On the Philosophical Foundations of Conceptual Models

Nicola GUARINO [a], Giancarlo GUIZZARDI [b] and John MYLOPOULOS [c,1]
[a] *ISTC-CNR Laboratory for Applied Ontology, Trento, Italy*
[b] *Free University of Bozen-Bolzano, Italy*
[c] *University of Trento, Italy*

Abstract. This paper contributes to the philosophical foundations of conceptual modeling by addressing a number of foundational questions such as: What is a conceptual model? Among models used in computer science, which are conceptual, and which are not? How are conceptual models different from other models used in the Sciences and Engineering? The paper takes a stance in answering these questions and, in order to do that, it draws from a broad literature in philosophy, cognitive science, Logics, as well as several areas of Computer Science (including Databases, Software Engineering, Artificial Intelligence, Information Systems Engineering, among others). After a brief history of conceptual modeling, the paper addresses the aforementioned questions by proposing a characterization of conceptual models with respect to conceptual semantics and ontological commitments. Finally, we position our work w.r.t. to a "Reference Framework for Conceptual" modeling recently proposed in the literature.

Keywords. philosophical foundations, characterizing conceptual models

1. Introduction

In a recent paper, Delcambre and colleagues [9] propose a "Reference framework for conceptual modelling" that attempts to synthesize different views in the conceptual modeling community into a coherent framework, answering questions such as "What is conceptual modelling? Who does it? How are conceptual models different from other kinds of models?" etc.

We applaude the efforts of our colleagues. Indeed, this (long overdue) discussion is definitely necessary, if Conceptual Modelling[2] is to come to enjoy top academic ranking among related research areas, such as Requirements Engineering, Ontologies, Business Process Management, Knowledge Representation, and Object-Oriented Models. More importantly perhaps, such a discussion can hopefully lead to more coherence in the research goals we pursue, the research methods we adopt, and the technical discussions we have in this and related venues.

[1]Corresponding Author: *john.mylopoulos@unitn.it*

[2]We adopt the convention that 'conceptual modelling' refers to the activity of building conceptual models, while 'Conceptual Modelling' refers to the research area concerned with concepts, methods and tools for supporting such activities.

The purpose of this paper is to take a stand and elaborate on some of the most fundamental questions raised by Delcambre et al. Specifically, we propose to address the following foundational questions: (i) What is a conceptual model? (ii) Among models used in Computer Science, which are conceptual, and which are not? (iii) How are conceptual models different from other models used in the Sciences and Engineering? Our perspective is broad: The literature we consider includes any work that tackles the problem of building conceptual models with the final goal of supporting computationally human communication, domain understanding and problem solving. This includes areas such as Databases and Software Engineering (SE), but also Artificial Intelligence (AI), Information Systems Engineering (ISE), Object-Oriented Models (OOM), and Business Process Management (BPM).

The main contribution of the paper is to argue for the thesis that conceptual models are models of conceptual mental representations that cognitive agents build, use and manipulate during cognition. As such, conceptual models are not models of a given domain, but rather models of how *we* conceive of that domain. As discussed in the sequel, this difference has profound implications on the nature of conceptual models, the expressiveness of conceptual modelling languages and the criteria we use to distinguish conceptual models from other types of models. This is not a traditional technical paper. Rather, it is a paper about the philosophical foundations of Conceptual Modelling and their implications.

The rest of the paper is structured as follows. Section 2 presents a brief overview of the history of Conceptual Modelling, Section 3 introduces our answers to the fundamental questions raised above. In Section 4 we propose a characterization of conceptual models with respect to semantics and ontological commitments. Finally, section 5 presents final considerations of the paper.

2. Historical Overview

In this section, we briefly revisit the history of conceptual modeling and discuss its basic foundations. This will help us characterize conceptual models in section 3.

The field of research we call Conceptual Modeling aims at developing theories, languages, tools and techniques for building and reasoning with conceptual models. A similar objective characterizes the related field of Knowledge Representation in AI. As further discussed in the sequel, there are important differences between these two disciplines. However, one that can be mentioned from the outset regards their purposes: in Conceptual Modelling, these models are used to support the design of databases, software, business processes, enterprises etc., whereas in Knowledge Representation, these models (aka knowledge bases) are used to endow an intelligent computational agent with suitable knowledge for the performance of an intelligent task, such as planning, diagnosis, design, etc.

As far back as Aristotle there have been theories that conceptual mental representations consist of concepts and associations that relate concepts. According to empiricists (for example, David Hume), associations come about when concepts systematically co-occur in the experiences of a cognitive agent. For example, the concepts of ‘Student’ and ‘Person’ co-occur every time you encounter a student, so a Student/Person association is meaningful, likely and strong. Most proposals for approaches to conceptual modelling adopt such an associationist stance, including semantic networks, object-oriented models and description logics.

