modelling approach may also entail disadvantages. A first possible disadvantage is the fact that, similar to ER [17], these approaches have initially been developed for the purpose of conceptual database design. However, as reported in e.g. [53, 26, 43], fact-based modelling can indeed also be used for general domain / ontology modelling.

A second possible disadvantage is the fact that the graphical notation, i.e. the *representation system*, of fact-based approaches tends to be rather elaborate. In the example we give below (see e.g. Figure 5), one can indeed observe how the graphical representation of constraints result in diagrams with a high visual complexity. Firstly, the constraints themselves, in terms of the dotted lines (see e.g. c) and d) in Figure 5) used, arrows, etc., add complexity. Secondly, since graphically expressed constraints need to be "anchored" unambiguously to the roles within fact types, it becomes necessary to include an explicit graphical representation of roles (e.g. by the so-called "role boxes").

At the same time, however, as argued by Moody [38], it is important to realise that a graphical model needs to reflect the complexities of the domain being modelled. Moody motivates this point in terms of Shannon and Weaver's information theory [52], in the sense that a model will need to reflect all information one wants to capture from a domain (given a modelling purpose). As such, one can only aim to avoid unnecessary complexity in the graphical model, where the necessity of complexity depends on the domain being modelled as well as the purpose for modelling. In this sense, the potential disadvantage of the graphical notation of fact-based approaches can be turned into an advantage, by making learners explicitly reflect about the *purpose* of the model, and the needed level of detail (and complexity) of the model and its graphical representation, and showing how (the graphical constructs / abbreviations) of higher level enterprise modelling languages enable them to more clearly focus the key "message" of the model in line with its purpose.

One could, of course, also choose to "hide" the (necessary) graphical complexity by using simpler graphical models and formulating the constraints in a (structured) textual format. This would be the approach as suggested by the SBVR [39] standard for business rules. This, however, would only transfer the inherent complexity of the constraints from the graphical representation of the model to the textual representation.

As part of the learning process, it could be beneficial to confront learners with different *concrete syntaxes* for the same *abstract syntax* in the context of basically the same representation system.

In the remainder of this Section, we will highlight the notion of grounded enterprise modelling, by grounding the example of Figure 3 using an ORM [20] fact-based model. Of course, enterprise modelling in general involves many more different models, including goal models, value models, organisational structures, etc, that can be expressed in even so many different enterprise modelling languages. In this sense, the example below only provides an illustration of the concept of grounded enterprise modelling.

In Figure 4, we see an ORM model[7] dealing with patients visiting a doctor. Patients fill out forms in order to register, they can be examined by a doctor, doctors produce diagnoses, as well as prescribe possibly prescriptions.

What is missing in Figure 4 is the temporal order in which these facts occur, as well as the fact that these activities take place in the context of a Doctor Visit. Adding these aspects, will of course increase the complexity of the graphical model, and as such,

[7] To keep the diagram clean, we have omitted all of the so-called reference schemes, which identify how e.g. a Doctor or a Patient is referred to in this domain

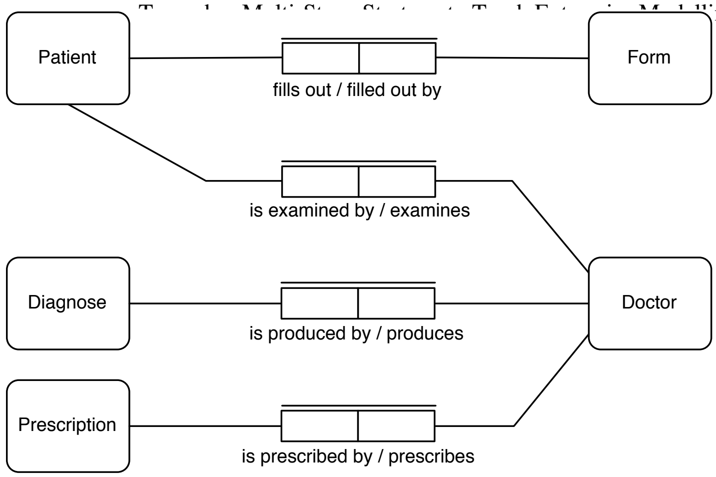

Fig. 4. Doctor Visit example; ORM grounding

prepare learners for the need to use a more purpose-oriented notation. This leads to the situation as shown in Figure 5.

In adding a temporal semantics to ORM [45, 11] we assume that the regular ORM constraints (cardinality, etc.) need to apply at each individual moment in time. So, a mandatory role constraint, such as the one marked with a), should apply at each individual moment in time. In other words, if a Register occurence takes place at some moment in time, then (also during that period in time), it must take place in the context of some Doctor Visit.

