Applied Ontology & Conceptual Modeling

Educational Series on Applied Ontology (ESAO)
Launch Event, September, 2021

Giancarlo Guizzardi

unibz
Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

UNIVERSITEIT TWENTE.
Acknowledgements

- Nicola Guarino
- João Paulo Almeida
- Ricardo Falbo (in memoriam)
- Tiago Sales
- Claudenir Fonseca
- John Mylopoulos
- Daniele Porello
- Alessander Botti
- Mattia Fumagalli
“Conceptual Modeling is the activity of representing the physical or social world for the purposes of communication, problem-solving and meaning negotiation among humans”

(Guarino, Mylopoulos & Guizzardi, 2020)
Philosophical Foundations for Conceptual Modeling
Conceptual Model
≈
Interface between Reality and Cognition
Conceptual Modeling ≠ Commitment to Conceptualism or Representation of Epistemic Issues
Another look at data

by GEORGE H. MEALY
Computer Consultant
Scituate, Massachusetts

INTRODUCTION

We have tried to face the facts as they are and adopt a particular ontology, we can avoid a quarrel by adopting their principles. If they contain certain data, we can avoid a quarrel by adopting their principles.

"data are fragments of a theory of the real world, and data processing juggles representations of these fragments of theory..."

Toward a theory of data

Relations

To fix our ideas, consider the following example of genealogical data, taken from Reference 2:
Another look at data

by GEORGE H. MEALY
Computer Consultant
Scituate, Massachusetts

INTRODUCTION

"data are fragments of a theory of the real world, and data processing juggles representations of these fragments of theory...The issue is ontology, or the question of what exists."

Toward a theory of data

Relations

To fix our ideas, consider the following example of genealogical data, taken from Reference 2:
The opposite of Ontology is not Non-Ontology is Bad Ontology!
ontology

A theory about the kinds of entities and their ties that are assumed to exist by an given description of reality
Why is this **important**?
Conceptual Model

≈

Meaning Contract

representing a worldview
Verification = “Did we build the model right?”

Validation = “Did we build the right model?”
Possible Interpretations of a Model
Possible Interpretations of a Model

Intended Interpretations of that Model
Under-constrained Model
Over-constrained Model
Contraints
Contraints
Contraints
A = B
Ontology ≈

An area devoted to developing these domain-independent “toolboxes” with “tools” for supporting ontological analysis
Ontology-Driven Conceptual Model

≈

A model representing the result of an ontological analysis over a given domain
Object Types, Identity and Taxonomic Structures, Part-Whole Relations, Intrinsic and Relational Properties, Weak Entities, Attributes and Datatypes, Events, Multi-Level Modeling,…
Kinds

K1

K2

K3

K4

K5
Roles and Phases
Roles and Phases
Rigid MIXINS
Dynamic MIXINS
Dynamic MIXINS
Endurant Type

Sortal Type

Rigid Sortal Type or **KIND**
(e.g., person, dog, organization, car)

**MIXIN**
(e.g., insurable entity, cultural heritage item)

Dynamic Sortal Type including **ROLES**
(e.g., student, singer) and **PHASES**
(e.g., living person, metropolis)
Endurant

OBJECT
(e.g., UNIBZ, Mick Jagger)

Monadic Aspect

QUALITY
(e.g., a color, a height)

MODE
(e.g., a dengue fever, a knowledge of Dutch)

Aspect

RELATOR
(e.g., a marriage, an employment)
Endurant

OBJECT
(e.g., UNIBZ, Mick Jagger)

Aspect

QUALITY
(e.g., a color, a height)

MODE
(e.g., a dengue fever, a knowledge of Dutch)

RELATOR
(e.g., a marriage, an employment)

existential dependence
What does that buy us?
ODCM Engineering

• Language which primitives reflect a rich system of **ontological distinctions** and grammar reflects ontological **rules**

• methodology reflecting ontological **meta-properties**
In 1950

- Brazilian
- Person
- Man
- Football player
- Philanthropist
- Adult
- Minister of sports
- Actor
- Husband
In 1970
In 1994

- Brazilian
- Adult
- Minister of sports
- Actor
- Husband
- Person
- Man
- Father
- Football player
- Philanthropist
In 2020

- Person
- Man
- Father
- Husband
- Philanthropist
- Adult
- Brazilian
- Football player
- Minister of sports
- Actor

The image suggests that in 2020, the individual was an adult Brazilian man who was married (husband) and a philanthropist. Other roles such as minister of sports, actor, and football player are marked as incorrect.
Solution

1. Characterizing the difference between:

• NATURAL TYPE/KIND (e.g., **PERSON**) = **RIGID SORTAL**
• ROLE (e.g., **MINISTER OF SPORTS, FOOTBALL PLAYER, ACTOR, HUSBAND**) = **DYNAMIC + RELATIONALLY DEPENDENT SORTAL**
• PHASE (e.g., **LIVING PERSON, ADULT MAN**) = **DYNAMIC + RELATIONALLY INDEPENDENT SORTAL**
• MIXIN (e.g., **CULTURAL HERITAGE ENTITY, PHYSICAL ENTITY, INSURABLE ITEM**) = **MIXIN**
Person?
Person?