The early history of Conceptual Modelling goes back to the 60s. Ross Quillian [30] proposed in his PhD thesis semantic networks as directed, labelled graphs that model human memory. Nodes of his semantic network denoted concepts (more precisely, word senses), while labelled edges denoted intensional relationships, i.e., relational concepts. His proposal was followed by many others in the early '70. In AI, there were proposals for semantic networks where an interpreter could draw inferences from the labels associated with relationships. Others had attached assertions or procedures to every node, capturing the semantics of the concept being represented ([5,22]). In Databases, there were proposals for semantic data models, notably Peter Chen's Entity-Relationship Model[3] [7], but also many others, such as the Semantic Model (SDM) [1] and Taxis [25]. These proposals did not make claims that their modeling constructs represented concepts. However, all of them claimed that their models were "direct and natural" representations of the domain, or "capture the meaning of their domain" in the sense that they reflect our own perspective of the domain. Among semantic network-based proposals for modelling languages, the KL-ONE knowledge representation language, proposed in the thesis of Ron Brachman, stands out for its treatment of concepts, and the reasoning support provided. In KL-ONE, conceptual mental representations are modelled as descriptions consisting of concepts and roles (aka associations). KL-ONE led to a family of languages known as Description Logics (DLs) that define the state-of-the-art for conceptual modelling languages. This includes the DL-based W3C standard known as the Web Ontology Language (OWL). OWL constitutes a key technology in making the Semantic Web a reality.

Roughly at the same time as Quillian, Ole-Johan Dahl proposed SIMULA 67, a programming language for simulation programs [26]. This language was defined as an extension of Algol 60. The main extension consisted of the notion of a class that had instances, each with associated code so such instances were active, instead of passive data structures. The idea behind SIMULA 67 was that when you want to simulate a domain, for example barber shops, *you define classes for the kinds of concepts* you will be simulating, such as barber shops, barbers, customers and haircuts. So, a simulation language should support the construction of models for a domain to be simulated that are "direct and natural" for modellers. SIMULA was followed by Smalltalk, developed at Xerox PARC, which formed a foundation for object-oriented programming and object-oriented modeling starting in the early 80s. The Unified Modeling Language (aka UML) constitutes a major achievement of this line of research, as it combined several proposals into one, rather loosely defined, language for modeling software designs. UML has been extended into SysML to support modeling systems, as opposed to mere software.

Douglas Ross proposed in the mid-'70s the Structured Analysis and Design Technique (SADT) as a "language for communicating ideas" [31]. The technique was used to specify requirements for software systems. According to SADT, the domain consists of activities and data. Each activity consumes and produces data, but also has data that control its execution but are neither consumed nor produced. Ross' contributions include a modeling language that can capture both static and dynamic aspects of a domain. Again, Ross didn't claim that his models reflect conceptual mental representations. However,

[3]Despite its name, the Entity-Relationship Model is a conceptual modelling language.

he did insist that SADT models should be intuitive and easy to understand, i.e., they correspond to the *way we think of a domain*.

In summary then, conceptual models were proposed and studied in at least four different fields of Computer Science: AI, Programming Languages, Databases and Software Engineering. And in all cases, they were proposed as models reflecting an intuitive, easy to understand, meaningful, direct and natural mental representation of a domain.

3. What are conceptual models

It is a fundamental tenet of Cognitive Science and Philosophy of Mind that cognitive processes generate, use and transform mental representations of the world. Such representations are "intensional" in the sense that they *refer to*, or are *about* something. Mental representations may be *conceptual*, such as thoughts or beliefs, or non-conceptual, such as sensations [29]. Conceptual mental representations rely on representation primitives called *concepts*, which can be seen as the "lens" through which reality is perceived and organized. Concepts reflect those regularities in reality that are cognitively relevant to us. Suppose for example that we want to navigate a given city by using the subway system. By using concepts such as *subway line, subway station, line direction, ordering of stations, intersection between lines*, we can produce a conceptual mental representation of those particular subway lines. In producing such representation, these concepts function as cognitive filters, allowing us to strip out a number of properties that are not germane to the problem we are addressing (e.g., width, luminosity, temperature, humidity of the tunnels).

We shall use the term *conceptualization* to roughly denote the set of concepts in the mind of an agent (see [16] for a more accurate definition). In particular, a conceptualization will include include a set *individual concepts* (such as 'Milan central') and and a set *relational concepts*, intended as intensional properties and relations involving individual concepts.

We shall consider conceptual mental representations as *mental models*, i.e., personal, partial accounts of the external reality, filtered through the lens of a conceptualization, that people use to interact with the world around them [21]. Such mental models may have different levels of generality. On one hand, they may reflect general beliefs about reality, assumed to hold independently of the actual state of affairs (e.g., "every metro line has at least two stations"). On the other hand, they may describe a perceived state of affairs (say, the present state of Milan metro lines).

As shown in Fig. 1, we see conceptual models as explicit descriptions of mental models. They are *information objects* (typically encoded in physical supports such as computer memories) that rely on a *Modeling Language* to describe any aspect of a domain of interest for purposes of understanding and communication. Since the signature of such language is interpreted in terms of (i.e., *commits to*) a *conceptualization*, we can conclude that each conceptual model reflects a conceptualization.