Normally, ORM uniqueness constraints are represented with a single bar over the involved roles. Now, consider the uniqueness constraint marked with b). If this one would have been marked with only a single bar, it would have signified that at each moment in time, a Register occurence can only be for one Doctor Visit. This would still make it possible for one Register occurence during some time period T to be assigned to two different Doctor Visits, but at non coinciding intervals in time T_1 and T_2, with $T_1, T_2 \subset T$. The double bar, therefore, signifies that the Registrer occurence can be part of a Doctor Visit once, ever. The patient can of course register for *an other* Doctor Visit by filling out *an other* form.

The required temporal order of events is depicted with an open arrow connecting the involved roles. See, for example, the one marked with c). This states that for Doctor Visit, we cannot see a Register occurrence after we have started to see (an) Examine occurence(s). We also see (the open arrow further below) that (the way it is modelled in the *example*) after a Diagnose occurence has taken place, for a given Doctor Visit, we can no longer see further Examine occurrences in the context of *this* Doctor Visit. Note also, that a Doctor Visit is only allowed to have one Diagnose occurence, but multiple Examine occurrences, as signified by the double bars.

The constraint pattern marked d) is also of interest. It insures that the Patient filling out the Form is also the Patient who is to be examined (in the context of one Doctor Visit). Similarly the Doctor doing the diagnosing is also required to be the Doctor writing the prescription.

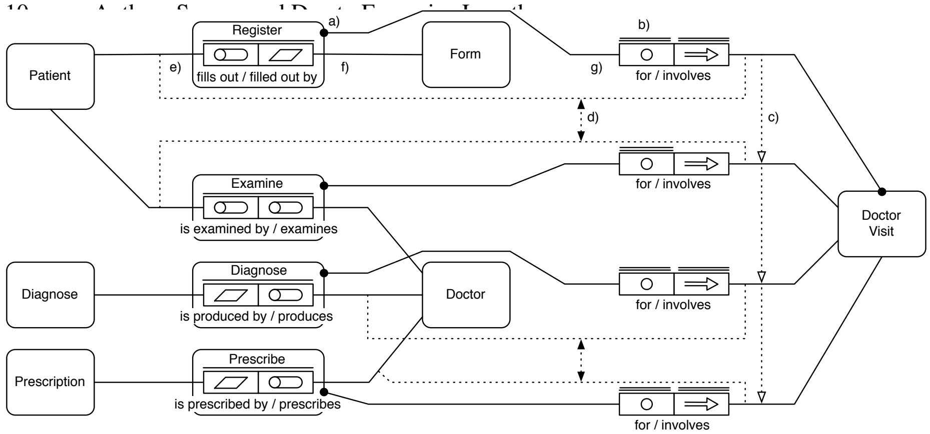

Fig. 5. Doctor Visit example with temporal ordering and ArchiMate mapping

The process flow as depicted in Figure 5 does not involve split / join junctions. Such structures could, however, also be modelled using similar temporal constraints. However, advanced workflow / temporal-ordering patterns, are probably best left to a dedicated modelling language [1]. In grounding enterprise models, we think it is wisest to focus on grounding the main conceptual structure of the domain.

Figure 5 also shows a classification, by means of icons, of roles in terms of the modelling concepts from the ArchiMate language [31]. Doing the latter, provides a transition from capturing the domain in a conceptual model, and mapping the concepts and relations to the modelling constructs offered by the targeted enterprise modelling language. Consider, for instance, the role marked with e). When a Patient fills out a form, then they are, in terms of ArchiMate enacting a *business role*. The form, see f), then plays the passive role of a *business object*. The Register occurrence, see g), plays the role of a *business activity* in the context of a composed *business process* Doctor Visit.

In the case of larger examples, even when limited to educational settings, diagrams in the style of Figure 5 can easily become rather large. Therefore, we would suggest to also use a graphical abbreviation in the ORM diagrams, in terms of a State Sequence (complex) object type, as used on the left hand side of Figure 6. Using such a graphical abbreviation, would also "prepare" learners for the need to switch to a more dedicated graphical notation for the modelling purpose at hand. The version represented on the right hand side, would actually result in a more ArchiMate-alike notation, while maintaining the more explicit verbalisation of the original ORM diagram, as well as the addition of the more specific constraints on role participations of Doctors and Patients

4 A Strategy to Teach Enterprise Modelling

In this Section, we highlight the suggested teaching strategy. As mentioned before, it is to be regarded as the first humble beginnings of such a strategy. As such, we certainly do not (cl)aim it to be a complete teaching method, including suggested teaching materials, evaluation points, etc. The suggested strategy involves five stages, the first four of which

