# Semantics and Conceptual Modelling

C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa,
mkeet@cs.uct.ac.za

*Fourth Interdisciplinary School on Applied Ontology (ISAO 2018)*
*10-15 September 2018, Cape Town, South Africa*

# Outline

# Dissecting the title

- Semantics, which has different meanings:

- Conceptual modelling

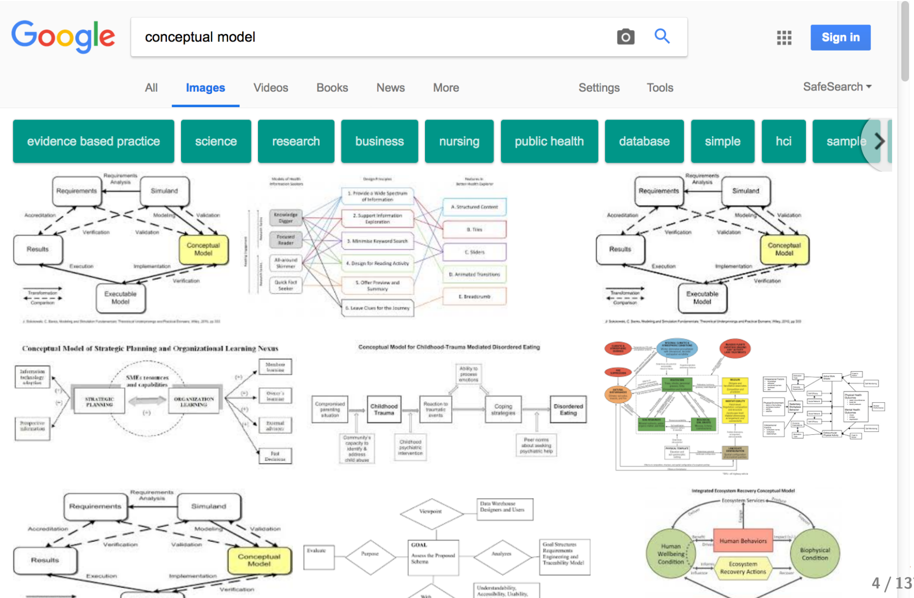
# Dissecting the title

- Semantics, which has different meanings:
  - *Logic*: formal meaning of the 'things' represented with the syntax of a language
  - *Subject domain*: meaning of something (e.g., the definition, characteristics of 'course', 'professor', 'attend' etc.)
- Conceptual modelling

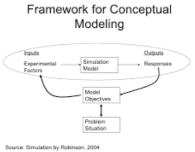
# Dissecting the title

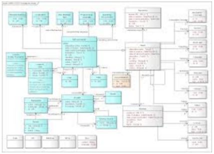
- Semantics, which has different meanings:
  - *Logic*: formal meaning of the 'things' represented with the syntax of a language
  - *Subject domain*: meaning of something (e.g., the definition, characteristics of 'course', 'professor', 'attend' etc.)
- Conceptual modelling
  - The process of creating conceptual models
  - Conceptual data models, like those represented in EER, UML Class diagram notation, ORM
  - Other conceptual models, such as conceptual graphs, petri nets, ....

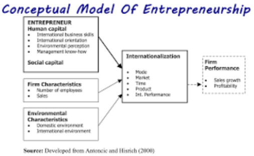
# “Conceptual model”

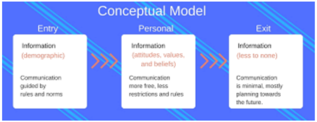


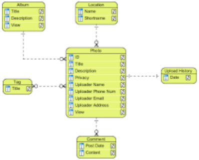
# “Conceptual model”

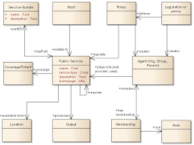


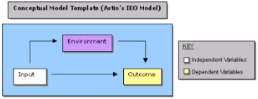



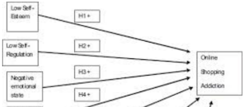


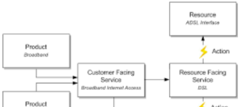



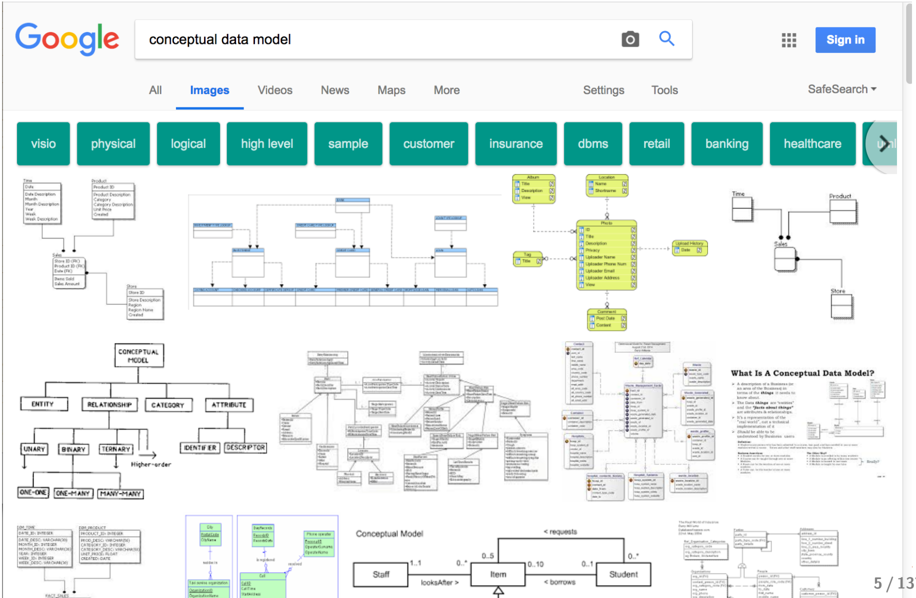






# “Conceptual data model”



# Dissecting the title

- ‘real’ conceptual models vs ‘computational-conceptual’ models
  - Conceptual models *do not* have implementation decisions embedded in them
  - Some models do have some computational decisions; e.g., PK/FK, data types

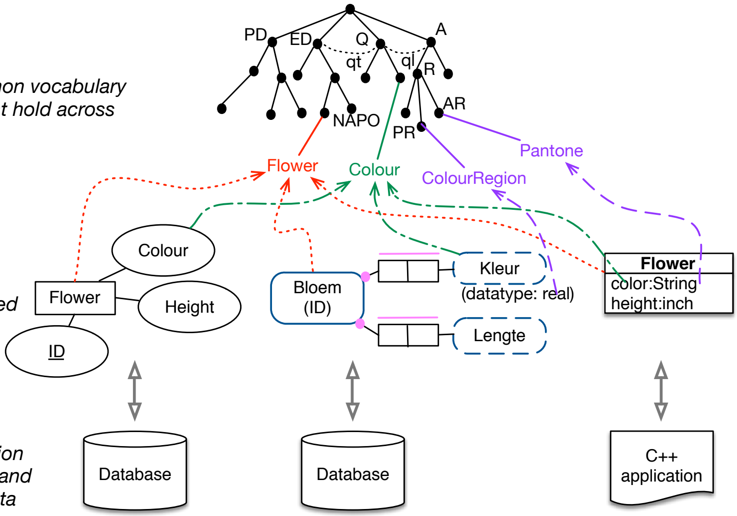
# Dissecting the title

- ‘real’ conceptual models vs ‘computational-conceptual’ models
  - Conceptual models *do not* have implementation decisions embedded in them
  - Some models do have some computational decisions; e.g., PK/FK, data types
- Difference(s) between conceptual models and ontologies (simplified/shorthand):
  - Conceptual models (in CS&IT) are application *dependent*
  - Ontologies are (in theory at least) application *independent*

**Ontology**
*provides the common vocabulary and constraints that hold across the applications*

**Conceptual model**
*shows what is stored in that particular application*

**Implementation**
*the actual information system that stores and manipulates the data*



# Dissecting the title

- How/where is ontology possibly useful for conceptual models and modelling?
  - Use ontology to improve the quality of a conceptual model
  - Reuse (part of) an ontology in a conceptual model
  - Use ontology to decide which language features should be available in a conceptual modelling language
  - Language ideally is logic-based so as to be (somewhat) precise

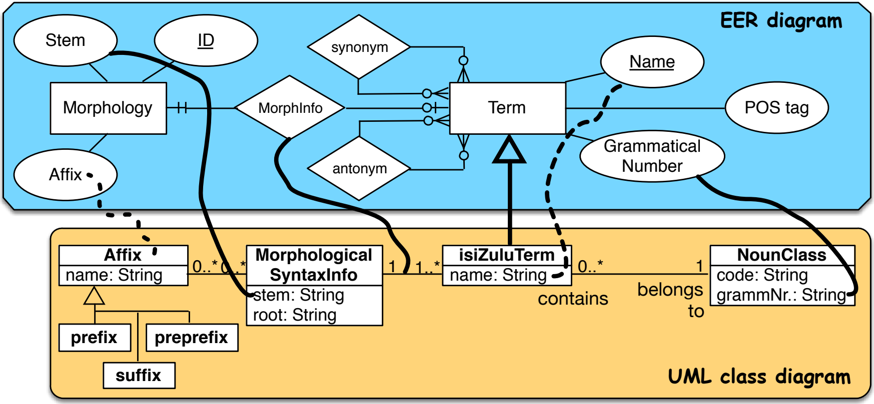
# What we will cover in the 3 sessions

1. Ontology
2. Languages for conceptual modelling
3. Temporal aspects (time permitting)

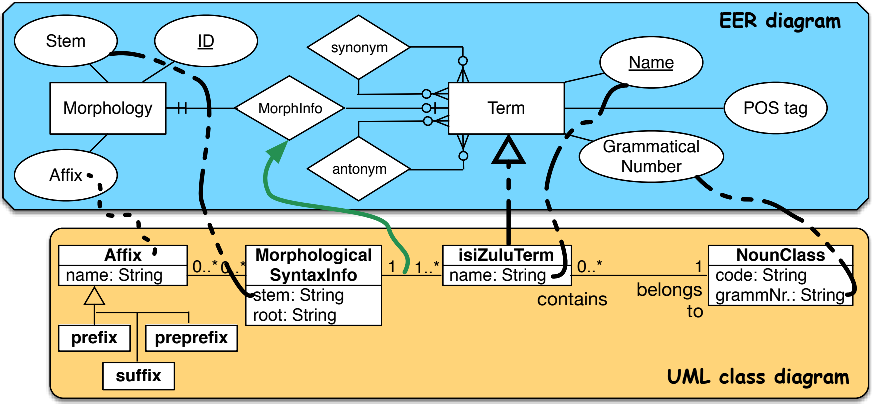
# Flavour of things to come: Ontology

- What are the core constructs (e.g., what's a relation?) and [how] does that affect the language?
- Do we need foundational ontology choices for modelling and if so, how?
- Modelling patterns—are some better than others, and if so: why?
- Refining aggregation/part-whole relations