Note that, being an information object, a conceptual model is always the result of an intentional act. In other words, conceptual models are *artifacts* produced with the deliberate intention of describing a conceptualized reality. As discussed by [12], isomorphism between a certain structure and the represented entity is not enough for that structure to

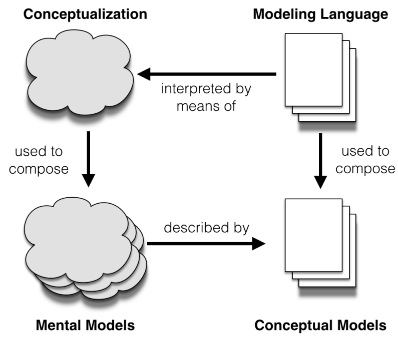

Figure 1. Mental Models and Conceptual Models (adapted from [17])

be a model. We, thus, defend that an explicit intentional act connecting a model and what it represents is needed for that[4].

4. Characterizing conceptual models

In this section we address three questions: What are the main characteristics of conceptual models? How are they different from other models used in the Sciences and Engineering? How can we tell if our model is a good one? In order to address these questions, in the sequel, we elaborate on a number of requirements for a representation to qualify as a conceptual model.

4.1. Conceptual models have a conceptual semantics

On the basis of the above discussion, let us observe that a conceptual model has always a *conceptual semantics*, since the linguistic constructs they use always denote concepts, so the reference to the (real or imaginary) world is mediated by human conceptualization and perception. Conversely, a model that does not have a conceptual semantics is not a conceptual model. As examples of the latter consider the Venn diagram of Fig.2(a) below or the Petri Net of Fig.2.(b). These are diagrams that do have a formal semantics. However, they are not conceptual models, unless their graphical constructs are interpreted as concepts. So, if the circles in the diagram of Fig.2(a) just denote sets, this is not a conceptual model. However, if A and B are mapped (for example) to the *concepts* of Person and Man respectively, then the diagram can be classified as a conceptual model, expressing a particular relationship between the two.

[4]In [12], the author also makes the point that isomorphism is not even necessary because there are many features of reality that we do not want to have represented in the model. This also applies to our case, since the signature of a conceptual modeling language may only cover part of a conceptualization (in general, the cognitively relevant concepts are much more than those we may want or be able to talk of). He also makes the point that isomorphism is a reflexive, symmetric and transitive relation, i.e., an equivalence relation, while a modeling relation is asymmetric, since if A is a model of B we don't say that B is a model of A. The non-reflexivity, asymmetry and non-transitivity of the modeling relation is explained in our case by the additional requirement of an intentional link that makes a certain structure a model of a conceptualized reality, and not the other way around.

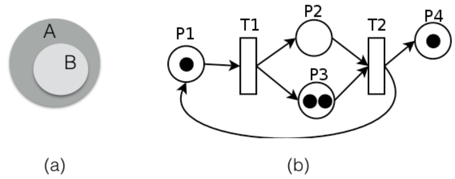

Figure 2. Graphical models but not (necessarily) conceptual models

Secondly, given a fixed conceptualization, we can evaluate the quality of a conceptual modeling language for such conceptualization by evaluating the level of *homomorphism* between these two entities [18]. Ideally, a conceptual modeling language should be: *complete* (to represent all relevant concepts in that conceptualization), *laconic* (to represent those concepts in an unequivocal way), *sound* (such that all the language constructs are interpreted in terms of concepts in that conceptualization), *lucid* (such that all the language constructs are interpreted in an unequivocal way in terms of those concepts). These properties are based on the conversational maxims of pragmatical efficiency put forth by the philosopher of language Paul Grice [15], and can also be used to evaluate the quality of a conceptual model with respect to a particular perceptual experience (a state of affairs) represented by a conceptual mental representation.

4.2. Conceptual models are computational independent models (CIMs)

Considering now the differences between conceptual models and other models used in science and engineering, one question that comes to mind is: are computer programs conceptual models? If not, why?

The short answer is that *some* computer programs may be considered as conceptual models (of a reality external to the computer itself), as long as they are specified by means of a language, i.e., a language whose primitives denote concepts that are abstractions of the external reality. For example, a Logo program can be seen as a conceptual model of a turtle graphic, and a robot program can be seen as a conceptual model of a robot's behavior, *as long as its primitives denote concepts that are abstractions of the robot's behavior*. On the other hand, if the programming language refers to operations inside the computer that controls the robot, then we argue that such a program is not a conceptual model. For sure, it is clear that it is not a conceptual model *of the robot's behavior*. Yet, we may say that a computer program is a conceptual model of the computer's internal behavior, but only as long as its programming language's primitives denote *concepts* concerning computer behavior. If they rather denote *data*, we conclude that such a computer program is not a conceptual model.

Moreover, we would say that those programs that are conceptual models are constructed with the deliberate aim of describing *what* a computer is supposed to do (according to the program's *functional requirements*), independently of non-functional requirements such as computational efficiency. Typical computer programs, in contrast, are built with a focus on addressing computational requirements.