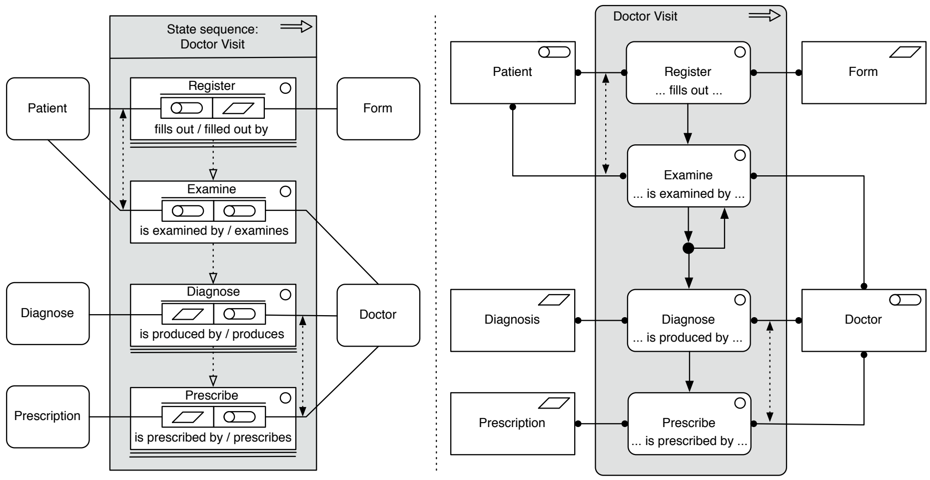

Fig. 6. Doctor Visit example, notational variations

cover the four steps in which we also introduced the Doctor Visit example in Section 3. The first two stages focus on learning conceptual modelling in its basic form, while the next two stages work towards the target enterprise modelling language. The final stage aims to increase the awareness of the role of purpose in modelling.

Stage 1: Time-agnostic domain modelling
At this stage, learners should learn basic conceptualisation skills. To this end, it is important for them to use a modelling approach that has a lightweight linguistic structure. As argued before, we consider fact-based modelling quite suitable for to purpose.

Most fact based approaches provide an elaborate procedure for conceptual modelling [20, 3]. In our teaching activities, we usually use the conceptual schema design procedure (CSDP) from ORM [20]. As its name suggest, this procedure was initially designed for the conceptual design of databases. However, the procedure can also be used when developing general domain models / ontologies. In its original form, the outline of this procedure reads:

1. Transform familiar examples into elementary facts, and apply quality checks.
2. Draw the fact types, and apply a population check.
3. Check for entity types that should be combined, and note any arithmetic derivations.
4. Add uniqueness constraints, and check arity of fact types.
5. Add mandatory role constraints, and check for logical derivations.
6. Add label, set comparison and subtyping constraints.
7. Add other constraints and perform final checks.

Step 1 may seem laborious. Nevertheless, as argued by [20], and supported by our own observations in teaching and modelling in practice, these verbalisations really bring essential conceptualisation decisions to the fore, also aiding in learning to model.

In the context of general domain modelling, one has to take specific care in step 3. Towards database design, one might want to combine entity types for “optimisation” [21] purposes, that, for general domain modelling purposes, might better be kept as individual entity types. Furthermore, steps 4 and 5 also involve the selection of "reference schemes" that define how instances of entity types are to be identified in terms of values / labels. While this, indeed, makes sense in the context of database design, this is not always strictly necessary for general domain modelling [53, 26, 43].

During this stage of teaching, it makes sense to have learners first do assignments on the conceptualisation of basic examples, and then move on to assignments involving examples of the domains that will (later) be modelled in the target enterprise modelling language (such as ArchiMate). However, at this first stage, it is advisable for learners to not yet (have to) concern themselves too much with temporal aspects.

Stage 2: Time-aware domain modelling
At the second stage, it is advised that learners become aware of the role of time, in particular towards the modelling of behaviour in a domain. This means that learners should learn about the concept of time in conceptual modelling, as well as constraints (see Figure 5) dealing with temporal ordering and cardinality over time.

During this stage, learners should apply the procedure as learned in the previous stage to examples involving behaviour, while then also taking temporal constraints into consideration, in particular an initial understanding of e.g. process flows.

We suggest that learners start by focussing on the basic activities in such a domain, and apply the ORM procedure for these first. This would result in models as shown in Figure 4. After this, they can apply the ORM procedure again to complement this model with more complex process aggregations, such as it is shown at the top of Figure 5. This may also lead to modifications of the first model.

Stage 3: Attribute roles with concepts from the target modelling language
At this stage, learners should start to think in terms of the modelling constructs of the "target" enterprise modelling language. With a larger (in terms of number of modelling concepts) language, such as ArchiMate, it makes sense to split this into several levels of specificity. For ArchiMate, following its anatomy [35], this could be:

1. *The layer (Business / Information / Technology) at which a modelling element can be positioned.* In other words, ask learners to mark which object types and in the ORM model belong at which of the three layers.
2. *The involvement of concepts in activities: the actual behaviour, a passive involvement (patiens), or an active involvement (aegens).* This means learners should be able to mark which object types pertain to behaviour, and then identify the kind of involvement by marking the associated roles.
3. *The marking of systemic borders in terms of internal concepts, and interfacing concepts (i.e. interface and service).* This involves learners marking which roles / object types are internal, and which ones are external
4. *The full set of concepts of the ArchiMate language.* This entails a further specialisation of the marking so far towards the actual concepts of the ArchiMate language.