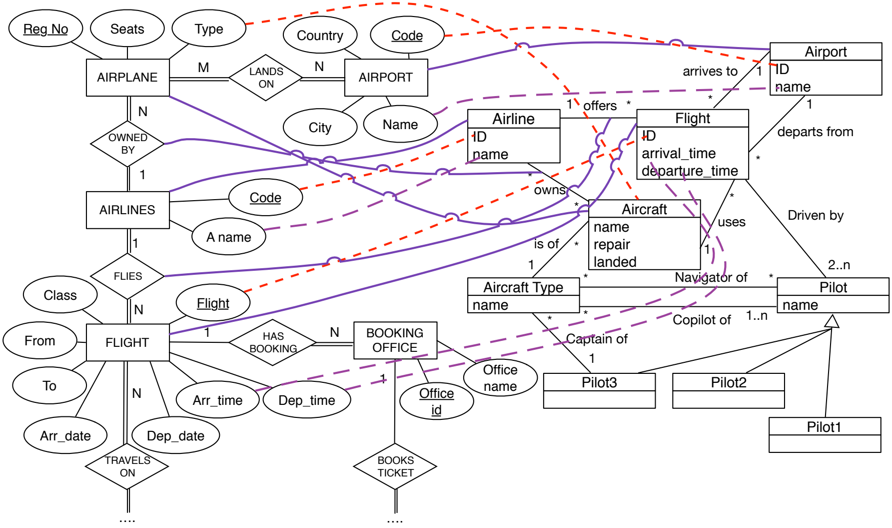
# Example scenario: isiZulu termbank (simplified)



# After logical and ontological analysis



# (still a small 'toy' example)



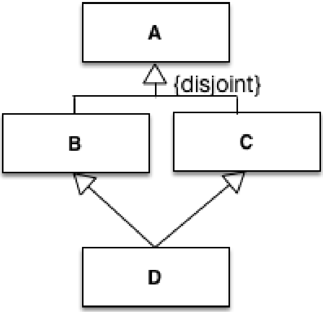



[Khan et al. (2016)]

# Flavour of things to come: language design

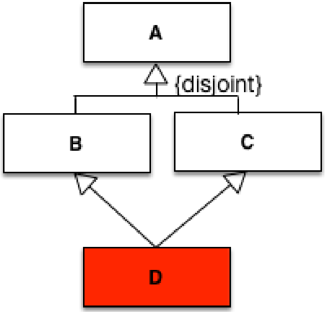
- How to give a formal semantics to the diagrams or controlled natural language?
- What does an ontologically well-founded logic (language) for conceptual modelling look like?
- What's the use of formalising it anyway?

# Conceptual data models–UML Class Diagram, inferences



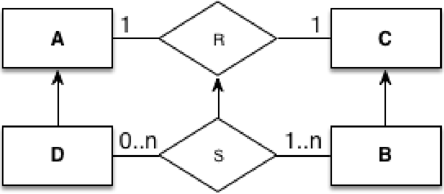

| | |
|---|---|
| $B \sqsubseteq A$ | $\forall x(B(x) \rightarrow A(x))$ |
| $C \sqsubseteq A$ | $\forall x(C(x) \rightarrow A(x))$ |
| $D \sqsubseteq B$ | $\forall x(D(x) \rightarrow B(x))$ |
| $D \sqsubseteq C$ | $\forall x(D(x) \rightarrow C(x))$ |
| $B \sqcap C \sqsubseteq \bot$ | $\forall x(B(x) \wedge C(x) \rightarrow \bot)$ |

# Conceptual data models–UML Class Diagram, inferences



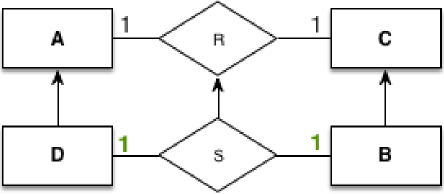

| | |
|---|---|
| $B \sqsubseteq A$ | $\forall x(B(x) \rightarrow A(x))$ |
| $C \sqsubseteq A$ | $\forall x(C(x) \rightarrow A(x))$ |
| $D \sqsubseteq B$ | $\forall x(D(x) \rightarrow B(x))$ |
| $D \sqsubseteq C$ | $\forall x(D(x) \rightarrow C(x))$ |
| $B \sqcap C \sqsubseteq \bot$ | $\forall x(B(x) \land C(x) \rightarrow \bot)$ |

# Conceptual data models–EER diagram, inferences




$R \sqsubseteq [i]A \sqcap [j]C$
$S \sqsubseteq [i]D \sqcap [j]B$
$A \sqsubseteq =1\,[i]R$
$C \sqsubseteq =1\,[j]R$
$B \sqsubseteq \exists[i]R$
$\forall x, y(R(x, y) \rightarrow A(x) \wedge C(y))$
$\forall x, y(S(x, y) \rightarrow D(x) \wedge B(y))$
$\forall x(A(x) \rightarrow \exists^{=1} y(R(x, y)))$
$\forall y(C(y) \rightarrow \exists^{=1} x(R(x, y)))$
$\forall y(B(y) \rightarrow \exists x(S(x, y)))$

# Conceptual data models–EER diagram, inferences




$R \sqsubseteq [i]A \sqcap [j]C$
$S \sqsubseteq [i]D \sqcap [j]B$
$A \sqsubseteq =1\,[i]R$
$C \sqsubseteq =1\,[j]R$
$B \sqsubseteq \exists[i]R$
$\forall x, y(R(x, y) \rightarrow A(x) \wedge C(y))$
$\forall x, y(S(x, y) \rightarrow D(x) \wedge B(y))$
$\forall x(A(x) \rightarrow \exists^{=1} y(R(x, y)))$
$\forall y(C(y) \rightarrow \exists^{=1} x(R(x, y)))$
$\forall y(B(y) \rightarrow \exists x(S(x, y)))$

# Typical *computational* usages for conceptual models

- **Reasoning over conceptual models to improve their quality**
  - With logic-based reconstructions in, and reasoners for, among others: DL [Artale et al.(2007b), Berardi et al.(2005), Keet(2009)]) and OWL [Wagih et al.(2013)], OCL [Queralt et al.(2012)], CLIF [Pan and Liu(2010)], Alloy [Braga et al.(2010)], Z [Jahangard Rafsanjani and Mirian-Hosseinabadi(2011)] (and many more)
- **Use of conceptual models during runtime**
  - Verification and validation [Cabot et al.(2008), Nizol et al.(2014)] (e.g., scalable test data generation [Smaragdakis et al.(2009)])
  - Designing [Bloesch and Halpin(1997)] and executing [Calvanese et al.(2010)] queries with the model's vocabulary; VQF/QBD [Soylu et al.(2017)]
  - Querying databases during the stage of query compilation [Toman and Weddell(2011)]
- Ontology-based data access and integration (tries both)

# Outline

# Elements in conceptual data models

- Class/Entity type
- Association/relationship/fact type, $n \geq 2$
- Attribute or Value Type
- One or more language specific elements, such as qualified association, aggregation association, objectified fact type
- Plethora of constraints

# Zooming in on the relations...

- **Poll**: are `teaches` and `taught by` two relations?

# Zooming in on the relations...

- **Poll**: are `teaches` and `taught by` two relations?
  - no... (more about that in the next slides)

# Zooming in on the relations...

- **Poll**: are `teaches` and `taught by` two relations?
  - no... (more about that in the next slides)
- **Poll**: How do you map UML's association ends (or ORM's roles) to an OWL object property (or vv.)?

# Zooming in on the relations...

- **Poll**: are `teaches` and `taught by` two relations?
  - no... (more about that in the next slides)
- **Poll**: How do you map UML's association ends (or ORM's roles) to an OWL object property (or vv.)?
  - Bit tricky, you have to make a modelling decision... (more about that later)

⇒ These two questions surface as a consequence of different ontological commitments as to what a relation or relationship really is (or what you're convinced of it is)

# A few more modelling questions for relations

- Should you introduce a minimum amount of properties, or many?
- Always (try to) declare domain and range axioms?
- Use explicit inverses (extending the vocabulary) or not?
- What about ternaries?
- How to find and fix mistakes and pitfalls?

# Outline

# A note from philosophy

- Relations investigated in philosophy
    - Nature and properties of some specific relations (parthood, portions, participation, causation)
    - ‘Categories’ of relations (material, formal)
    - Nature of relation itself (standard, positionalist, anti-positionalist)
- Some results more useful for ontologies and conceptual modelling than others, some even for tool development

# On relations

- Early ideas were put forward by [Williamson(1985)] and have been elaborated on and structured in [Fine(2000), van Inwagen(2006), Leo(2008), Cross(2002)]
- Three different ontological commitments about relations and relationships, which are, in Fine's [Fine(2000)] terminology, the *standard view*, the *positionalist*, and the *anti-positionalist* commitment

# The 'standard view' commitment

- Relies on linguistics and the English language in particular
- Take the fact *John loves Mary*, then one could be led to assume that *loves* is the name of the relation and *John* and *Mary* are the objects participating in the relation
- Then *Mary loves John* is not guaranteed to have the same truth value as the former fact—changing the verb does, i.e., *Mary is loved by John*
- We (seem to) have two relations, *loves* and its inverse *is loved by*

# Problems with the 'standard view' (1/2)

- For names *a* and *b*, *a loves b* holds iff what *a* denotes (in the reality we aim to represent) loves what *b* denotes.
- *John loves Mary* is not about language but about John loving Mary, so John and Mary are non-linguistic; cf. *'cabeza' translates into 'head'*

# Problems with the 'standard view' (1/2)

- For names *a* and *b*, *a loves b* holds iff what *a* denotes (in the reality we aim to represent) loves what *b* denotes.
- *John loves Mary* is not about language but about John loving Mary, so John and Mary are non-linguistic; cf. *'cabeza' translates into 'head'*
- Then, that John loves Mary and Mary is being loved by John refer to only one state of affairs between John and Mary
- Why should we want, let alone feel the need, to have *two relations* to describe it?

# Toward the 'positionalist' commitment

- Designate the two aforementioned facts to be relational expressions and not to let the verb used in the fact automatically also denote the name of the relation
- Then we can have many relational expressions standing in for the single relation that captures the state of affairs between John and Mary
- In analogy, we can have many relational expressions for one relationship at the type level

# Problems with the 'standard view' (2/2)

- Second, the specific order of the relation: changing the order does not mean the same for verbs that indicate an asymmetric relation; different for some other languages.

# Problems with the 'standard view' (2/2)

- Second, the specific order of the relation: changing the order does not mean the same for verbs that indicate an asymmetric relation; different for some other languages.
- Consider *John kills the dragon*. In Latin we have:
  *Johann*<u>*us*</u> *anguigen*<u>*am*</u> *caedit*, or
  *anguigen*<u>*am*</u> *caedit Johann*<u>*us*</u>, or
  *Johann*<u>*us*</u> *caedit anguigen*<u>*am*</u>,
  which all refer to the same state of affairs
- But *Johann*<u>*um*</u> *anguigen*<u>*a*</u> *caedit* is a different story alltogether
- Likewise for *John loves Mary* and *Johannus Mariam amat* versus *Johannum Maria amat*.