1. Human Being?
2. Legally Recognized Human Being?
3. Cognitively Capable Human Being?
4. Legal Person?
Person?

1. Human Being? **KIND**

2. Legally Recognized Human Being? **ROLE**

3. Cognitively Capable Human Being? **PHASE**

4. Legal Person? **MIXIN**
How many kinds of rock?

Brachman, Fikes and Levesque, 1985
How many kinds of rock?

Brachman, Fikes and Levesque, 1985
How many kinds of rock?

**Fig. 1.** Kinds of rocks (From [5])

It is clear that, in this example, Brachman and colleagues understood the term “rock kind” in a very simple, minimalist way (perhaps as synonymous with “rock class”), ignoring the fact that, for many people, there are just three kinds of rocks, as taught at high school: Igneous, Metamorphic, and Sedimentary. On the other hand, two of the same authors, in an earlier paper on terminological competence in knowledge representation [6] stressed the importance of distinguishing an “enhancement mode transistor” (which is “a kind of transistor”) from a “pass transistor” (which is “a role a transistor plays in a larger circuit”).

So why was this distinction ignored? My own conclusion is that important issues related to the different ontological assumptions underlying our use of terms have been simply given up while striving for logical simplification and computational tractability. As a consequence, most representation languages, including “ontology languages” like OWL, do not offer constructs able to distinguish among terms having similar logical structure but different ontological implications. In our example, clearly “large rock” and “sedimentary rock” have the same logical structure, being both interpreted as the conjunction of two (primitive) logical properties; yet we tend to believe that there is something radically different between the two: why? To answer this question we have to investigate:

- the nature of the primitive properties “being a rock”, “being large”, and “being sedimentary”;
- the way they combine together in a structured term, while modifying each other.

Unfortunately, while current representation languages offer us powerful tools to build structured descriptions whose formal semantics is carefully controlled to provide efficient reasoning services, still no agreement has been reached concerning the need to adopt proper mechanisms to control the ontological commitments of structured terms.
How many kinds of rock?

Brachman, Fikes and Levesque, 1985
How many kinds of rock?

Fig. 1. Kinds of rocks (From [5])

It is clear that, in this example, Brachman and colleagues understood the term "rock kind" in a very simple, minimalist way (perhaps as synonymous with "rock class"), ignoring the fact that, for many people, there are just three kinds of rocks, as taught at high school: Igneous, Metamorphic, and Sedimentary. On the other hand, two of the same authors, in an earlier paper on terminological competence in knowledge representation [6] stressed the importance of distinguishing an "enhancement mode transistor" (which is "a kind of transistor") from a "pass transistor" (which is "a role a transistor plays in a larger circuit").

So why was this distinction ignored? My own conclusion is that important issues related to the different ontological assumptions underlying our use of terms have been simply given up while striving for logical simplification and computational tractability. As a consequence, most representation languages, including "ontology languages" like OWL, do not offer constructs able to distinguish among terms having similar logical structure but different ontological implications. In our example, clearly "large rock" and "sedimentary rock" have the same logical structure, being both interpreted as the conjunction of two (primitive) logical properties; yet we tend to believe that there is something radically different between the two: why? To answer this question we have to investigate:

• the nature of the primitive properties "being a rock", "being large", and "being sedimentary";
• the way they combine together in a structured term, while modifying each other.

Unfortunately, while current representation languages offer us powerful tools to build structured descriptions whose formal semantics is carefully controlled to provide efficient reasoning services, still no agreement has been reached concerning the need to adopt proper mechanisms to control the ontological commitments of structured terms.

Brachman, Fikes and Levesque, 1985
How many kinds of rock?

The Ontological Level: Revisiting 30 Years of Knowledge Representation

Fig. 1. Kinds of rocks (From [5])

It is clear that, in this example, Brachman and colleagues understood the term "rock kind" in a very simple, minimalist way (perhaps as synonymous with "rock class"), ignoring the fact that, for many people, there are just three kinds of rocks, as taught at high school: Igneous, Metamorphic, and Sedimentary. On the other hand, two of the same authors, in an earlier paper on terminological competence in knowledge representation [6] stressed the importance of distinguishing an "enhancement mode transistor" (which is "a kind of transistor") from a "pass transistor" (which is "a role a transistor plays in a larger circuit").