Note that the above process may lead learners to further refine their conceptual model of the domain, as it may (and for didactic purposes should) also lead to further conceptual insight into the domains being modelled.

The suggested overall process would now be: (1) create basic domain model, (2) add temporal aspects in terms of additional object types and constraints, (3) label the resulting model with a mapping to the constructs of the target language.

Stage 4: Convert model to target language, and complete
This stage requires the learners to express the models in the target enterprise modelling language. Initially, assignments should ask learners to go through the entire process from a basic domain model to the final result in the target language.

As a next step, learners can be asked to further complete the model as formulated in the target enterprise modelling language. For example, as already mentioned in Section 3, it would certainly go too far for teaching purpose to be able to e.g. "mimic" advanced workflow patterns at a generic conceptual modelling. These can now be added at the level of the target enterprise modelling language.

Stage 5: Awareness of modelling purpose and the influence of language
At this stage, learners can be confronted with more contextual considerations regarding the context of modelling, and the purpose for modelling.

As discussed in Section 2, when the purpose for the creation of an enterprise model is not clear, modellers not always have good criteria to decide on scoping and the needed level of detail. At this stage, we therefore suggest to teach learners about the agile principles [5] and agile modelling [2] on the one hand, and different relevant qualities of models on the other hand [33, 32]. Furthermore, using assignments, learners can be asked to reflect on the purpose of a model at hand, and the consequences for scoping of the model the needed modelling strategy, and even the requirements on the modelling language.

With regard to the latter, this would also be a good moment to confront learners with different *concrete syntaxes* for the same *abstract syntax* in the context of basically the same representation system, e.g. with regards to the earlier (Section 3) mentioned considerations on the (necessary) complexity in graphical models. This could be supported by assignments, where learners are invited to produce different representations for the same underlying conceptual structures, while reflecting on the suitability of these towards different purposes.

In teaching enterprise modelling, we also have good experiences with learners working in groups. It makes sense for learners to work alone during stage 1 and 2, but once they have acquired a basic level of modelling skils, the interactions involved when working in groups on larger assignments is likely to drive the exploration and learning process, as different views of the group members need to be reconciled.

Especially stages 4 and 5 can really benefit from group based assignments, preferably in combination with some role playing. It is suggested to ensure that the groups jointly articulate the purpose of the model, and the overall modelling strategy to follow.

Within the groups, discussions can be stimulated (e.g. by means of extra questions in the assignments, or by interventions of a lecturer) regarding scoping, purpose of the model, modelling strategy used, concepts of the language, etc. These discussions may take time, but they are likely to prove the joint understanding. See Figure 7 for an example taken from a recent lecture on ArchiMate modelling.

In general, it is also wise to ask the groups to capture their decisions. On the one hand, this invites the groups to be more explicit in their considerations. It, on the other hand, makes it easier for the lecturers / coaches to give feedback.

Stage 6: Tooling
Challenge 4 suggests to separate learning to model in the target modelling language,

Fig. 7. Groups in action

and learning to use an associated modelling tool. During stage 1 to 4, we therefore suggest to avoid the use of such tools, and rather use pen-and-paper based "tooling". This allows learners to focus on first learning to conceptualise (stage 1 and 2), and then focus on getting acquainted with the target enterprise modelling language (stage 3 and 4). We, therefore, suggest to introduce tooling as the last stage of the learning strategy.

Once modellers have gained basic modelling skills (i.e. stages 1 to 5), it would probably be wisest to first use tools that provide modellers with feedback during modelling tasks [50, 51, 48].

5 Reflection

In this Section, we reflect on the validity of the teaching strategy as outlined in Section 4, as well as identify required future work. We will do so from four different angles: (1) foundations of modelling, (2) elaboration of the teaching strategy, (3) utilisation of modelling tools that provide feedback during (learning of) modelling, (4) integration with theories of learning, and (5) empirical validation of the proposed learning strategy.

Foundations: Even though Section 2 provided a theoretical base for the presented strategy, more theoretical underpinnings of the concepts and ideas would be welcome. Three streams of thought that we would like to combine and / or confront with are (1) the notion of basic level categorisation from Lakoff [34] and (2) earlier work on conceptualisation as a linguistic [24] and cognitive phenomenon [37, 60]. Lakoff's notion of basic categories suggest that by the way we have come to experience the world around us, we develop a basic level of categories, that can then be specialised into more refined categories, or generalised towards more abstracted concepts. The underlying mechanics can be useful to provide more fundamental guidance during the initial conceptualisation of domains (stage 1 and 2), in particular to learners. In the same vain, stages 1 and 2 can benefit from fundamental insights into the process of conceptualisation.