# Toward the 'positionalist' commitment

- A linguistic version of *argument places* (roles) thanks to the nominative and the accusative that are linguistically clearly indicated
- The order of the argument places is not relevant for the relation itself

# Toward the ‘positionalist’ commitment

- A linguistic version of *argument places* (roles) thanks to the nominative and the accusative that are linguistically clearly indicated
- The order of the argument places is not relevant for the relation itself
- English without such declensions that change the terms so as to disambiguate the meaning of a relational expression
- *Inverses for seemingly asymmetrical relations necessarily exist in reality and descriptions of reality in English, but not in other languages even when they represent the same state of affairs???*

# Toward the ‘positionalist’ commitment

- A linguistic version of *argument places* (roles) thanks to the nominative and the accusative that are linguistically clearly indicated
- The order of the argument places is not relevant for the relation itself
- English without such declensions that change the terms so as to disambiguate the meaning of a relational expression
- *Inverses for seemingly asymmetrical relations necessarily exist in reality and descriptions of reality in English, but not in other languages even when they represent the same state of affairs???*
- Asymmetric **relational expressions**, but this does not imply that the **relation** it verbalises is asymmetric

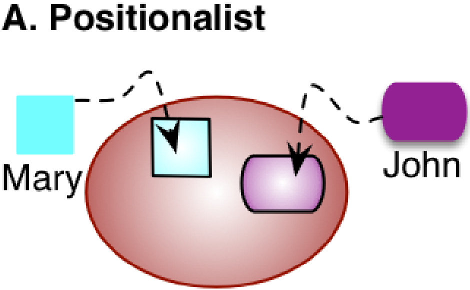
# The 'positionalist' commitment

- Binary relation *killing* and identify the argument places—"argument positions" [Fine(2000)] to have "distinguishability of the slots" [Cross(2002)]—*killer* and *deceased* (loosely, a place for the nominative and a place for the accusative), assign *John* to *killer* and *the dragon* to *deceased* and order the three elements in any arrangement
- Relation(ship) and several distinguishable 'holes' and we put each object in its suitable hole.

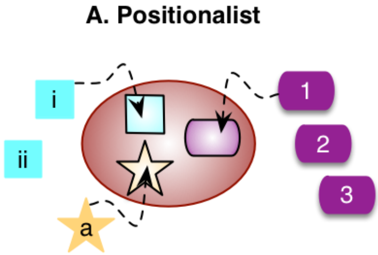
# The 'positionalist' commitment

- Binary relation *killing* and identify the argument places—"argument positions" [Fine(2000)] to have "distinguishability of the slots" [Cross(2002)]—*killer* and *deceased* (loosely, a place for the nominative and a place for the accusative), assign *John* to *killer* and *the dragon* to *deceased* and order the three elements in any arrangement
- Relation(ship) and several distinguishable 'holes' and we put each object in its suitable hole.
- There are no asymmetrical relations, because a relationship $R$ and its inverse $R^{-}$, or their instances, say, $r$ and $r'$, are *identical*, i.e., the same thing [Williamson(1985), Fine(2000), Cross(2002)]

# A conceptual view of the positionalist commitment–Mary&John/John&theDragon



# A conceptual view of the positionalist commitment–generally



# The 'positionalist' commitment

- Ingredients
  (i) an $n$-ary relationship $R$ with $A_1, \ldots, A_m$ participating object types ($m \leq n$),
  (ii) $n$ argument places $\pi_1, \ldots, \pi_n$, and
  (iii) $n$ assignments $\alpha_1, \ldots, \alpha_n$ that link each object $o_1, \ldots, o_n$ (each object instantiating an $A_i$) to an argument place ($\alpha \mapsto \pi \times o$)

# The 'positionalist' commitment

- Ingredients
  (i) an $n$-ary relationship $R$ with $A_1, \ldots, A_m$ participating object types ($m \leq n$),
  (ii) $n$ argument places $\pi_1, \ldots, \pi_n$, and
  (iii) $n$ assignments $\alpha_1, \ldots, \alpha_n$ that link each object $o_1, \ldots, o_n$ (each object instantiating an $A_i$) to an argument place ($\alpha \mapsto \pi \times o$)
- $R$, $\pi_1$, $\pi_2$, $\pi_3$, $r \in R$, $o_1 \in A_1$, $o_2 \in A_2$, $o_3 \in A_3$, then any of $\forall x, y, z(R(x, y, z) \rightarrow A_1(x) \wedge A_2(y) \wedge A_3(z))$ and its permutations with corresponding argument places—i.e., $R[\pi_1, \pi_2, \pi_3]$, and e.g., $R[\pi_2, \pi_1, \pi_3]$, and $[\pi_2 \pi_3]R[\pi_1]$—all denote the same SoA under the same assignment $o_1$ to $\pi_1$, $o_2$ to $\pi_2$, and $o_3$ to $\pi_3$ for the extension
- Thus, $r(o_1, o_2, o_3)$, $r(o_2, o_1, o_3)$, and $o_2 o_3 r o_1$ are different representations of the same SoA where objects $o_1$, $o_2$, and $o_3$ are related to each other by means of relation $r$.

# Problems with the ‘positionalist’ commitment

- From an ontological viewpoint, it requires identifiable argument positions to be part of the fundamental furniture of the universe.
- Then also in the signature of the formal language

# Problems with the 'positionalist' commitment

- From an ontological viewpoint, it requires identifiable argument positions to be part of the fundamental furniture of the universe.
- Then also in the signature of the formal language
- Symmetric relations and relationships, such as *adjacent to*, are problematic:
  i. Take $\pi_a$ and $\pi_b$ of a symmetric binary relation $r$, assign $o_1$ to position $\pi_a$ and $o_2$ to $\pi_b$ in state $s$.
  ii. One can do a reverse assignment of $o_1$ to position $\pi_b$ and $o_2$ to $\pi_a$ in state $s'$
  iii. But then $o_1$ and $o_2$ do not occupy the same positions as they did in $s$, so $s$ and $s'$ must be different, which should not be the case.

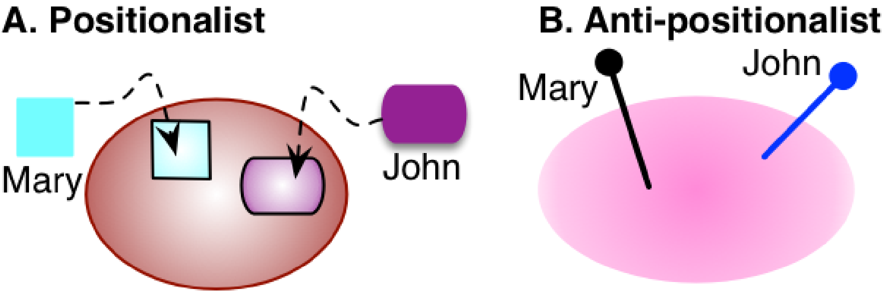
# The 'anti-positionalist' commitment

- No argument positions, but just a relation and objects that yield states by "combining" into "a single complex" [Fine(2000)]

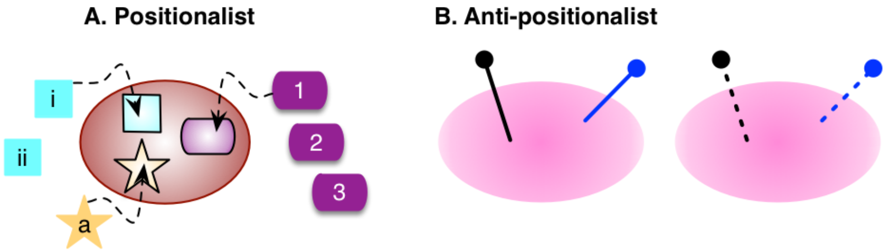
# The 'anti-positionalist' commitment

- No argument positions, but just a relation and objects that yield states by "combining" into "a single complex" [Fine(2000)]
- Solves the problems with the standard view
- Solves the positionalist's problem with symmetric relations
- But it needs a substitution relation
- (How to formalise this idea in a KR language is another problem)

# A conceptual view of positionalist and anti-positionalist–Mary&John/John&theDragon



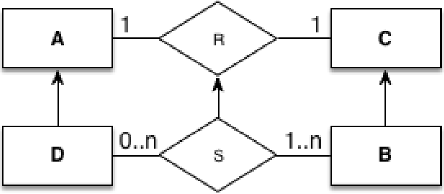
# A conceptual view of the positionalist and anti-positionalist–generally




- Note: UML Class Diagrams, ORM, ER all positionalist [Keet and Fillottrani(2013)], most of DL and FOL with standard view

# Exercise: Conceptual data models–EER diagram (again)

Task: Explain the contents of this slide




$R \sqsubseteq [i]A \sqcap [j]C$
$S \sqsubseteq [i]D \sqcap [j]B$
$A \sqsubseteq =1\,[i]R$
$C \sqsubseteq =1\,[j]R$
$B \sqsubseteq \exists[i]R$
$\forall x, y(R(x, y) \rightarrow A(x) \wedge C(y))$
$\forall x, y(S(x, y) \rightarrow D(x) \wedge B(y))$
$\forall x(A(x) \rightarrow \exists^{=1} y(R(x, y)))$
$\forall y(C(y) \rightarrow \exists^{=1} x(R(x, y)))$
$\forall y(B(y) \rightarrow \exists x(S(x, y)))$

# Outline

# What are the core elements in conceptual models?

- Exercise: name all language features of EER or of UML Class Diagrams, or ...

# What are the core elements in conceptual models?

- Exercise: name all language features of EER or of UML Class Diagrams, or ...
- e.g., both have attributes, but not in the same way
- ORM has value types; how does that differ in theory from the attributes, if at all?
- Which elements are present in non-CDMLs?

# What are the core elements in conceptual models?

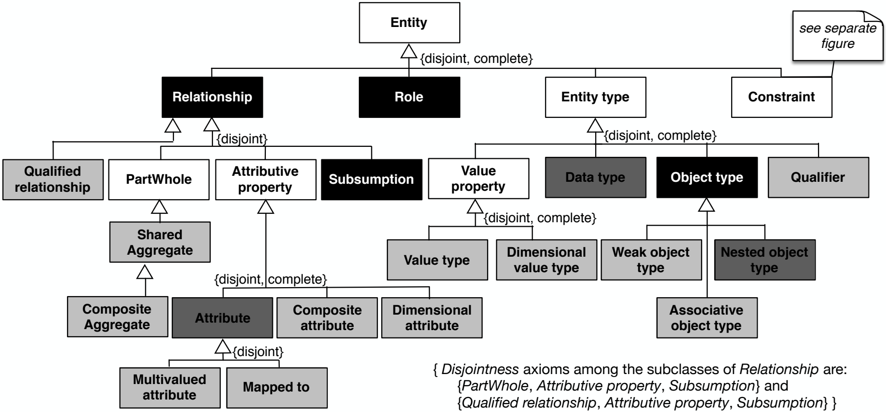
- Exercise: name all language features of EER or of UML Class Diagrams, or ...
- e.g., both have attributes, but not in the same way
- ORM has value types; how does that differ in theory from the attributes, if at all?
- Which elements are present in non-CDMLs?
- Let's first make an inventory of what we have in the (CDML) languages, then improve on that

# Metamodel: overview

- Captures all structural elements in the selected CDMLs [Keet and Fillottrani(2015)]
- Captures also their relations and constraints
- Describes the rules in which they may be combined
- The metamodel is designed in UML Class Diagram notation, formalized in FOL (precision) and OWL (practical usability)[1]

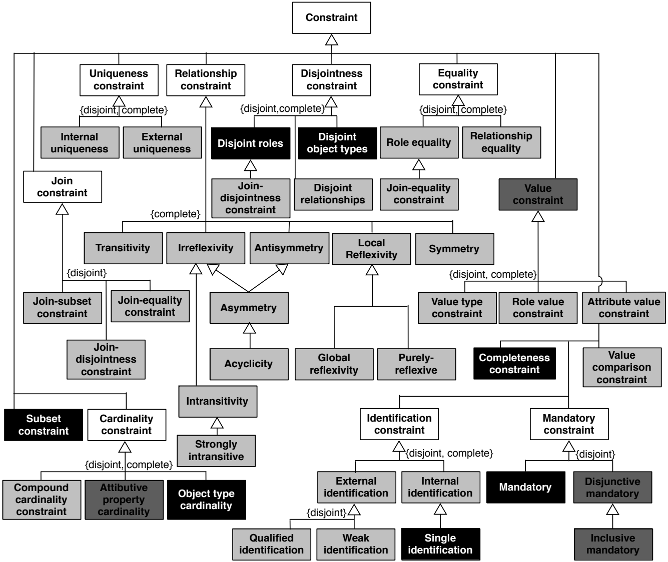
[1] Fillottrani, P.R., Keet, C.M.. *KF metamodel formalization*. Technical Report, Arxiv.org http://arxiv.org/abs/1412.6545. Dec 19, 2014. 26p.