So why was this distinction ignored? My own conclusion is that important issues related to the different ontological assumptions underlying our use of terms have been simply given up while striving for logical simplification and computational tractability. As a consequence, most representation languages, including "ontology languages" like OWL, do not offer constructs able to distinguish among terms having similar logical structure but different ontological implications. In our example, clearly "large rock" and "sedimentary rock" have the same logical structure, being both interpreted as the conjunction of two (primitive) logical properties; yet we tend to believe that there is something radically different between the two: why? To answer this question we have to investigate:

• the nature of the primitive properties "being a rock", "being large", and "being sedimentary";
• the way they combine together in a structured term, while modifying each other.

Unfortunately, while current representation languages offer us powerful tools to build structured descriptions whose formal semantics is carefully controlled to provide efficient reasoning services, still no agreement has been reached concerning the need to adopt proper mechanisms to control the ontological commitments of structured terms.
How many kinds of rock?

- kind
- igneous rock
- large rock
- grey rock
- sedimentary rock
- metamorphic rock
- pet metamorphic rock
- large grey igneous rock

Brachman, Fikes and Levesque, 1985
Role

• All instances of a given ROLE are of the same KIND (e.g., all Students are Person)
• All instances of a ROLE instantiate that type only contingently (e.g., no Student is necessarily a Student)
• Instances of a KIND instantiate that ROLE when participating in a certain RELATIONAL CONTEXT (e.g., instances of Person instantiate the Role Student when enrolled in an Educational Institution)
• A ROLE cannot be a supertype of a Rigid Type
ERROR: Every sortal class must specialize a unique ultimate sortal. The class Employer must specialize (directly or indirectly) a unique class decorated as one of the following: «kind», «collective», «individual», «abstract».

ERROR: Missing 'allowed' natures meta-property. The class Employer is missing the 'allowed' natures meta-property.
The Emerging **Role** Pattern

```
«kind»
A

...  m ≥ 1

«role»
B

enrolled at

C

m..n
```
Figures are used to illustrate the relationships between different categories. In this diagram, the relationship between Person and its subcategories, such as Deceased Person and Living Person, is shown. The goal is to represent the specialization of categories and the kind of relators in a precise manner. The UML diagram provides a visual representation of the relationships and constraints among the entities involved, which are important for understanding the principles of the domain. This visualization helps in providing a more complete and intrinsic model of the domain, which can then be used for further analysis and design.
In the standard UML conceptualizations that disambiguate two examplifications of how Figure 8.11 from the corresponding founding relators (the roles and (Supplier, type)

Since this process model of the traditional UML association no constrains, becomes even more evident in n-ary relations with n > 2.

This example, the upper constraint in figures 8.11.a and 8.11.b below. Moreover, the relation is the result of (d mediation

The upper constraint of the traditional UML association no can be omitted but whenever a constraint appears on the end connected to G in the H relation end) and the cardinality

models connected to G in the H relation end) and the cardinality

Firstly, this activity, Reflecting the same information (from the conceptualizations. As discussed in sect 6.3.3, the benefits of explicitly

Ontological primitives (e.g., Material, Pattern, Ontology constructs)

Moreover, this idea is reinforced in the following manner:

UN

The upper constraint of the traditional UML association no can be omitted but whenever a constraint appears on the end connected to G in the H relation end) and the cardinality

Models connected to G in the H relation end) and the cardinality

Some temporal affairs which depend on the end connected to G may be translated for temporal simulation. In other words, the relations themselves. In other words, the

Temporal affairs which depend on the end connected to G may be translated for temporal simulation. In other words, the relations themselves. In other words, the

models connected to G in the H relation end) and the cardinality

For temporal affairs which depend on the end connected to G may be translated for temporal simulation. In other words, the relations themselves. In other words, the

models connected to G in the H relation end) and the cardinality

The upper constraint of the traditional UML association no can be omitted but whenever a constraint appears on the end connected to G in the H relation end) and the cardinality

Firstly, this activity, Reflecting the same information (from the conceptualizations. As discussed in sect 6.3.3, the benefits of explicitly

Ontological primitives (e.g., Material, Pattern, Ontology constructs)

Moreover, this idea is reinforced in the following manner:

UN
Fig. 1: A conceptual model in OntoUML in which relators are highlighted in green.
Fig. 1 A conceptual model in OntoUML in which relators are highlighted in green.
Fig. 1 A conceptual model in OntoUML in which relators are highlighted in green.
Fig. 1: A conceptual model in OntoUML in which relators are highlighted in green.
A Relator-Centric Clustering of a model $M$ is a set of views symbolized as $RCC(M) = \{M_1 \sqcup \ldots \sqcup M_n\}$ such that for every $M_i \in RCC(M)$ there is a type $rel$ such that $rel \in C(M)$ and $RC(M_i, M, rel)$. Figure 2 depicts the application of this notion of RCC to the model of Figure 1. Here we represent each Relational Context using UML packages and name these packages with the homonymous focal relator. As one can observe, the original model can be broken down into four contexts, namely: the Car Rental, the Marriage, the Car Ownership, and the Employment contexts. Each of these modules contains a view of the original model with all the information required to understand each of the contexts.
Conceptual Modeling

DESIGN

Implementation₁

Implementation₂

Implementation₃

Figure 3 presents the schema that results from the application of these transformation steps in the conceptual model in Figure 2. We obtain the five tables corresponding to object kinds: PERSON, ORGANIZATION, and three corresponding to relator kinds: EMPLOYMENT, ENROLLMENT, and SUPPLY

CONTRACT. An additional table for the discriminator that results from the overlapping generalization set nationality is introduced (PERSON-NATIONALITY, representing a qua-entity connecting a person to a particular nationality type). Finally, for all the tables representing dependent entities types, we introduce the corresponding dependency keys.

Discussion and Comparison to Alternative Approaches

Table 2 summarizes the comparison between the proposed one table per kind strategy and the three dominant strategies in the literature, where:

- \( n \) is the total number of classes in the source conceptual model,
- \( h \) is the maximum height of the hierarchy (i.e., maximum path size from a top-level class to a leaf class),
- \( n_l \) is the number of leaf classes in the hierarchy,
- \( n_t \) the number of top-level classes, and
- \( n_k \) is the number of kinds.

Note that the number of kinds \((n_k)\) is equal to or lower than the number of leaf classes \((n_l)\) and that they are equal \((n_k = n_l)\) only in case there are no subkinds, roles and phases. Thus, the number of tables to required to represent entities in the domain in the proposed one table per kind strategy is equal to or lower than that required by one table per class and one table per leaf class. The comparison with one table per hierarchy requires us to consider the number of top-level classes \((n_t)\). The two approaches result in the same number of tables when there are no non-sortals \((n_k = n_t)\).

The table also presents worst-case figures for the retrieval and insertion of an entity (with all its attributes). One table per class fares poorly in this comparison, with \(h \) joins...
```
sig Person_Set in Concept { Person: some World }
2{
3 Person in existsIn
4 all w1: World | w1 in Person -> (all w2:
5 w1.access | (w2 in existsIn) -> (w2 in
6 Person)) -- Rigidity
7 some w: World | w in this.Child -- Phase
8 some w: World | w in this.Teenager -- Phase
9 some w: World | w in this.Adult -- Phase
10 : ;
11 }
```
A = B
∀xSupplier(x) \rightarrow \text{ActiveOrganization}(x)

∀xSupplier(x) \rightarrow \lozenge \neg \text{Supplier}(x)

\Box (\forall x \text{Supplier}(x) \rightarrow \exists y \text{Customer}(y) \land \text{contractsWith}(y, x))
Constraints
Contraints
Property3
(Economic Agreement)

Object1
(CorporateCustomer, Organization, Supplier)

Object0
(CorporateCustomer, Organization, Supplier)
The Emerging Anti-Pattern: Relation Between Overlapping Types (RelOver)

Fig. 3. Structural configuration illustrating (a) AC, (b) IA and (c) RWOR.

4.6 Relator With Overlapping Roles (RWOR)

The generic structure of the Relator With Overlapping Roles (RWOR) anti-pattern is depicted in Fig. 3(c). It is characterized by a Relator (R1) mediating two or more Roles, (T1, T2… Tn) whose extensions overlap, i.e. have their identity principle provided by a common Kind as a super-type (ST). In addition, the roles are not explicitly declared disjoint. This modeling structure is prone to be overly permissive, since there are no restriction for an instance to act as multiples roles for the same relator. The possible commonly identified intended interpretations are that: the roles are actually disjoint (disjoint roles), i.e., no instance of ST may act as more than one role for the same instance of a relator Rel1 (mutually exclusive roles); some roles may be played by the same instance of ST, while others may not (partially exclusive roles). An alternative case is one in which all or a subset of the roles in question are mutually exclusive but across different relators. An instance of RWOR is our running example is discussed in section 5.