Elaboration: As mentioned in Section 4, the strategy as outlined in this paper only provides the humble beginnings towards a more elaborate strategy to learn enterprise modelling. As such, more elaboration of the strategy is needed, possibly even resulting in concrete suggestions for teaching materials and tests.

Feedback: In the discussion of Stage 6 of the proposed (overall) learning strategy, we also suggested that, when starting to use software-based modelling tools, it would be wisest to use tools that provide modellers with feedback during modelling tasks. Experiences by other researchers [50, 51, 48] suggest this may be rather beneficial. This might be combined with strategies to also utilise explicitly captured modelling strategies [29, 13], together with additional explanations and feedback.

Theories of learning: Although the presented strategy, is based on a theoretical underpinning from the perspective of conceptual / enterprise modelling, a theoretical underpinning from a *theories of learning* [22, 40] is lacking. We would expect this to certainly strengthen the theoretical underpinning of the presented strategy

Empirical validation: The presented strategy is based on the collective experiences of the authors in teaching and practicing enterprise modelling. However, more controlled experiments are certainly called for to test if the suggested strategy provides repeatable benefits in teaching enterprise modelling to learners. *Does it really aid learners in their learning process? Does it lead to better modellers?*

In moving towards such experiments, we foresee two strategies. Firstly, during and after the learning process, learners can be asked to fill out a survey. This will allow us to validate if the suggested learning strategy results in the desired insights and effects with the learners, in particular when these data are combined with the results of the assignments and / or exams. Ideally, these surveys should continue once the learners have started to model in practice. Secondly, it would of course be ideal to have control groups. This would enable comparative experiments (across the learning stages) of two groups of similar learners, whereby one group learns a new enterprise modelling language "the traditional way", and one group using the suggested strategy.

6 Conclusion

In this paper we presented the humble beginnings for a multi-stage strategy to teach enterprise modelling. This strategy is both rooted on a theoretical perspective of enterprise modelling, and conceptual modelling in general, as well as the practical experiences of the authors in teaching and practicing modelling.

The paper briefly discussed our theoretical perspective on conceptual modelling, as well as the basic idea to use generic fact-based conceptual models to underpin more specific enterprise models. Based on these, we then discussed the suggested strategy to teach enterprise modelling, involving five stages, that takes learners from learning basic conceptualisation skills, to gradually being able to interpret the domain in terms of the target enterprise modelling language. The last stage of the strategy involves more advanced topics concerning the purpose of the model, and the modelling context.

Before concluding, we also reflected on the need for further theoretical and empirical underpinning, towards the further validation and elaboration of the presented learning strategy.

As a first next step, we aim to develop tool support for the idea of grounding enterprise models, as discussed in Section 3. More specifically, a modelling environment that allows for a gradual “interpretation” [46] of a “flat” conceptual model in terms of a more specific modelling language (such as ArchiMate). This will also enable us to conduct experiments with novice modellers, to validate the expected positive effects of the suggested learning strategy.

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases **14**(1), 5–51 (2003). https://doi.org/10.1023/A:1022883727209
2. Ambler, S.W., Jeffries, R.: Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons, New York, New York (2002)
3. Bakema, G.P., Zwart, J.P.C., Lek, H.v.d.: Fully Communication Oriented NIAM. In: Proceedings of NIAM–ISDM 2. pp. 1–35 (August 1994)
4. Barjis, J.: Collaborative, Participative and Interactive Enterprise Modeling. In: Filipe, J., Cordeiro, J. (eds.) Enterprise Information Systems, 11th International Conference, ICEIS 2009, Milan, Italy, May 6-10, 2009. Proceedings. Lecture Notes in Business Information Processing, vol. 24, pp. 651–662. Springer, Heidelberg, Germany (2009). https://doi.org/10.1007/978-3-642-01347-8_54
5. Beck, K., Beedle, M., Bennekum, A.v., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J. Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development (2001), http://www.agilemanifesto.org, accessed 14 June 2013.
6. Bjeković, M., Proper, H.A., Sottet, J.S.: Embracing Pragmatics. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) Proceedings of the 33rd International Conference on Conceptual Modeling (ER 2014), Atlanta, Georgia. Lecture Notes in Computer Science, vol. 8824, pp. 431–444. Springer, Heidelberg, Germany (2014). https://doi.org/10.1007/978-3-319-12206-9
7. Bjeković, M., Proper, H.A., Sottet, J.S.: Enterprise Modelling Languages – Just Enough Standardisation? In: Shishkov, B. (ed.) Business Modeling and Software Design, Third International Symposium, BMSD 2013, Noordwijkerhout, the Netherlands, July 8-10, 2013, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 173, pp. 1–23. Springer, Heidelberg, Germany (2014)
8. Bjeković, M., Sottet, J.S., Favre, J.M., Proper, H.A.: A Framework for Natural Enterprise Modelling. In: Proceedings of the 15th IEEE Conference on Business Informatics (CBI 2013), Vienna, Austria. pp. 79–84. IEEE Computer Society Press, Los Alamitos, California (2013). https://doi.org/10.1109/CBI.2013.20
9. Bleeker, A.I., Proper, H.A., Hoppenbrouwers, S.J.B.A.: The Role of Concept Management in System Development – A practical and a theoretical perspective. In: Grabis, J., Persson, A., Stirna, J. (eds.) Forum proceedings of the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia. pp. 73–82. Faculty of Computer Science and Information Technology, Riga, Latvia (June 2004)
10. van Bommel, P., Buitenhuis, P.G., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Architecture Principles – A Regulative Perspective on Enterprise Architecture. In: Reichert, M., Strecker, S., Turowski, K. (eds.) Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), St. Goar am Rhein, Germany. pp. 47–60. No. 119 in Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, Germany (2007)