# Static entities in EER, UML CD, ORM

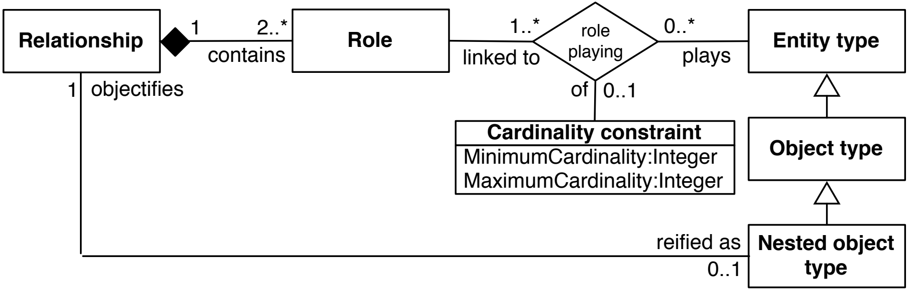



{ *Disjointness* axioms among the subclasses of *Relationship* are:
{*PartWhole*, *Attributive property*, *Subsumption*} and
{*Qualified relationship*, *Attributive property*, *Subsumption*} }

# Their constraints



# Selection of constraints between the entities



# Selection of constraints between them

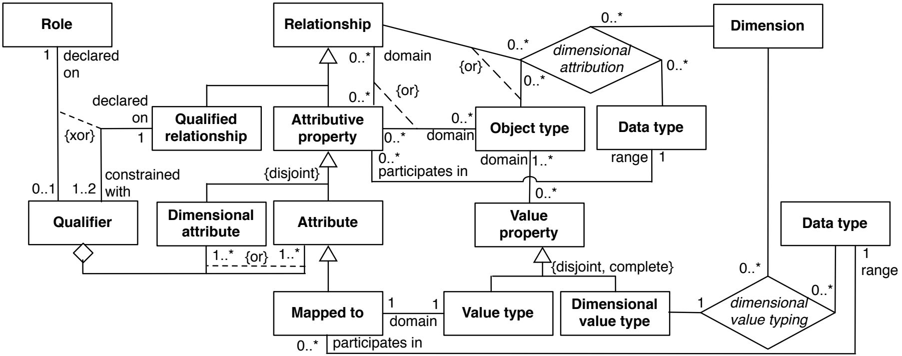



Figure: Metamodel fragment for value properties and simple attributes; Dimensional attribute is a reified version of the ternary relation dimensional attribution, and likewise for Dimensional value type and dimensional value typing.

# Selection of constraints between them

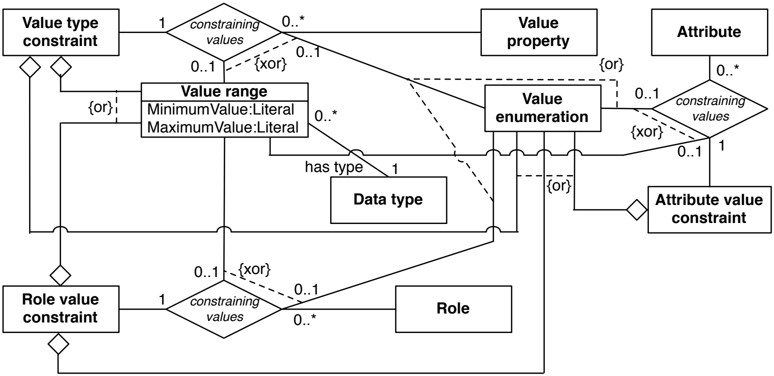



Figure: Value type, role, and attribute value constraints.

# Now back to those attributes and value types

- Structurally, they are different.
- What does Ontology say?

# Now back to those attributes and value types

- Structurally, they are different.
- What does Ontology say?
- First distinctions:

**Universalism**

$$\begin{tikzcd} a \arrow[r, "\text{inst}"] & F \\ b \arrow[ur, "\text{inst}"'] & \end{tikzcd}$$

**Trope theory**

$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\in"] \arrow[d, leftrightarrow, "\approx"'] & {|F|_\approx} \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\in"'] & \end{tikzcd}$$

**Universals+Tropes**

$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\text{inst}"] & F \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\text{inst}"'] & \end{tikzcd}$$

  - Universalism: $a$, $b$ instantiate $F$ (and $F$ is wholly present in $a$ and $b$)
  - Tropes: $a_F$ is the $F$-trope of $a$, inheres in $a$, and $F$ as equivalence class of resembling tropes
  - Merger: universalism adopted to classify the tropes

# Now back to those attributes and value types

- Structurally, they are different.
- What does Ontology say?
- First distinctions:

**Universalism**

$$\begin{tikzcd} a \arrow[r, "\text{inst}"] & F \\ b \arrow[ur, "\text{inst}"'] & \end{tikzcd}$$

**Trope theory**

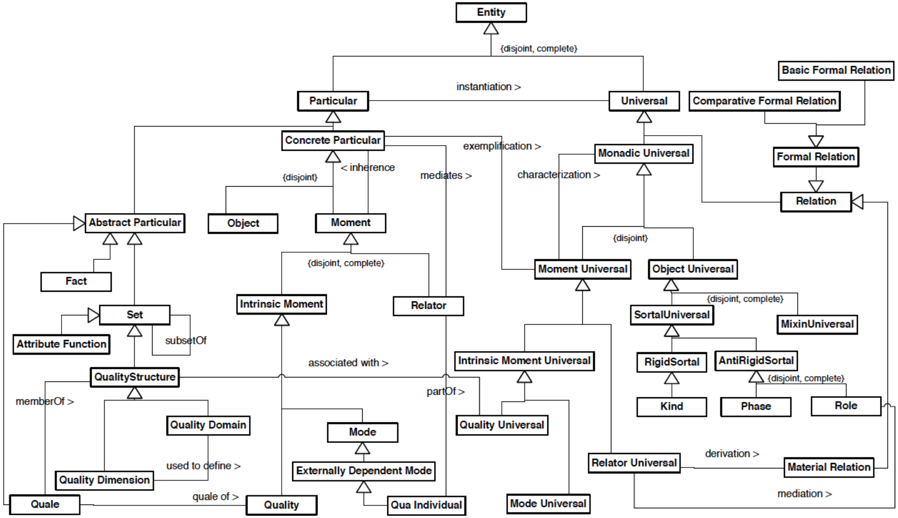
$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\in"] \arrow[d, leftrightarrow, "\approx"'] & {|F|_{\approx}} \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\in"'] & \end{tikzcd}$$

**Universals+Tropes**

$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\text{inst}"] & F \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\text{inst}"'] & \end{tikzcd}$$

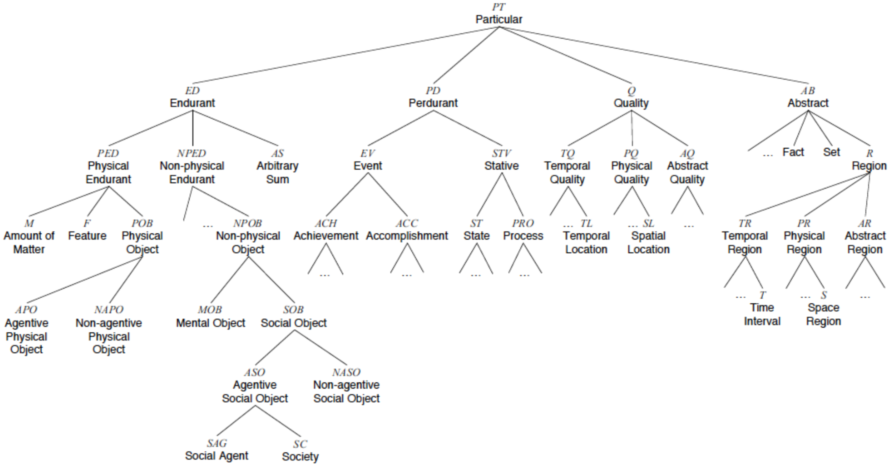
  - Universalism: $a$, $b$ instantiate $F$ (and $F$ is wholly present in $a$ and $b$)
  - Tropes: $a_F$ is the $F$-trope of $a$, inheres in $a$, and $F$ as equivalence class of resembling tropes
  - Merger: universalism adopted to classify the tropes
- Secondary distinctions: how they appear in a foundational ontology

# Example: UFO

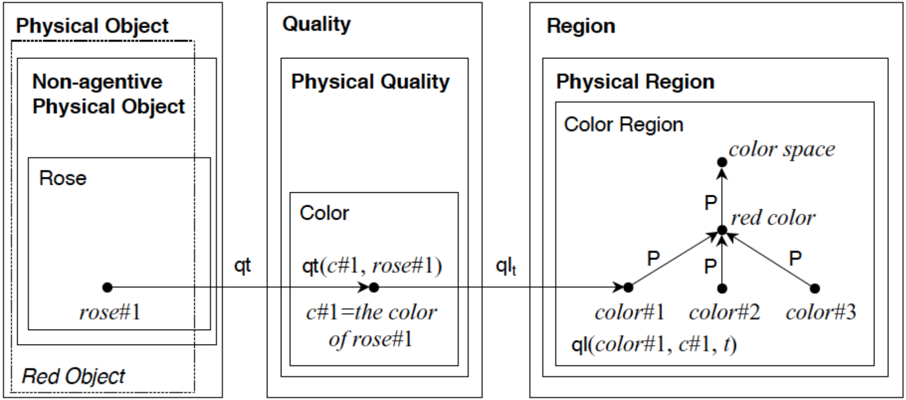


Entity
{disjoint, complete}
instantiation >
Particular
Universal
Basic Formal Relation
Comparative Formal Relation
Concrete Particular
exemplification >
Monadic Universal
Formal Relation
< inherence
{disjoint}
mediates >
characterization >
Relation
Abstract Particular
Object
Moment
{disjoint}
Fact
{disjoint, complete}
Moment Universal
Object Universal
Intrinsic Moment
Relator
{disjoint, complete}
Set
SortalUniversal
MixinUniversal
Attribute Function
subsetOf
associated with >
Intrinsic Moment Universal
RigidSortal
AntiRigidSortal
QualityStructure
{disjoint, complete}
partOf >
memberOf >
Kind
Phase
Role
Quality Domain
Mode
Quality Universal
derivation >
used to define >
Relator Universal
Material Relation
Externally Dependent Mode
Quality Dimension
mediation >
quale of >
Quale
Quality
Qua Individual
Mode Universal

# Outline of DOLCE categories



# Example: DOLCE's basic relations w.r.t. qualities



# Exercise: what do others say about attributes/qualities?

- BFO
- GFO
- SUMO
- Yamato

# Exercise: what do others say about attributes/qualities?

- BFO
  - .
- GFO
  - .
- SUMO
  - .
- Yamato
  - .