In the Model-Driven Software Engineering literature, there is a clear distinction between so-called Computational Independent Models (CIMs), Platform-Independent Models (PIMs) and Platform-Specific Models (PSMs). Given their respective focus on requirements of nature, we have that conceptual models are CIMs, and PIM and PSM models are not conceptual models. This is not to say that conceptual models are oblivious to: (i) how computational processes can effectively support their construction, verification, validation, analysis; (ii) the fact that, in Computer Science, they are used with the ultimate goal of (through design and implementation) producing software artifacts. Regarding (i), in fact, some of us have defended elsewhere that conceptual models cannot be validated without computational support, and that the real value of conceptual models can only be perceived via the computational analysis they afford [19].

Regarding (ii), certain conceptual modeling languages have received particular appreciation exactly because they can guide the efficient choice of design and implementation constructs [28].

4.3. Conceptual models are not restricted to type-level phenomena

An example of a conceptual model in the sense of Fig.1 is shown in Fig. 3(a). This model describes a mental model of a particular organization, which only focuses on aspects such as subordination relationships and participation in committees. The only concepts used are those of *Organization*, *Organizational Branch*, *Branch Division*, *Work Commission*, *assignment to Branch*, *membership in Commission*, and *being subordinate to*. Consequently, properties of the employees such as height, weight, salary, eye color, etc. are filtered out. Figure 3(b) documents the modeling language used for this representation, which uses graphical primitives. Since its primitives are designed for a specific domain of interest (organizations, in this case), this language is an example of a *Domain-Specific Modeling Language*. Besides being oriented towards a specific domain, most domain-specific languages (including i*, ArchiMate and CORAS [23]) have a further peculiarity: some of their constructs represent instances[5] together with the types they instantiate. In the case of Figure 3, *all* modeling constructs just denote typed instances, so each model built using this language represents a specific state of affairs. While representing Mary we also represent the fact that Mary is an instance of Woman, and when representing that particular division to which Mary is a member, we also represent the fact that this is an Organizational Division.

Another typical example of a conceptual model that represents instances and types connected in this way are *maps*, which rely on a conceptual modeling language (whose semantics is expressed by the map's legend) whose constructs connect individual streets, crossings, etc., with the universals they instantiate [3]. Maps are very different then from *directly depicting representations* – to use a Wittgensteinian expression – such as pictures, etchings or sculptures, which are not conceptual models, being analogical representations instead.

4.4. Conceptual models are ontologically grounded

Consider now mathematical models, such as, for example, Newton's Second Law, which states that $F = M * A$, where F stands for 'Force', M for 'Mass' and A for 'Acceleration'. Would it count as a conceptual model? According to our definition, the answer is negative, since conceptual models are models whose modeling constructs denote concepts. In

[5]Such instances are, more exactly, individual concepts [20]. As demonstrated there, the notion of object-identifier can be seen as a formal representation of the notion of individual concepts tracing the identity of particulars in different states and in different situations.

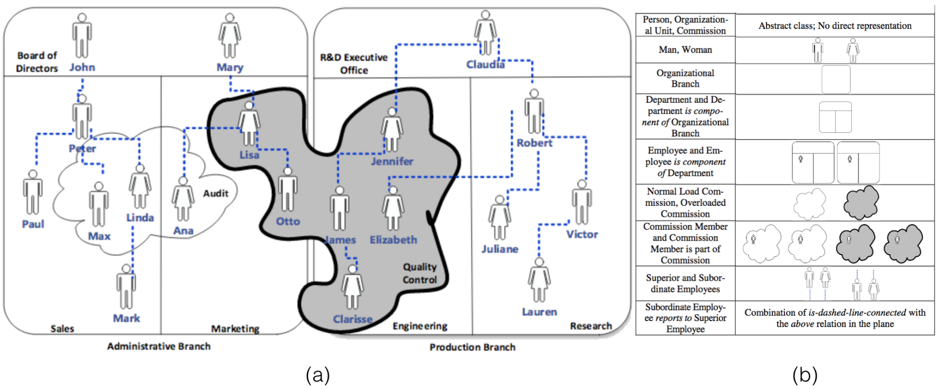

Figure 3. Domain-specific conceptual model (from [32])

the formula above, the modeling constructs are operators and variables. The latter do not denote concepts, but rather actual *values* of physical quantities. The formula establishes a relationship among these values. Of course we need a mental model to make sense of such relationship, but such mental model is just presupposed, and not made explicit.

In contrast, a conceptual model of the phenomenon described by Newton's second law would not represent the values of such physical quantities without representing the physical quantities themselves, which would be considered as *qualities* in ontologies such as DOLCE or UFO[6]. But just having qualities in the conceptual model would not be enough: we cannot have a free-floating quality (say, a mass) without representing its bearer, which is a physical object. More in general, since conceptual models describe mental models, and mental models need to be properly "attached" to the reality at hand in order to be used to interact with the world, as a result conceptual models need to reflect the *hooks* people use for such attachment. This is what we call the *grounding requirement*, which can be understood as sort of *completeness requirement* for a conceptual model. Formally, this means that a conceptual model needs to include the concepts needed to i) track the identity of the individuals it refers to and ii) evaluate the application conditions of the concepts it uses. So, no concepts like *Mass* without *Physical object*, and no concepts like *Student* without *School*.