11. van Bommel, P., Frederiks, P.J.M., van der Weide, T.P.: Object–Oriented Modeling based on Logbooks. The Computer Journal **39**(9), 793–799 (1996)
12. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., Roelofs, J.: Concepts and Strategies for Quality of Modeling. In: Halpin, T.A., Krogstie, J., Proper, H.A. (eds.) Innovations in Information Systems Modeling, chap. 9. IGI Publishing, Hershey, Pennsylvania (2008)
13. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: Exploring Modelling Strategies in a Meta-modelling Context. In: Meersman, R., Tari, Z., Herrero, P. (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops - OTM Confederated International Workshops and Posters, AWESOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Acadamy Doctoral Consortium, RDDS, SWWS, and SebGIS, Proceedings, Part II, Montpellier, France. Lecture Notes in Computer Science, vol. 4278, pp. 1128–1137. Springer, Heidelberg, Germany (October/November 2006)
14. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: On the Use of Object-Role Modeling For Modeling Active Domains, pp. 123–145. Research Issues in System Analysis and Design, Databases and Software Development, IGI Publishing, Hershey, Pennsylvania (2007)
15. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: QoMo: A Modelling Process Quality Framework based on SEQUAL. In: Proper, H.A., Halpin, T.A., Krogstie, J. (eds.) Proceedings of the 12th Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2007), held in conjunction with the 19th Conference on Advanced Information Systems (CAiSE 2007), Trondheim, Norway. pp. 118–127. CEUR-WS.org (2007)
16. van Bommel, P., Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: On the use of Object-Role Modelling to Model Active Domains. In: Halpin, T.A., Krogstie, J., Proper, H.A. (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. vol. 337, pp. 473–484. CEUR-WS.org (June 2008)
17. Chen, P.P.: The Entity–Relationship Model: Towards a Unified View of Data. ACM Transactions on Database Systems **1**(1), 9–36 (March 1976)
18. Coenen, A., Gils, B.v., Bouvy, C., Kerkhofs, R., Meijer, S.: Business Modeling Experience for a State Pension Voluntary Insurance Case. In: Harmsen, A.F., Proper, H.A., Schalkwijk, F., Barjis, J., Overbeek, S.J. (eds.) Proceedings of the 2nd Working Conference on Practice-driven Research on Enterprise Transformation (PRET 2010). Lecture Notes in Business Information Processing, vol. 69, pp. 46–60. Springer, Heidelberg, Germany, Delft, the Netherlands (November 2010). https://doi.org/10.1007/978-3-642-16782-9_4
19. Falkenberg, E.D., Verrijn–Stuart, A.A., Voss, K., Hesse, W., Lindgreen, P., Nilsson, B.E., Oei, J.L.H., Rolland, C., Stamper, R.K. (eds.): A Framework of Information Systems Concepts. IFIP WG 8.1 Task Group FRISCO, IFIP, Laxenburg, Austria (1998)
20. Halpin, T.A., Morgan, T.: Information Modeling and Relational Databases. Data Management Systems, Morgan Kaufman, 2nd edn. (2008)
21. Halpin, T.A., Proper, H.A.: Database schema transformation and optimization. In: Papazoglou, M.P. (ed.) Proceedings of the OOER'95, 14th International Object-Oriented and Entity-Relationship Modelling Conference, Gold Coast, Queensland, Australia. Lecture Notes in Computer Science, vol. 1021, pp. 191–203. Springer, Heidelberg, Germany (December 1995)
22. Hilgard, E.R., Bower, G.H.: Theories of Learning. Prentice Hall, Englewood Cliffs, New Jersey (1975)