# Various commitments regarding ‘attributes’

- ‘attribute’ (attribution, quality) is a unary entity;
    - e.g., UFO: trope theory; DOLCE: universalism
    - choice of foundational ontology affects what we (assume to) have in our conceptual model

# Various commitments regarding ‘attributes’

- ‘attribute’ (attribution, quality) is a unary entity;
  - e.g., UFO: trope theory; DOLCE: universalism
  - choice of foundational ontology affects what we (assume to) have in our conceptual model
- attribute is a binary relation between class & data type
  - e.g., OWL’s `DataProperty`; UML’s attribute
  - ignores foundational ontologies

# Various commitments regarding 'attributes'

- 'attribute' (attribution, quality) is a unary entity;
    - e.g., UFO: trope theory; DOLCE: universalism
    - choice of foundational ontology affects what we (assume to) have in our conceptual model
- attribute is a binary relation between class & data type
    - e.g., OWL's `DataProperty`; UML's attribute
    - ignores foundational ontologies
- Trade-offs
    - More compact notation with attributes
    - Modelling is based on arbitrary (practical, application) decisions
        - increases chance of incompatibilities across diagrams
        - less reusable within and across models
        - instability of model

# Outline

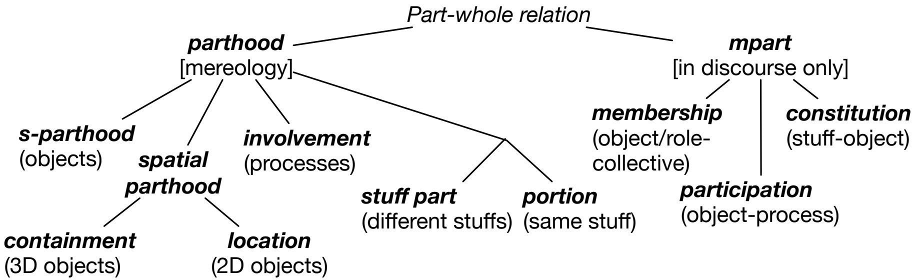
# UML's aggregation, part-whole relations, and mereology

- Lots of fun problems, widely investigated
- Converged to a set of common part-whole relations for conceptual modelling
- Which I'd like to cover, but there are already two talks about mereology at ISAO'18

# UML's aggregation, part-whole relations, and mereology

- Lots of fun problems, widely investigated
- Converged to a set of common part-whole relations for conceptual modelling
- Which I'd like to cover, but there are already two talks about mereology at ISAO'18
- Therefore, for now:
  - Only the next slide as a summary (based on [Keet and Artale(2008)])
  - Turns out the 'common' ones may not be that common (paper at FOIS'18, [Keet and Khumalo(2018)])

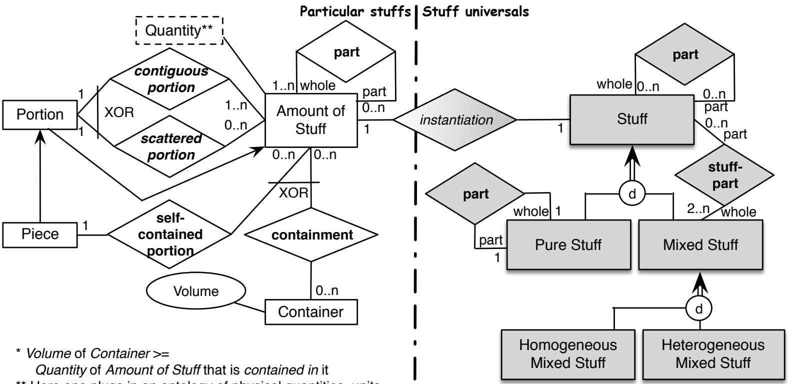
# Common part-whole relations on conceptual modelling (informally)



# Refining 'object type'—somehow

- OntoClean and hierarchies in CDMs; e.g., OntoUML
- Stuff and quantities
  - e.g., need to design a model for a database for tracking food stuffs
  - Ingredients, quantities, masses, amounts of matter, ....
  - Can a FO help you with that? if so, which one(s)?

# Stuff (informally)

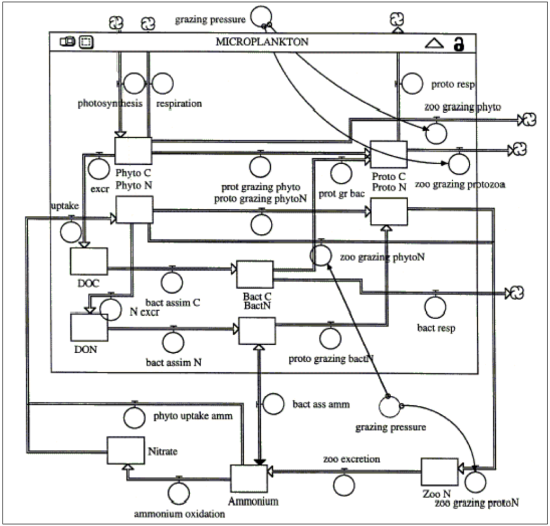



* *Volume* of *Container* >= *Quantity* of *Amount of Stuff* that is *contained in* it
** Here one plugs in an ontology of physical quantities, units, and measurements

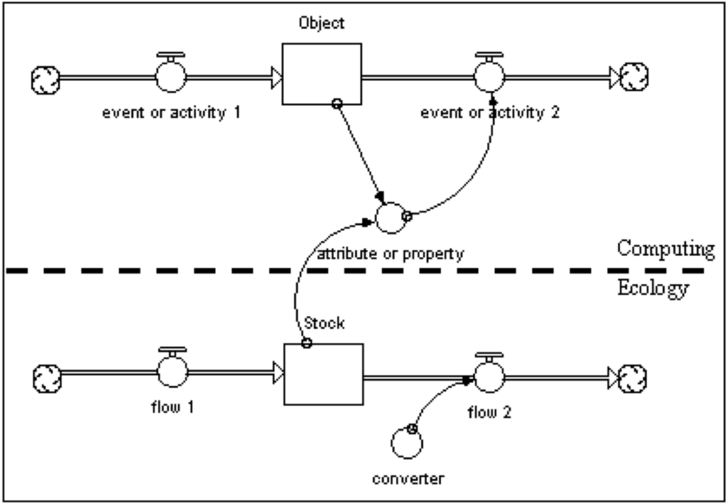
*(more details in [Keet (2016)])*

# Outline

# Conceptual models/diagrams in biology



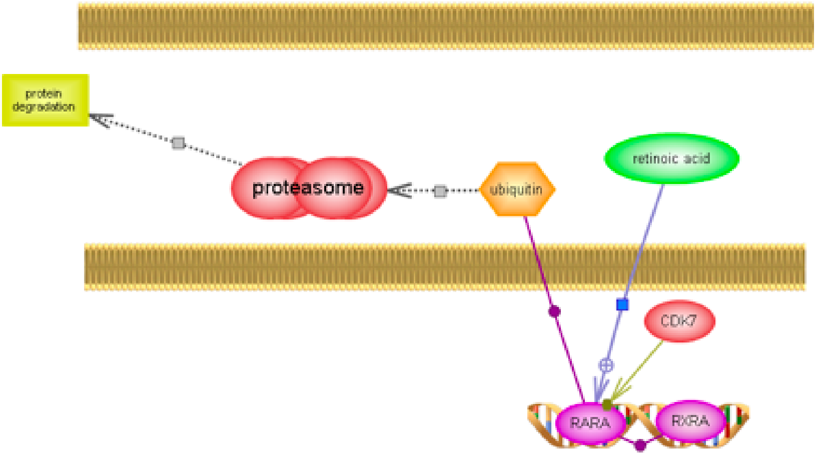
# The main elements in STELLA)



# From STELLA model to ontology

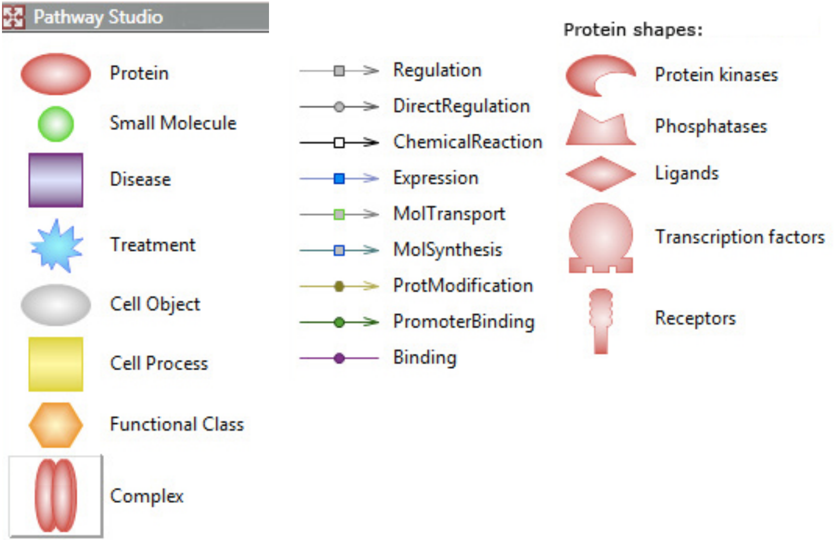
- Key aspects in the ecological model:
  - A Stock correspond to a noun (particular or universal)
  - Flow to verb
  - Converter to attribute related to Flow or Stock
  - Action Connector relates the former
- How could that map to elements in ontologies?
  - Object is candidate for an endurant
  - Event_or_activity for a method or perdurant
  - Converter an attribute or property
  - Action Connector candidate for relationship between any two of Flow, Stock and Converter
- Analysis and details in [Keet(2005)]

# Another diagram, in Pathway Studio's notation




(Degradation of the RAR and RXR by the proteasome)

# Pathway Studio's legend



# Guidance in this process?

- Methodology for conceptual models: from **Di**agram to **D**omain **On**tology, **DiDOn** [Keet(2012)]
- ONSET to compare FOs (ontological commitments, content, practical) `http://www.meteck.org/files/onset/` [Khan and Keet(2012)]

# Guidance in this process?

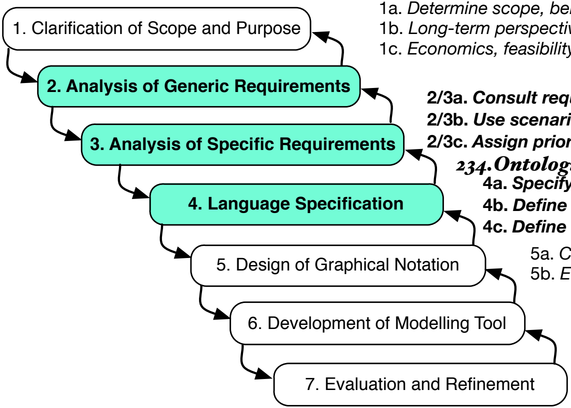
- Methodology for conceptual models: from **Di**agram to **D**omain **On**tology, **DiDOn** [Keet(2012)]
- ONSET to compare FOs (ontological commitments, content, practical) `http://www.meteck.org/files/onset/` [Khan and Keet(2012)]
- How to use the FO with the conceptual (data) model?
  - UML stereotypes
  - 'subclassing' the FO
  - Design new language with additional constructs (formal semantics with a many-sorted logic)