Coming back to mathematical models, they are not conceptual models since they describe a certain phenomenon in an abstract way, presupposing that somebody *grounds* the equation by attaching variables to values resulting from measurements of particular aspects of such phenomena. Conceptual models, on the contrary, are supposed to provide already the way to ground them. In summary, in line with [4], we argue that conceptual models ultimately represent *phenomena, not data*. Equations describe data, and data are not concepts.

4.5. Conceptual models are not purely logical specifications

Let us now compare conceptual models with logical theories. Our claim is that, although all conceptual models can be represented as logical theories, not all logical theories can

[6]While mass is clearly a quality, the case of force and acceleration is more complex. See [24] for an interesting ontological analysis of forces.

be seen as a conceptual models. To show this, we shall first distinguish between concepts and logical predicates, and then between mental models and epistemic states. We shall then conclude that, in general, logical theories describing epistemic states are not conceptual models, because they do not describe a mental model.

4.5.1. Concepts vs. logical predicates.

As previously discussed, conceptual modelling constructs always denote concepts belonging to a conceptualization. As discussed in depth in [17,20], these concepts are supposed to represent the *universals* used to organize our worldviews. However, as convincingly argued by Armstrong [2] and Bunge [6], although these universals may be represented as logical predicates, it is not the case that for any logical predicate there is a corresponding universal, i.e., a concept in a conceptualization.

To see this, consider a predicate $P(x) \triangleq C(x) \vee M(x)$, defined to be true for all objects x that have either an electrical charge or a mass. Let a and b be two individuals such that both $P(a)$ and $P(b)$ hold. $P(a)$ holds because a has a charge, and $P(b)$ holds because b has a mass. Now, from the fact that the disjunctive predicate P applies to both a and b, can one say that in any serious sense that a and b have something in common? The answer is no, so we conclude there is no genuine universal associated to a disjunctive predicate.

A further argument, according to Armstrong, is based on the link between universals and causality. If a thing instantiates a certain universal, then, by virtue of that, it may causally interact with the world in a certain way. Different universals determine different causal interaction behaviors. For instance, by virtue of having a certain mass, b can act upon a scale. Likewise, by virtue of having a certain charge, a can repel certain things. Now, were P a genuine property of a (or b), it would add something to a's (or b's) behavior. This is clearly not the case.

A similar case can be made for negative predicates. If C is a predicate denoting a universal, then, although the predicate $Q(x) \triangleq \neg C(x)$ would be perfectly acceptable in the language, it does not correspond to a genuine universal in reality, since: (i) there is nothing necessarily in common between two things to which Q applies; (ii) there are no causal behaviors which are determined by not instantiating a universal.

Yet a further reason for distinguishing concepts from logical predicates comes from cognitive considerations. It is well understood in cognitive science that one should differentiate between the property that things have and the *perceptions* we have of these properties according to the perception instruments employed. Bunge [6], for instance, uses the term *substantial property* for the ontological entity and the term *attribute* (or predicate) for its logical or linguistic counterpart, and emphasizes the lack of direct correspondence between the two. According to [14], only attributes representing substantial (i.e., genuine) properties will form *convex* regions (called *quality regions*) in a Conceptual Space. So, for example, while *Red* is a quality region in the color tridimensional space, *Red or Blue* or *not Red* aren't. The philosopher David Lewis [1986] names these genuine properties *natural properties*, as opposed to merely logico-linguistic predicates that denote *abundant properties*.

Finally, denying the one-to-one mapping between logical predicates and genuine properties does not imply denying that, some negative or disjunctive properties can be co-extensional with genuine ones. For example, in a simplistic conceptualization of gender, one may assume that the type *Person* is co-extensional with the disjunctive union of the types *Man* and *Woman*. However, there are indeed genuine properties captured by Person as a type; properties that are common to all persons (e.g., common anatomic parts and physiological processes). Analogously, the type Man is co-extensional with *Person and not Woman*. Once more, *Man* is a genuine type positing genuine properties to its instances. As the philosopher Kit Fine puts it [11], it is not in the very nature of Man that they are not Woman. It just happens to be the case the type *Man super-determinates* [13] some general properties of *Person* and in a way that is incompatible with how the type *Woman* super-determinates these and other properties.

We conclude from the above discussion that a careful distinction must be made between concepts that belong to a conceptualization and the logical predicates that may represent them or their logical combinations. Composite predicates resulting from disjunctions and/or negations of primitive predicates do not represent concepts, as their instances have no common properties.