23. Hoppenbrouwers, S., Wilmont, I.: Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In: PoEM. pp. 190–204 (2010). https://doi.org/10.1007/978-3-642-16782-9_14
24. Hoppenbrouwers, S.J.B.A.: A Functionalist Approach to Conceptualisation. In: Proceedings of the Fourth International Workshop on the Language Action Perspective on Communication Modelling (LAP 2000). Aachener Informatik-Berichte, RWTH Aachen, Aachen, Germany (2000)
25. Hoppenbrouwers, S.J.B.A.: Asking Questions about Asking Questions in Collaborative Enterprise Modelling. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) The Practice of Enterprise Modeling, 5th IFIP WG8.1 Working Conference, PoEM 2012 (Rostock, Germany). vol. 134, pp. 16–30 (2012)
26. Hoppenbrouwers, S.J.B.A., Bleeker, A.I., Proper, H.A.: Facing the Conceptual Complexities in Business Domain Modeling. Computing Letters **1**(2), 59–68 (2005)
27. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: A Fundamental View on the Process of Conceptual Modeling. In: Delcambre, L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O. (eds.) Conceptual Modeling – ER 2005 – 24 International Conference on Conceptual Modeling, Klagenfurt, Austria. Lecture Notes in Computer Science, vol. 3716, pp. 128–143. Springer, Heidelberg, Germany (June 2005). https://doi.org/10.1007/11568322_9
28. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: Formal Modelling as a Grounded Conversation. In: Goldkuhl, G., Lind, M., Haraldson, S. (eds.) Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling (LAP'05). pp. 139–155. Linköpings Universitet and Hogskolan I Boras, Linköping, Sweden, Kiruna, Sweden (June 2005)
29. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: Towards explicit strategies for modeling. In: Halpin, T.A., Siau, K., Krogstie, J. (eds.) Proceedings of the 10th Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'05), held in conjunction with the 17th Conference on Advanced Information Systems (CAiSE'05), Porto, Portugal. pp. 485–492. FEUP, Porto, Portugal (2005)
30. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: Understanding the Requirements on Modelling Techniques. In: Pastor, O., Falcao e Cunha, J. (eds.) 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005, Porto, Portugal. Lecture Notes in Computer Science, vol. 3520, pp. 262–276. Springer, Heidelberg, Germany (June 2005). https://doi.org/10.1007/11431855_19
31. Iacob, M.E., Jonkers, H., Lankhorst, M.M., Proper, H.A., Quartel, D.A.C.: ArchiMate 2.0 Specification. The Open Group (2012)
32. Krogstie, J.: A Semiotic Approach to Quality in Requirements Specifications. In: Kecheng, L., Clarke, R.J., Andersen, P.B., Stamper, R.K., Abou–Zeid, E.S. (eds.) Proceedings of the IFIP TC8 / WG8.1 Working Conference on Organizational Semiotics: Evolving a Science of Information Systems. pp. 231–250. Kluwer, Deventer, the Netherlands (2002)
33. Krogstie, J., Sølvberg, A.: Information Systems Engineering – Conceptual Modeling in a Quality Perspective. The Norwegian University of Science and Technology, Stockholm, Norway (2000)
34. Lakoff, G.: Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. University of Chicago Press, Chicago, Illinois (1997)
35. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The Anatomy of the ArchiMate Language. International Journal of Information System Modeling and Design (IJISMD) **1**(1), 1–32 (2010). https://doi.org/10.1007/978-3-642-01862-6_30
36. Lankhorst, M.M., Torre, L.v.d., Proper, H.A., Arbab, F., Boer, F.S.d., Bonsangue, M.: Foundations (of ArchiMate). In: Enterprise Architecture at Work: Modelling, Communication and

Analysis, pp. 47–66. Enterprise Engineering Series, Springer, Heidelberg, Germany, 4th edn. (2017)