# Outline

# Outline

# Guidance on language design

- A logic/language can be seen as a ‘product’/solution that solves a problem
- In analogy of other products: is there a development process, with requirements to meet etc.?
- No methodology for design of a logic
- There is one for design of Domain Specific languages (DSLs) [Frank(2013)]
- Adapt that for our purpose




1a. *Determine scope, benefits*
1b. *Long-term perspective*
1c. *Economics, feasibility*

**2/3a. *Consult requirements catalogue***
**2/3b. *Use scenarios***
**2/3c. *Assign priorities***

***234. Ontological analysis of language features***

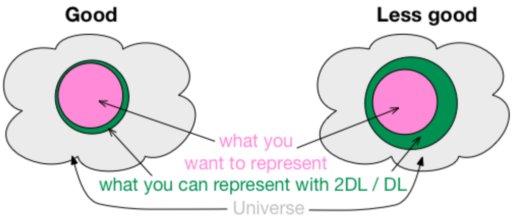
**4a. *Specify syntax and semantics***
**4b. *Define glossary***
**4c. *Define metamodel***

5a. *Create sample diagrams*
5b. *Evaluate notation*

7a. *Test cases*
7b. *Analyse against requirements*
7c. *Analyse effect of use against current practice*

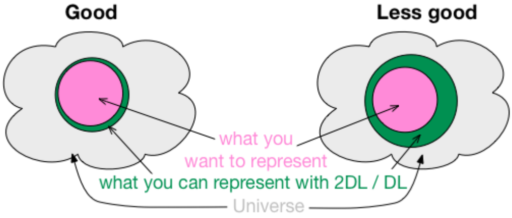
# "234. Ontological analysis of language features"

- Affordances and features of the logic concern:
  - Ability to represent the conceptualisation/reality more or less precisely with more or less constraints; e.g.
    - $Human \sqsubseteq \exists hasPart.Eye$ or $Human \sqsubseteq = 2\ hasPart$ (OWL DL)
    - $Human \sqsubseteq = 2\ hasPart.Eye$ (OWL 2 DL)



# "234. Ontological analysis of language features"

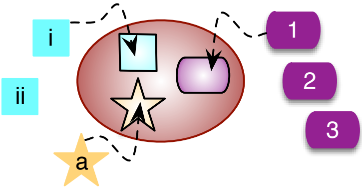
- Affordances and features of the logic concern:
  - Ability to represent the conceptualisation/reality more or less precisely with more or less constraints; e.g.
    - $Human \sqsubseteq \exists hasPart.Eye$ or $Human \sqsubseteq = 2\ hasPart$ (OWL DL)
    - $Human \sqsubseteq = 2\ hasPart.Eye$ (OWL 2 DL)




  - whether the language contributes to support, or even shape, the conceptualisation and one's analysis for the conceptual (data) model, or embeds certain philosophical assumptions and positions

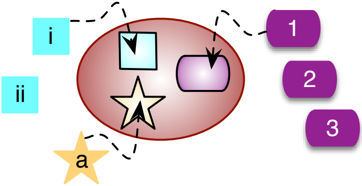
# Choices – ontology

- Whether the roles that objects play are fundamental components of relationships (positionalist) or not (standard view); i.e.: if roles should be elements of the language; e.g.
  - $\exists teaches \sqsubseteq Course$ and $\exists teaches^{-} \sqsubseteq Prof$ (most DLs, FOL)
  - $teach \sqsubseteq [lect]Prof \sqcap [taught]Course$ ($\mathcal{DLR}$ family, DBs)



# Choices – ontology

- Whether the roles that objects play are fundamental components of relationships (positionalist) or not (standard view); i.e.: if roles should be elements of the language; e.g.
  - $\exists teaches \sqsubseteq Course$ and $\exists teaches^{-} \sqsubseteq Prof$ (most DLs, FOL)
  - $teach \sqsubseteq [lect]Prof \sqcap [taught]Course$ ($\mathcal{DLR}$ family, DBs)




- 4D view on the world (space-time worms) or 3D objects with optional temporal extension
- Inherent vagueness (rough, fuzzy), or the world is crisp

# Choices – (im)precision in elements

- Whether refinements on the kinds of general elements—that then have their own representation element—would result in a different (better) conceptual model. e.g.:
  - Add element for aggregation or parthood (in addition to not just Relationship and subsumption)
  - not just Object type but also, say, sortal with rigid property ($\forall x \phi(x) \rightarrow \Box \phi(x)$) or class with anti-rigid property ($\forall x \phi(x) \rightarrow \neg \Box \phi(x)$), with stereotypes or separate graphical elements
  - If binary relationships only (cf. *n*-aries), would the modeller would assume there are only binaries in the world?

# Choices – (im)precision in elements

- Whether refinements on the kinds of general elements—that then have their own representation element—would result in a different (better) conceptual model. e.g.:
  - Add element for aggregation or parthood (in addition to not just Relationship and subsumption)
  - not just Object type but also, say, sortal with rigid property ($\forall x \phi(x) \rightarrow \Box \phi(x)$) or class with anti-rigid property ($\forall x \phi(x) \rightarrow \neg \Box \phi(x)$), with stereotypes or separate graphical elements
  - If binary relationships only (cf. *n*-aries), would the modeller would assume there are only binaries in the world?
- 'truly conceptual' or or also somewhat computational; i.e., to represent only *what* vs. *what* & *how*
  - data types of attributes (UML) or not (ER), with attribute being $A \mapsto C \times \texttt{Datatype}$

# Choices – (im)precision in elements

- Whether refinements on the kinds of general elements—that then have their own representation element—would result in a different (better) conceptual model. e.g.:
  - Add element for aggregation or parthood (in addition to not just Relationship and subsumption)
  - not just Object type but also, say, sortal with rigid property ($\forall x \phi(x) \rightarrow \Box \phi(x)$) or class with anti-rigid property ($\forall x \phi(x) \rightarrow \neg \Box \phi(x)$), with stereotypes or separate graphical elements
  - If binary relationships only (cf. *n*-aries), would the modeller would assume there are only binaries in the world?
- 'truly conceptual' or or also somewhat computational; i.e., to represent only *what* vs. *what & how*
  - data types of attributes (UML) or not (ER), with attribute being $A \mapsto C \times \texttt{Datatype}$
- What should be named?

# Outline

# The choices in UML, ER, ORM

- Ontology: positionalist, 3D, crisp world
- Features: *n*-aries, UML with aggregation, just object types, ER no datatypes
- Data showed that UML has disproportionally
  - fewer *n*-aries (look across is ambiguous)
  - more aggregation (if the construct is there, modellers see it everywhere?)

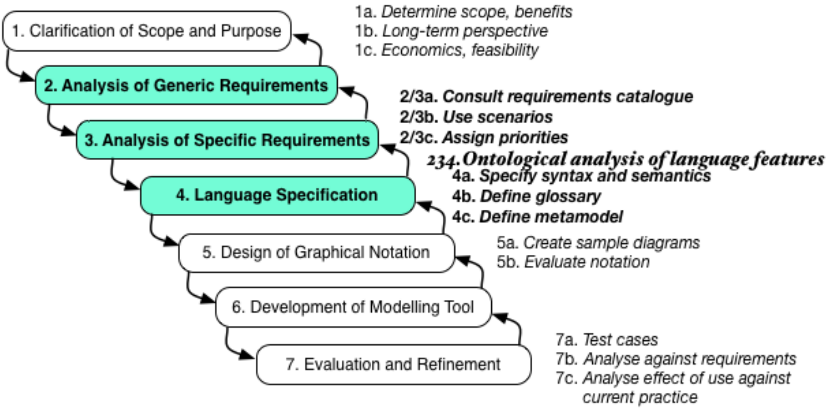
Table: Popular logics for logic-based reconstructions of CDMLs assessed against a set of requirements (1/2).

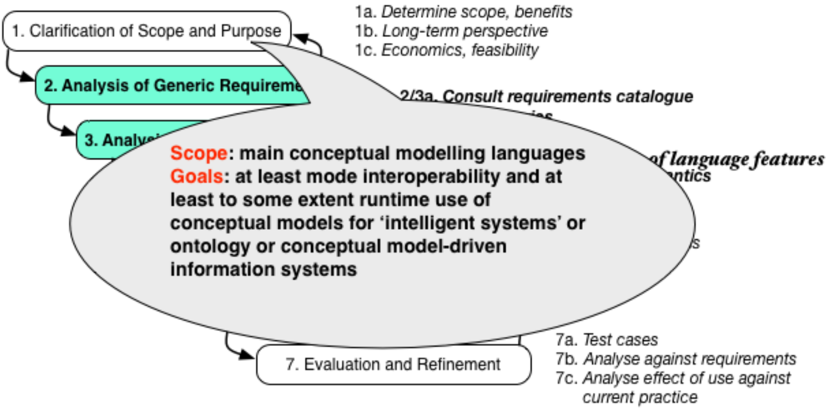
| DL-Lite$_{\mathcal{A}}$ | $\mathcal{DLR}_{ifd}$ | OWL 2 DL | FOL |
|---|---|---|---|
| *Language features* | | | |
| – standard view | + positionalist | – standard view | – standard view |
| – with datatypes | – with datatypes | – with datatypes | + no datatypes |
| – no parthood primitive | – no parthood primitive | – no parthood primitive | – no parthood primitive |
| – no *n*-aries | + with *n*-aries | – no *n*-aries | + with *n*-aries |
| + 3D | + 3D | + 3D | + 3D |
| – very few features; large feature mismatch | + little feature mismatch | ± some feature mismatch, with overlapping sets | + little feature mismatch |
| – formalisation to complete | + formalisation exist | – formalisation to complete | ± formalisation exist |

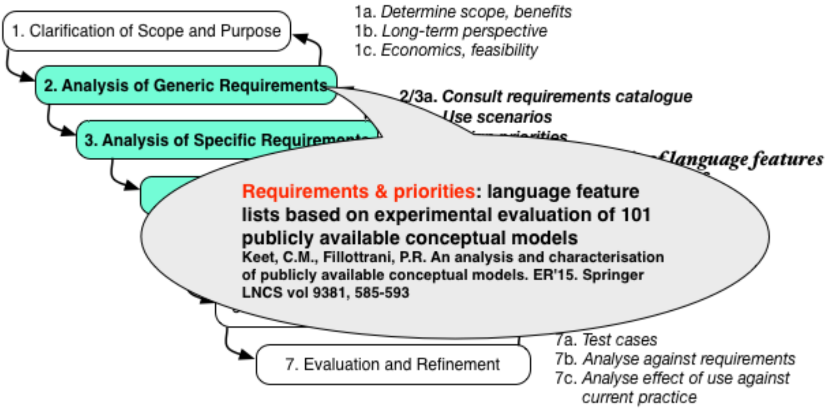
Table: Popular logics for logic-based reconstructions of CDMLs assessed against a set of requirements (2/2).