4.5.2. Mental models vs. epistemic states.

In light of the above, we now move to generalize from concepts to mental models, and from logical properties to logical theories. We see a mental model as a kind of macroscopic concept, which behaves similarly to concepts for what concerns its logical structure. So, if we interpret a mental model as a set of beliefs about a conceptualized reality, the logical form of such beliefs should not include disjunctions nor negations. In other words, if we want to think of a mental model as an epistemic state, it is a *direct*, *positive* epistemic state, not an arbitrary one. In conclusion, since we defined a conceptual model as an explicit description of a mental model, a logical theory whose signature denotes concepts will *not* count as a conceptual model if it includes disjunctions or negations.

Therefore, a conceptual model differs from a knowledge representation. Indeed, they are artifacts aiming at different goals. The former aims at describing the mental model of an agent, which tends to remain constant when the epistemic state changes. The latter, by definition, aims at describing the epistemic state itself and its evolution when new knowledge is acquired. As a consequence, the roles of logical axioms are different. In Knowledge Representation, the role of axioms is to support useful inference; in Conceptual Modeling, it is to rule out unintended interpretations.

4.6. Conceptual models inevitably make ontological commitments

As we have previously discussed, it is a pre-requisite of a conceptual model that it is connected to a conceptualization that provides its *conceptual semantics*. The converse way to put this is that all conceptual models commit to a conceptualization, i.e., to the worldview captured by that conceptualization. In other worlds, all conceptual models make an *ontological commitment*. This is inevitable: any information-bearing artifact that has a conceptual semantics makes an inevitable ontological commitment.

Take for example a model representing the Milan metro network. This model commits to the existence of metro lines, metro stations, line intersections, to the fact that metro lines are composed of stations, that these stations are ordered within these lines, etc. In other words, it commits to the existence of a number of entities, their qualities, their relations, as well as laws constraining how these can be combined, i.e., to a *theory of what exists* in that domain. *Mutatis mutandis*, the same can be said about the model of Fig. 3(b): it commits to the existence of people of certain genders, of organizational branches, divisions, commissions, to the fact that people can be assigned to organizational divisions, that people can be part of commissions, that people in an organization form hierarchies of subordination, but also to domain laws such as that a person cannot be assigned to more than one division, that hierarchies cannot form cycles, etc. In summary, it commits to a conceptualization that reflects a particular *ontological view* of organizations — hence the ontological commitment.

As depicted in Fig.1, conceptual models are artifacts created using modeling languages. Not only specific models commit to particular worldviews (mental models), but the languages in which they are built commit to particular conceptualizations. So, the language used to create Fig.1(a) is one that primarily commits to the ontological view we have just discussed. Note however that such commitment simply reflects the modeler's *intention* to use the language in a particular way, according to the *intended* conceptualization. Nobody forbids to interpret a conceptual model according to a different conceptualization, unless some constraints are introduced either in the modeling language or in the model itself.

So, the main role of constrains for a conceptual model (or a conceptual modeling language) is to make explicit the mental models (or conceptualization) intended by the modeler (or language creator), thus, eliminating as much as possible *unintended interpretations* of that model (or language)[7]. This means that a conceptual model (or conceptual modeling language) should represent not only the different concepts at hand, but also that the domain laws should give rise to model (or language) constraints.

As an example of an unintended interpretation, take a simple model of marriages composed of one single class *Person* and one single relation *married-to*. Now, suppose that there are no additional constraints in that model. As such, this model would allow for liberal conceptualizations in which acceptable interpretations would include: people marrying several people at the same time, people marrying people who are already deceased, people marrying people of the same gender, people marrying themselves, people marrying at the same time people alive and decease, same and different sex, themselves, etc. Checking for these unintended interpretations is very important, since frequently modelers don't realize that—in a sense—the model is making a commitment on their behalf. In summary, while many conceptual models have their own ontological commitment, often implicit, if they are to serve as support for *semantic interoperability* (something that many authors claim to be one of their "killer applications" [27]), conceptual models need to be thought of as sorts of *Meaning Contracts*, i.e., artifacts aiming at precisely capturing and communicating a particular ontological commitment[8]. To put it in a different way, if conceptual models are *intentionally* created descriptions with the purpose of supporting domain understanding and communication [25], we must make sure that they communicate and afford the understanding of the *intended* conceptualizations (mental models).

[7] Given this role, the most important constraints in conceptual modeling are *integrity constraints*. In contrast, in knowledge representation, the main role of constraints is to support inference, i.e., there we have a primacy of *derivation constraints*.

[8] Note that in this case, like a lawyer drafting a contract, the modeler doesn't even have to agree with the worldview represented there, it must, however, guarantee that it is a consistent and explicit representation of the worldview of a given community of stakeholders.

5. Final Considerations

Given the requirements that we put forth in this paper, it should be clear to the reader that conceptual models include a myriad of artifacts used in Computer Science (but not only) to support domain understanding, communication and human problem solving.