37. van der Linden, D.J.T., Proper, H.A., Hoppenbrouwers, S.J.B.A.: Conceptual understanding of conceptual modeling concepts: a longitudinal study amongst students learning to model. In: Proceedings of the 2nd International Workshop on Cognitive Aspects of Information Systems Engineering (COGNISE 2014) (2014), To appear
38. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering **35**(6), 756–779 (2009). https://doi.org/10.1109/TSE.2009.67
39. Semantics of Business Vocabulary and Rules (SBVR). Tech. Rep. dtc/06–03–02, Object Management Group, Needham, Massachusetts (March 2006)
40. Pask, G.: Conversation, cognition, and learning: a cybernetic theory and methodology. Elsevier, Amsterdam, the Netherlands (1975)
41. Persson, A.: Enterprise Modelling in Practice: Situational Factors and their Influence on Adopting a Participative Approach. Ph.D. thesis, Department of Computer and Systems Sciences Stockholm University/Royal Institute of Technology, Kista, Sweden (2001)
42. Proper, H.A.: Grounded Enterprise Modelling. DaVinci Series, Nijmegen Institute for Information and Computing Sciences, Radboud University, Nijmegen, the Netherlands (2008)
43. Proper, H.A., Bleeker, A.I., Hoppenbrouwers, S.J.B.A.: Object–Role Modelling as a Domain Modelling Approach. In: Grundspenkis, J., Kirikova, M. (eds.) Proceedings of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'04), held in conjunctiun with the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004),. vol. 3, pp. 317–328. Faculty of Computer Science and Information Technology, Riga, Latvia (June 2004)
44. Proper, H.A., Verrijn–Stuart, A.A., Hoppenbrouwers, S.J.B.A.: Towards Utility–based Selection of Architecture–Modelling Concepts. In: Hartmann, S., Stumptner, M. (eds.) Proceedings of the Second Asia–Pacific Conference on Conceptual Modelling (APCCM 2005), Newcastle, New South Wales, Australia. Conferences in Research and Practice in Information Technology Series, vol. 42, pp. 25–36. Australian Computer Society, Sydney, New South Wales, Australia (January 2005)
45. Proper, H.A., van der Weide, T.P.: EVORM – A Conceptual Modelling Technique for Evolving Application Domains. Data & Knowledge Engineering **12**, 313–359 (1994)
46. Proper, H.A., van der Weide, T.P.: Modelling as Selection of Interpretation. In: Mayr, H.C., Breu, H. (eds.) Modellierung 2006. Lecture Notes in Informatics, vol. P82, pp. 223–232. Gesellschaft für Informatik, Bonn, Germany (March 2006)
47. Rothenberg, J.: The Nature of Modeling. In: Artificial intelligence, simulation & modeling, pp. 75–92. John Wiley & Sons, New York, New York, United States of America (1989)
48. Ruiz, J., Serral, E., Snoeck, M.: A fully implemented didactic tool for the teaching of interactive software systems. In: Hammoudi, S., Ferreira Pires, L., Selic, B. (eds.) Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2018, Funchal, Madeira - Portugal, January 22-24, 2018. pp. 95–105. SciTePress (2018). https://doi.org/10.5220/0006579600950105
49. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Enterprise Modeling: Tackling Business Challenges with the 4EM Method. Springer, Heidelberg, Germany (2014)
50. Sedrakyan, G., Snoeck, M., Poelmans, S.: Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. Computers & Education (78), 367–382 (2014). https://doi.org/10.1016/j.compedu.2014.06.014
51. Serral, E., De Weerdt, J., Sedrakyan, G., Snoeck, M.: Automating immediate and personalized feedback taking conceptual modelling education to a next level. In: Tenth IEEE International Conference on Research Challenges in Information Science, RCIS 2016, Grenoble,

France, June 1-3, 2016. pp. 1–6. IEEE (2016). https://doi.org/10.1109/RCIS.2016.7549293
52. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Chicago, Illinois (1949)
53. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. ACM SIGMOD Record **31**(4), 12–17 (2002)
54. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Heidelberg, Germany (1973). https://doi.org/10.1007/978-3-7091-8327-4
55. Thalheim, B.: The Theory of Conceptual Models, the Theory of Conceptual Modelling and Foundations of Conceptual Modelling. In: Handbook of Conceptual Modeling, pp. 543–577. Springer, Heidelberg, Germany (2011)
56. Thalheim, B.: Syntax, Semantics and Pragmatics of Conceptual Modelling. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) Natural Language Processing and Information Systems - 17th International Conference on Applications of Natural Language to Information Systems, NLDB 2012, Groningen, the Netherlands, June 26-28, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7337, pp. 1–10. Springer, Heidelberg, Germany (2012). https://doi.org/10.1007/978-3-642-31178-9
57. Tohidian, I.: Examining Linguistic Relativity Hypothesis as One of the Main Views on the Relationship Between Language and Thought. Journal of Psycholinguistic Research **38**(1), 65–74 (2009)
58. Tulinayo, F., Hoppenbrouwers, S.J.B.A., Proper, H.A.: Integrating System Dynamics with Object-Role Modeling. In: Persson, A., Stirna, J. (eds.) First IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling: from Business Strategies to Enterprise Architectures Stockholm, Sweden 12-13 November, 2008. Lecture Notes in Business Information Processing, Springer, Heidelberg, Germany (November 2008)
59. Tulinayo, F.P., van Bommel, P., Proper, H.A.: Enhancing the System Dynamics Modeling Proces with a Domain Modeling Method. International Journal of Cooperative Information Systems **22**(02) (2013). https://doi.org/10.1142/S0218843013500111
60. Wilmont, I., Barendsen, E., Hoppenbrouwers, S.J.B.A., Hengeveld, S.: Abstract Reasoning in Collaborative Modeling. In: Hoppenbrouwers, S.J.B.A., Rouwette, E.A.J.A., Rittgen, P. (eds.) proceedings of the 45th Hawaiian International Conference on the System Sciences, HICSS-45; Collaborative Systems track, Collaborative Modeling minitrack. IEEE Explore, Los Alamitos, California (2012)