| **DL-Lite$_{\mathcal{A}}$** | $\mathcal{DLR}_{ifd}$ | **OWL 2 DL** | **FOL** |
|---|---|---|---|
| | *Computation and implementability* | | |
| + PTIME (TBox); $AC^0$ (ABox) | ± EXPTIME-complete | ± N2EXPTIME-complete | – undecidable |
| + very scalable (TBox and ABox) | ± somewhat scalable (TBox) | ± somewhat scalable (TBox) | – not scalable |
| + several reasoners | – no implementation | + several reasoners | – few reasoners |
| + linking with ontologies doable | – no interoperability | + linking with ontologies doable | – no interoperability with existing infrastructures |
| + 'integration' with OntoIOP | – no integration with OntoIOP | + 'integration' with OntoIOP | + 'integration' with OntoIOP |
| + modularity infrastructure | – modularity infrastructure | + modularity infrastructure | – modularity infrastructure |

# Outline

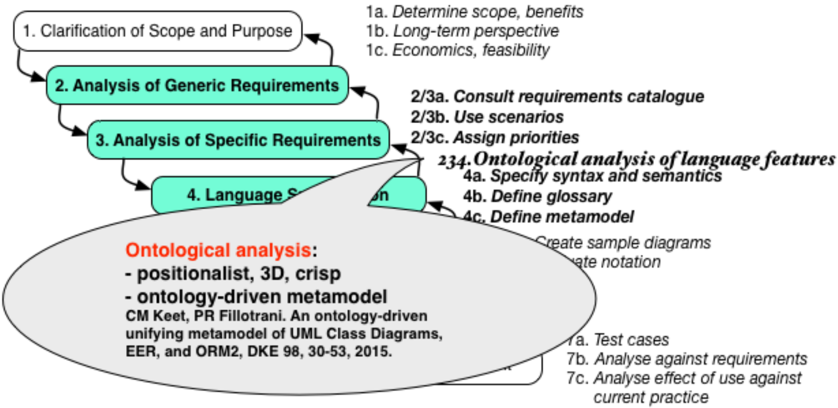

1. Clarification of Scope and Purpose
1a. Determine scope, benefits
1b. Long-term perspective
1c. Economics, feasibility
2. Analysis of Generic Requirements
3. Analysis of Specific Requirements
2/3a. Consult requirements catalogue
2/3b. Use scenarios
2/3c. Assign priorities
234. Ontological analysis of language features
4. Language Specification
4a. Specify syntax and semantics
4b. Define glossary
4c. Define metamodel
5. Design of Graphical Notation
5a. Create sample diagrams
5b. Evaluate notation
6. Development of Modelling Tool
7. Evaluation and Refinement
7a. Test cases
7b. Analyse against requirements
7c. Analyse effect of use against current practice

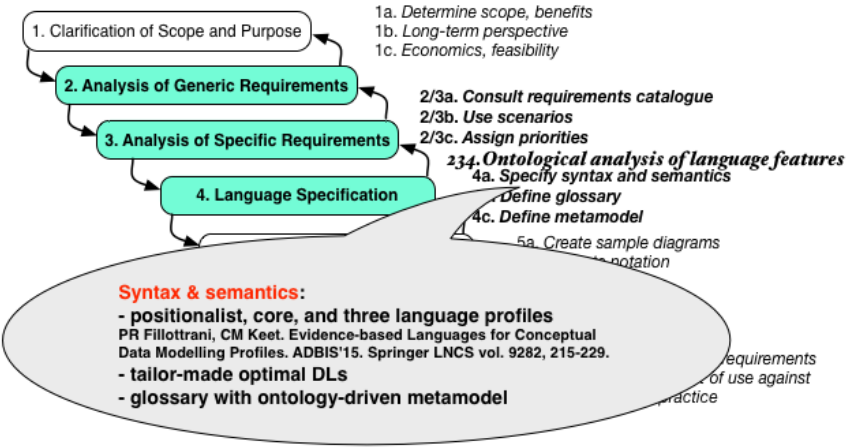

1. Clarification of Scope and Purpose
1a. Determine scope, benefits
1b. Long-term perspective
1c. Economics, feasibility
2. Analysis of Generic Requireme
2/3a. Consult requirements catalogue
3. Analy
of language features
ntics
Scope: main conceptual modelling languages
Goals: at least mode interoperability and at least to some extent runtime use of conceptual models for 'intelligent systems' or ontology or conceptual model-driven information systems
7. Evaluation and Refinement
7a. Test cases
7b. Analyse against requirements
7c. Analyse effect of use against current practice




**Requirements & priorities: language feature lists based on experimental evaluation of 101 publicly available conceptual models**

**Keet, C.M., Fillottrani, P.R. An analysis and characterisation of publicly available conceptual models. ER'15. Springer LNCS vol 9381, 585-593**


1. Clarification of Scope and Purpose
1a. Determine scope, benefits
1b. Long-term perspective
1c. Economics, feasibility
2. Analysis of Generic Requirements
2/3a. Consult requirements catalogue
2/3b. Use scenarios
2/3c. Assign priorities
3. Analysis of Specific Requirements
234. Ontological analysis of language features
4a. Specify syntax and semantics
4b. Define glossary
4c. Define metamodel
Create sample diagrams
ate notation
a. Test cases
7b. Analyse against requirements
7c. Analyse effect of use against current practice
Ontological analysis:
- positionalist, 3D, crisp
- ontology-driven metamodel
CM Keet, PR Fillotrani. An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2, DKE 98, 30-53, 2015.


1. Clarification of Scope and Purpose
1a. Determine scope, benefits
1b. Long-term perspective
1c. Economics, feasibility
2. Analysis of Generic Requirements
2/3a. Consult requirements catalogue
2/3b. Use scenarios
2/3c. Assign priorities
3. Analysis of Specific Requirements
234. Ontological analysis of language features
4a. Specify syntax and semantics
4. Language Specification
Define glossary
4c. Define metamodel
5a. Create sample diagrams
Syntax & semantics:
- positionalist, core, and three language profiles
PR Fillottrani, CM Keet. Evidence-based Languages for Conceptual Data Modelling Profiles. ADBIS'15. Springer LNCS vol. 9282, 215-229.
- tailor-made optimal DLs
- glossary with ontology-driven metamodel

# Logic foundation for profiles

- How to formalise the diagrams in which logic?

$\Rightarrow$ *Which DL (or other logic) is most appropriate, and why?*

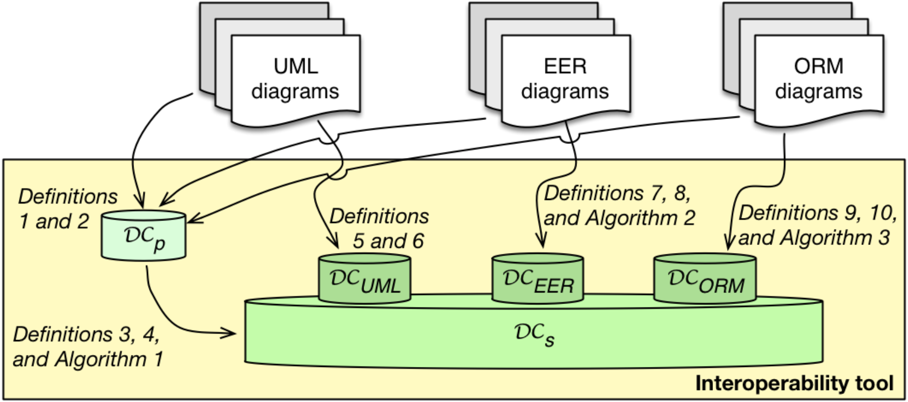
$\Rightarrow$ Analyse contents of publicly available conceptual data models [Fillottrani and Keet(2015)]

- Try as high a coverage of the most used features

# Considerations in the formalisation

- Positionalist relations and relationships complicates formalisation (computationally more costly), and implementation ($\mathcal{DLR}$ has one very much proof-of-concept implementation [Calvanese et al.(2011)])
- Did both positionalist and standard core, with algorithm

# Orchestration of profiles and algorithms



## Definition (Positionalist core profile)

Given a conceptual model in any of the analysed CDMLs, we construct a *knowledge base* in $\mathcal{DC}_p$ by applying the rules:

- we take the set all of object types $A$, binary relationships $P$, datatypes $T$ and attributes $a$ in the model as the basic elements in the knowledge base.
- for each binary relationship $P$ formed by object types $A$ and $B$, we add to the knowledge base the assertions $\geq 1[1]P \sqsubseteq A$ and $\geq 1[2]P \sqsubseteq B$.
- for each attribute $a$ of datatype $T$ within an object type $A$, including the transformation of ORM's Value Type following the rule given in [Fillottrani and Keet(2014)], we add the assertion $A \sqsubseteq \exists a.T \sqcap \leq 1a$.
- subsumption between two object types $A$ and $B$ is represented by the assertion $A \sqsubseteq B$.

*Continues on next slide....*

## Definition (Positionalist core profile)

Given a conceptual model in any of the analysed CDMLs, we construct a *knowledge base* in $\mathcal{DC}_p$ by applying the rules:

... *continued from previous slide*

- for each object type cardinality $m..n$ in relationship $P$ with respect to its $i$-th component $A$, we add the assertions $A \sqsubseteq \leq n[i]P \sqcap \geq m[i]P$.
- we add for each mandatory constraints of a concept $A$ in a relationship $P$ the axiom $A \sqsubseteq \geq 1[1]P$ or $A \sqsubseteq \geq 1[2]P$ depending on the position played by $A$ in $P$. This is a special case of the previous one, with $n = 1$.
- for each single identification in object type $A$ with respect to an attribute $a$ of datatype $T$ we add the axiom $\mathrm{id}\, A\, a$.

# Positionalist Core profile in DL syntax

$\mathcal{DC}_p$ can be represented by the following DL syntax. Starting from atomic elements, we can construct binary relations $R$, arbitrary concepts $C$ and axioms $X$ according to the rules:

$$
\begin{aligned}
C &\longrightarrow \top \mid A \mid \leq k[i]R \mid \geq k[i]R \mid \forall a.T \mid \exists a.T \mid \leq 1\, a \mid C \sqcap D \\
R &\longrightarrow \top_2 \mid P \mid (i : C) \\
X &\longrightarrow C \sqsubseteq D \mid \mathrm{id}\; C\; a
\end{aligned}
$$

where $i = 1, 2$ and $0 < k$. For convenience of presentation, we use the numbers 1 and 2 to name the role places, but they can be any number or string and do not impose an order.

# Positionalist Core profile in DL, semantics (1/2)

### Definition

An $\mathcal{DC}_p$ *interpretation* $\mathcal{I} = (\cdot^{\mathcal{I}}_{\mathcal{C}}, \cdot^{\mathcal{I}}_{T}, \cdot^{\mathcal{I}})$ for a knowledge base in $\mathcal{DC}_p$ consists of a set of objects $\Delta^{\mathcal{I}}_{C}$, a set of datatype values $\Delta^{\mathcal{I}}_{T}$, and a function $\cdot^{\mathcal{I}}$ satisfying the constraints shown in Table 3. It is said that $\mathcal{I}$ *satisfies* the assertion $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; and it *satisfies* the assertion id $C$ $a$ iff exists $T$ such that $C^{\mathcal{I}} \subseteq (\exists a.T \sqcap \leq 1a)^{\mathcal{I}}$ (mandatory 1) and for all $v \in T^{\mathcal{I}}$ it holds that $\#\{c | c \in C^{\mathcal{I}} \wedge (c, v) \in a^{\mathcal{I}}\} \leq 1$ (inverse functional).