Conceptual models are not design or implementation models, i.e, they are not *about* how to best organize and process information inside a computer to address non-functional requirements such as computational efficiency, tractability, portability, adaptability, etc., i.e., in this sense they are CIMs (Computational Independent Models). This is, however, far from downplaying the essential role of computational supporting tools for conceptual modeling. In fact, we defend that conceptual models cannot be validated and managed for complexity without the proper computational support, and therefore the real value of conceptual models in supporting communication, understanding and problem solving can only be achieved through computational analysis. This also does not diminish in any way the fundamental role of conceptual models in guiding the construction of *domain appropriate* and stable PIMs and PSMs.

As descriptions of mental models, of course, conceptual models are not limited to graphical representations. For example, TAXIS and TELOS [25] were early modeling languages that had a phrasal syntax. It should also be crystal clear that conceptual modeling is not restricted to producing (E)ER diagrams! Quite the contrary, there are many notations that may be used for that including (Onto)UML, ORM, OO-Method. Furthermore, conceptual modeling is not restricted to modeling structural aspects of a domain. So, goal models (e.g., in KAOS or i*), business processes models (e.g., in BPMN, or ARIS), value models (e.g., in VDML or e3-value) can all be examples of conceptual models. Finally, conceptual models are not restricted to representing types (schema-level information). So, models that focus on representing instances such as the ones typically produced by domain-specific languages (e.g., risk management models produced in CORAS, or city maps, subway maps) or models that mix instances and types (e.g., enterprise models in ArchiMate) can all be examples of conceptual models.

In summary, it is not by being represented in a specific language or by representing a specific kind of phenomenon in reality that a representation qualifies as a conceptual model. As we propose here, conceptual models are: artifacts (hence, intentionally created for a specific purpose); they are created to describe the mental models of a given domain. In that sense, they are not purely logical models nor models of epistemic states; they are made out of concepts, i.e., they reflect a conceptualization, which gives them a *conceptual semantics*, and to which they make an inevitable *ontological commitment*, which, in turn, must be properly communicated; they are *ontologically grounded*, i.e., they aim at representing phenomena by providing a way to anchor the modeling constructs to the reality at hand.

The positions we defend in this paper are very much aligned with many of the aspects represented in the "Reference Framework for Conceptual Modeling" [9]. By capturing the opinion of a large number of experienced researchers in the field, the framework also highlights that a *"conceptual model is an artifact"*, thus, built with a purpose, that conceptual models should *"expresses the semantics of the scenario of interest"*, and that the purposes of a conceptual model include to: *"answer questions about a domain, improve understanding, and promote knowledge sharing; expose (the author of a conceptual model's) assumptions about a domain; promote communication among people* *developing a conceptual model, or among people who (later) use a conceptual model"*. Finally, they also acknowledge that conceptual models can exist at different *"genericity"* levels, i.e., they *"may be generic and thus intended for use in any application domain"* or be domain-specific. We highlight, nonetheless, that our approach here is very different in method from the contribution of our colleagues, albeit complementary to theirs. Their main objective was to systematize a number of characteristics attributed to conceptual models by consulting the literature as well as experts in the field. In contrast, besides defending some of these characteristics, we also take a stance in addressing the questions of why this is the case. In order to do that, we built on a number of results from the literature of several areas in computer science concerned with the constructions of conceptual models, as well as theories coming from the areas of Philosophy, Logics and Cognitive Science.

Historically, and much before they were used in Computer Science, conceptual models have been fundamental for learning, communicating and reasoning about engineering artifacts. As the historian of technology Eugene S. Ferguson beautifully puts it, a fundamental part of engineering is about reasoning with 'the mind's eye' [10]: *"The mind's eye is a well-developed organ that not only reviews the contents of the visual memory but also forms such new or modified images, as the mind's thoughts require. As one thinks about a machine, reasoning through successive steps in a dynamic process, one can turn it over in one's mind. The engineering designer, who brings elements together in new combinations, is able to assemble and manipulate in his or her mind devices that as yet do not exist. If we are to understand the nature of engineering, we must appreciate this important although unnoticed mode of thought....Pyramids, cathedrals, and rockets exist not because of geometry, theories of structures, or thermodynamics, but because they were first pictures – literally visions – in the minds of those who conceived them."*. Moreover, as Ferguson shows, a radical change in this tradition occurred in engineering education in the past several decades: (*"[s]ince the World War II, the dominant trend in engineering has been away from knowledge that cannot be expressed as mathematical relationships. The art of engineering has been pushed aside in favor of the analytical 'engineering sciences', which are higher in status and easier to teach"*). As he argues, an engineering education that ignores this rich heritage *"will produce graduates that are dangerously ignorant of the myriad subtle ways in which the real world differs from the mathematical models their professors teach them."*.

In this paper, we aimed at revisiting and defending this 'rich heritage of reasoning with the mind's eye' in Computer Science. However, we also strongly defended that elaborating on the philosophical foundations of conceptual modeling is absolutely fundamental for establishing the scientific maturity of the field. As put by the philosopher of science Mario Bunge [6]: *"Every science presupposes some metaphysics"*, hence, a scientific field can either choose to develop and make explicit its philosophical foundations or to remain oblivious to its inevitable and often ad hoc ontological commitments. Or, as nicely put by [8], *"the alternative to philosophy is not no philosophy, but bad philosophy"*.