4.7 Twin Relator Instances (TRI)

This anti-pattern occurs when a relator is connected to two or more «mediation» associations, such that the upper bound cardinalities at the relator end are greater than one. The problem associated with this anti-pattern is that it opens the possibility for two distinct instances of the same relator type to co-exist connecting the very same relata instances. We empirically found that the existence of these relator instances in this situation should frequently be subject to several different types of constraints. For instance, it can the case that there cannot be two different relator instances of the same type connecting the very same relata. An example in the domain depicted in fig. 1 could be: one cannot be the subject of a second criminal investigation as a suspect and be investigated by the same detectives that interrogate the same witnesses.
Relation Specialization (RS)
Antipattern Catalogue

- Association Cycle
- Binary Relation Between Over. Types
- Deceiving Intersection
- Free Role Specialization
- Imprecise Abstraction
- Multiple Relational Dependency
- Part Composing Over. Roles
- Whole Composed by Over. Parts
- Relator Mediating Over. Types
- Relation Composition
- Relator Mediating Rigid Types
- Relation Specialization
- Repeatable Relator Instances
- Relationally Dependent Phase
- Generalization Set With Mixed Rigidity
- Heterogeneous Collective
- Homogeneous Functional Complex
- Mixin With Same Identity
- Mixin With Same Rigidity
- Undefined Formal Association
- Undefined Phase Partition
$M = \{ \forall x \text{Woman}(x) \rightarrow \text{Person}(x), \forall x \text{Man}(x) \rightarrow \text{Person}(x) \}$
Secondly, consider a scenario where several people simulate the same model and people diverge on what they assign as intended and unintended configurations. We can then offer to the modelers possible options giving them an indication of how often people chose each of the options. This is about repairing a particular model by learning from a collective judgment (in this case, a type of meaning negotiation activity).

In summary, from the marriage between model validation, for finding faults, and machine learning, for suggesting repairs, a fruitful synergy emerges, which can support knowledge engineers in understanding how to design and refine rigorous models.
c) negative/positive example set generation.

Towards Automated Support for Conceptual Model Diagnosis and Repair 7

The overall phase from the input conceptual model, through the generation of multiple simulations, to the annotation and the generation of the negative/positive examples set, can be formalized as a composed function of multiple simulations, to the annotation and the generation of the negative/positive examples set.

With some additional features (some of them already provided by the modeler), this will involve an additional input from the modeler, namely the annotation of what instance in the simulation makes the configuration to be marked as "negative" are '#22' and '#23'.

ad hoc...
Secondly, consider a scenario where several people simulate the same model and people diverge on what they assign as intended and unintended configurations. We can then offer to the modelers possible options giving them an indication of how often people chose each of the options. This is about repairing a particular model by learning from a collective judgment (in this case, a type of meaning negotiation activity).

In summary, from the marriage between model validation, for finding faults, and machine learning, for suggesting repairs, a fruitful synergy emerges, which can support knowledge engineers in understanding how to design and refine rigorous models.

Towards Automated Support for Conceptual Model Diagnosis and Repair

9

When instances of 'Person' are neither instances of 'Man' nor 'Woman' ('1');

ii) when instances of 'Person' are both 'Woman' and 'Man' ('4'). The following formulas represent a First Order Logic (FOL) formalization of the derived 'negative' rules.

\[
M = \{ \forall x \text{Woman}(x) \rightarrow \text{Person}(x), \forall x \text{Man}(x) \rightarrow \text{Person}(x) \}\]

\[
M^R = \{ M, \forall x \text{Person}(x) \rightarrow (\text{Woman}(x) \lor \text{Man}(x)), \forall x \text{Man}(x) \rightarrow \neg \text{Woman}(x) \}\]
Take Away Messages

• conceptual modeling is about defining the **ontology of the domain**

• conceptual modeling (domain ontology engineering) is about representing the **result of ontological analysis** over that domain

• All conceptual modeling (domain ontology engineering) should be **Ontology-driven**
No ontology without Ontology!
References

- All papers can be downloaded from:
  - tinyurl.com/3a8s4f7z

- Tools:
  - https://github.com/OntoUML/ontouml-vp-plugin
  - https://nemo-ufes.github.io/gufo/

- Some starting points are:
  - Guizzardi, G., Bernasconi, A., Pastor, O., Storey, V., Ontological Unpacking as Explanation: The Case of the Viral Conceptual Model, 40th International Conference on Conceptual Modeling (ER 2021), St. John's, Canada, 2021.
  - GUIZZARDI, G., Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling, invited companion paper to the Keynote Speech delivered at the 33rd International Conference on Conceptual Modeling (ER 2014), Atlanta, USA.