Acknowledgements. The authors would like to thank João Paulo Almeida and Erik Proper for fruitful discussions on the topic of this article.

References

[1] Abrial, J.R., et al.: Data semantics. Université scientifique et médicale (1974)
[2] Armstrong, D.M.: Universals: An opinionated introduction. Routledge (1989)
[3] Batini, C., Mylopoulos, J.: Abstraction in conceptual models, maps and graphs. In: Tutorial presented at the 37th Intl. Conf. on Conceptual Modelong, ER 2018, Xi'an, China (2018)
[4] Bogen, J., Woodward, J.: Saving the phenomena. The Philosophical Review **97**(3) (1988)
[5] Brachman, R.J.: What's in a concept: Structural foundations for semantic networks. International Journal of Man-Machine Studies **9**(2), 127–152 (1977)
[6] Bunge, M.: Treatise on basic philosophy: Ontology I: the furniture of the world, vol. 3. Springer Science & Business Media (1977)
[7] Chen, P.P.: The entity-relationship model: Towards a unified view of data. ACM Transactions on Database Systems **1**(1) (1976)
[8] Collier, A.: Critical realism: an introduction to roy bhaskar's philosophy (1994)
[9] Delcambre, L.M., Liddle, S.W., Pastor, O., Storey, V.C.: A reference framework for conceptual modeling. In: International Conference on Conceptual Modeling. Springer (2018)
[10] Ferguson, E.S.: Engineering and the Mind's Eye. MIT press (1992)
[11] Fine, K.: Essence and modality: The second philosophical perspectives lecture. Philosophical perspectives **8**, 1–16 (1994)
[12] Frigg, R.: Scientific representation and the semantic view of theories. Theoria. Revista de Teoría, Historia y Fundamentos de la Ciencia **21**(1), 49–65 (2006)
[13] Funkhouser, E.: The determinable-determinate relation. Noûs pp. 548–569 (2006)
[14] Gärdenfors, P.: Conceptual spaces: The geometry of thought. MIT press (2004)
[15] Grice, H.P.: Logic and conversation. 1975 pp. 41–58 (1975)
[16] Guarino, N.: Formal ontology in information systems. In: Proceedings of FOIS'98 (1998)
[17] Guizzardi, G.: Ontological foundations for structural conceptual models. Ph.D. thesis, University of Twente, Enschede (2005), `http://doc.utwente.nl/50826/`
[18] Guizzardi, G.: Ontology-based evaluation and design of visual conceptual modeling languages. In: Domain engineering, pp. 317–347. Springer (2013)
[19] Guizzardi, G.: Ontological patterns, anti-patterns and pattern languages for next-generation conceptual modeling. In: Intl. Conf. on Conceptual Modeling. Springer (2014)
[20] Guizzardi, G.: Logical, ontological and cognitive aspects of object types and cross-world identity with applications to the theory of conceptual spaces. In: Applications of Conceptual Spaces, pp. 165–186. Springer (2015)
[21] Jones, N., Ross, H., Lynam, T., Perez, P., Leitch, A.: Mental models: An interdisciplinary synthesis of theory and methods. Ecology and Society **16**(1) (Mar 2011)
[22] Levesque, H., Mylopoulos, J.: A procedural semantics for semantic networks. Associative Networks: Representation and Use of Knowledge by Computers pp. 93–120 (1979)
[23] Lund, M.S., Solhaug, B., Stølen, K.: Model-driven risk analysis: the CORAS approach. Springer Science & Business Media (2010)
[24] Massin, O.: The metaphysics of forces. Dialectica **63**(4), 555 – 589 (2009). https://doi.org/10.1111/j.1746-8361.2009.01213.x
[25] Mylopoulos, J.: Information modeling in the time of the revolution. Inf. Syst. **23**(3-4) (1998)
[26] O-J. Dahl, B. Myhrhaug, K.N.: Some features of the simula 67 language. In: 2nd Conference on Applications of Simulations. pp. 29–31 (1968)
[27] OMG: Semantic information model federation (simf): Candidates and gaps, available online: http://www.omgwiki.org/architecture-ecosystem/.
[28] Pastor, O., Gómez, J., Insfrán, E., Pelechano, V.: The oo-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. Information Systems **26**(7), 507–534 (2001)
[29] Pitt, D.: Mental representation. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, winter 2018 edn. (2018)
[30] Quillian, M.R.: Semantic memory, Semantic Information Processing. MIT Press (1968)
[31] Ross, D.T.: Structured analysis (SA): A language for communicating ideas. IEEE Trans. Software Eng. **3**(1), 16–34 (1977)
[32] Teixeira, M.G.S. et al.: Can ontologies systematically help in the design of domain-specific visual languages? In: OTM Confederated International Conferences. pp. 737–754. Springer (2013)