# Positionalist Core profile in DL, semantics (2/2)

Table: Semantics of $\mathcal{DC}_p$.

$$
\begin{array}{c}
\top^{\mathcal{I}} \subseteq \Delta_C^{\mathcal{I}} \\
A^{\mathcal{I}} \subseteq \top^{\mathcal{I}} \\
\top_2^{\mathcal{I}} = \top^{\mathcal{I}} \times \top^{\mathcal{I}} \\
P^{\mathcal{I}} \subseteq \top_2^{\mathcal{I}} \\
T^{\mathcal{I}} \subseteq \Delta_T^{\mathcal{I}} \\
a^{\mathcal{I}} \subseteq \top^{\mathcal{I}} \times \Delta_T^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}
\end{array}
\qquad
\begin{array}{c}
(\leq k[i]R)^{\mathcal{I}} = \{c \in \Delta_C^{\mathcal{I}} | \#\{(d_1, d_2) \in R^{\mathcal{I}}.d_i = c\} \leq k\} \\
(\geq k[i]R)^{\mathcal{I}} = \{c \in \Delta_C^{\mathcal{I}} | \#\{d_1, d_2) \in R^{\mathcal{I}}.d_i = c\} \geq k\} \\
(\exists a.T)^{\mathcal{I}} = \{c \in \Delta_C^{\mathcal{I}} | \exists b \in T^{\mathcal{I}}.(c, b) \in a^{\mathcal{I}}\} \\
(\forall a.T)^{\mathcal{I}} = \{c \in \Delta_C^{\mathcal{I}} | \forall v \in \Delta_T^{\mathcal{I}}.(c, v) \in a^{\mathcal{I}} \to v \in T^{\mathcal{I}}\} \\
(\leq 1\, a)^{\mathcal{I}} = \{c \in \Delta_C^{\mathcal{I}} | \#\{(c, v) \in a^{\mathcal{I}}\} \leq 1\} \\
(i : C)^{\mathcal{I}} = \{(d_1, d_2) \in \top_2^{\mathcal{I}} | d_i \in C^{\mathcal{I}}\}
\end{array}
$$

# Some observations

- All the entities in the core profile sum up to 87.57% of the entities in all the analysed models, covering 91,88% of UML models, 73.29% of ORM models, and 94.64% of ER/EER models
- Excluded due to their low incidence in the model set (despite overlap): Role (DL role component) and Relationship (DL role) Subsumption, and Completeness and Disjointness constraints

# Some observations

- All the entities in the core profile sum up to 87.57% of the entities in all the analysed models, covering 91,88% of UML models, 73.29% of ORM models, and 94.64% of ER/EER models
- Excluded due to their low incidence in the model set (despite overlap): Role (DL role component) and Relationship (DL role) Subsumption, and Completeness and Disjointness constraints
- No completeness and disjointness, so reasoning is quite simple
- Can code negation only with cardinality constraints [Baader et al.(2008), chapter 3], but then we need to reify each negated concept as a new idempotent role, which is not possible to get from the $\mathcal{DC}_p$ rules
- Can embed $\mathcal{DC}_p$ into $\mathcal{DLR}$, but latter is more expressive than needed

# Standard core profile

- Convert $\mathcal{DC}_p$ into a standard core, $\mathcal{DC}_s$

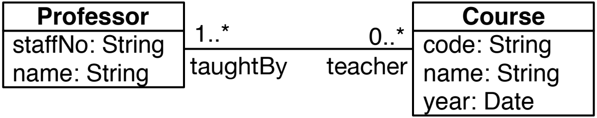
## Definition

Given a conceptual model in any of the analysed CDMLs, we construct a *knowledge based* in $\mathcal{DC}_s$ by applying Algorithm 1 to its $\mathcal{DC}_p$ knowledge base.

- With inverse relations to keep connected both relationships generated by reifying roles
- DL syntax approximation (noting construction rules from $\mathcal{DC}_p$):

$$C \longrightarrow \top_1 \,|\, A \,|\, \forall R.A \,|\, \exists R.A \,|\, \leq k\, R \,|\, \geq k\, R \,|\, \forall a.T \,|\, \exists a.T \,|\, \leq 1\, a.T \,|\, C \sqcap D$$
$$R \longrightarrow \top_2 \,|\, P \,|\, P^-$$
$$X \longrightarrow C \sqsubseteq D \,|\, \mathrm{id}\; C\; a$$

# Positionalist to standard choices




*teacher* and *taughtBy* are named association ends, not a name of the association (DL role). Options to formalise it:

- make each association end a DL role, `teacher` and `taughtBy`, then choose:
  - declare them inverse of each other with $\texttt{teacher} \equiv \texttt{taughtBy}^-$
  - do not declare them inverses
- 'bump up' either *teacher* or *taughtBy* to DL role, and use the other through a direct inverse and do not extend vocabulary with the other (`teacher` and $\texttt{teacher}^-$ cf. adding also `taughtBy`)

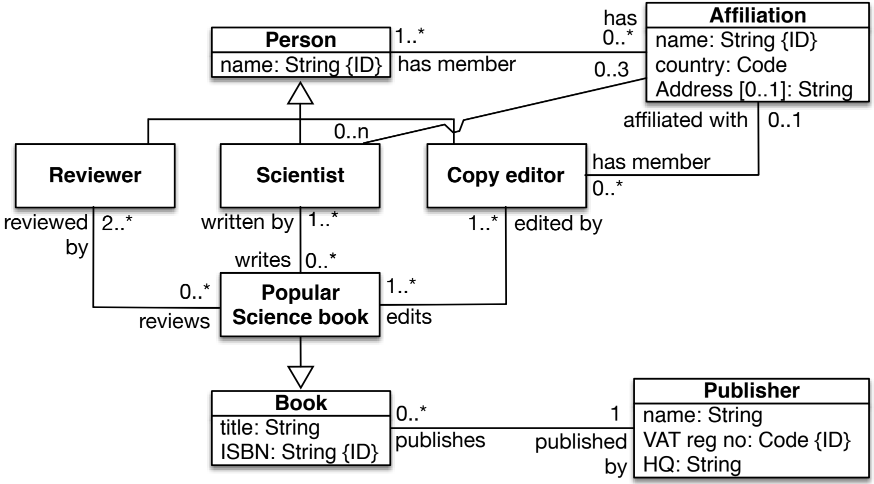
**Algorithm 1** *Positionalist Core to Standard Core*

$P$ an atomic binary relationship; $D_P$ domain of $P$; $R_P$ range of $P$
**if** $D_P \neq R_P$ **then**
  Rename $P$ to two 'directional' readings, $Pe_1$ and $Pe_2$
  Make $Pe_1$ and $Pe_2$ a DL relation (role)
  Type the relations with $\top \sqsubseteq \forall Pe_1.D_P \sqcap \forall Pe_1^-.R_P$
  Declare inverses with $Pe_1 \equiv Pe_2^-$
**else**
  **if** $D_P = R_P$ **then**
    **if** $i = 1, 2$ is named **then**
      $Pe_i \leftarrow i$
    **else**
      $Pe_i \leftarrow$ user-added label or auto generated label
    **end if**
    Make $Pe_i$ a DL relation (role)
    Type one $Pe_i$, i.e., $\top \sqsubseteq \forall Pe_i.D_P \sqcap \forall Pe_i^-.R_P$
    Declare inverses with $Pe_i \equiv Pe_2^-$
  **end if**
**end if**

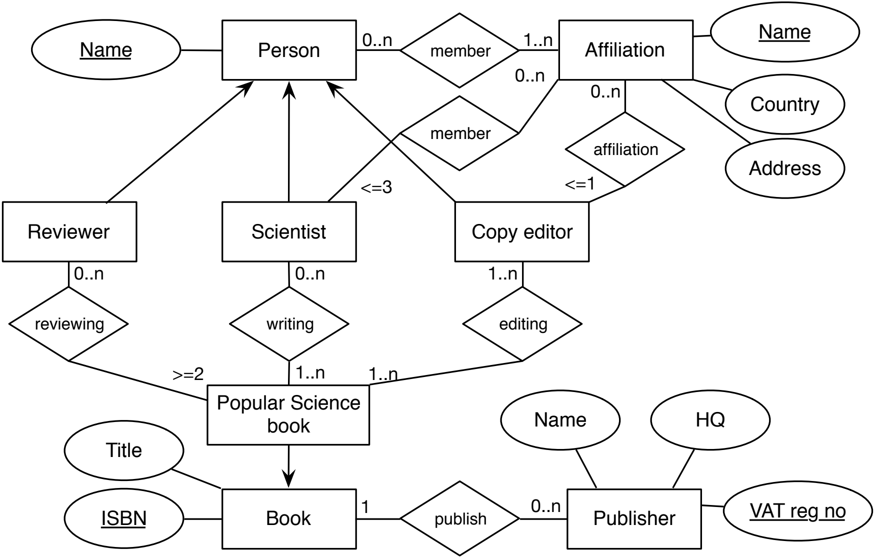
# Some observations on $\mathcal{DC}_s$

- Simple, too
- Main reasoning problem still class subsumption and equivalence
- At most the DL $\mathcal{ALNI}$ (called $\mathcal{PL}_1$ in [Donini et al.(1991)])
- $\mathcal{PL}_1$ has polynomial subsumption; data complexity unknown
- Tweaking with interaction between role inclusions and number restrictions, and UNA: $DL\text{-}Lite_{core}^{(\mathcal{HN})}$ (NLOGSPACE)
- As aside: adding class disjointness, then reduction to $DL\text{-}Lite_{bool}^{(\mathcal{HN})}$ (NP-hard) [Artale et al.(2009)]

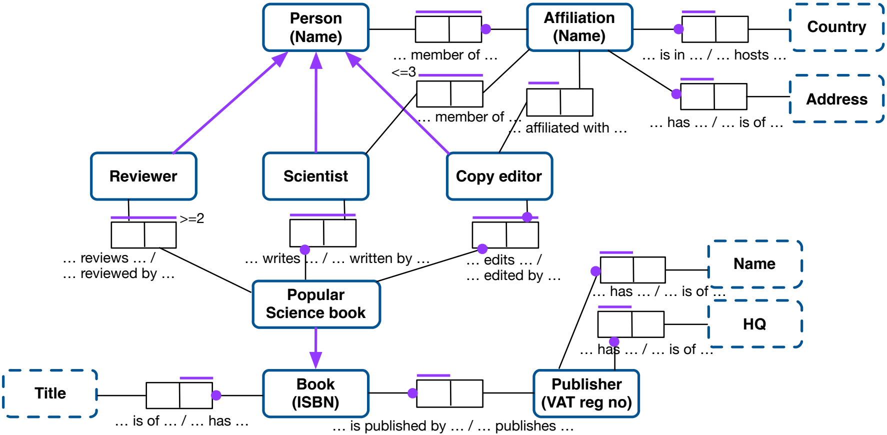
# Sample diagrams using all $\mathcal{DC}_s$ features



# Sample diagrams using all $\mathcal{DC}_s$ features



# Sample diagrams using all $\mathcal{DC}_s$ features



# Or as business rules (fragment shown)

- Each popular science book is reviewed by at least 2 reviewers.
- Each reviewer may review a popular science book.
- Each book must be published by exactly one publisher.
- Each publisher has one HQ.

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p+$ Algorithm 1)
- Obtain set of OTs (`{Person, ...}`) `and DTs ({Name, ...})`

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p$+ Algorithm 1)
- Obtain set of OTs ({`Person, ...`}) and DTs ({`Name, ...`})
- For Relationships, use Algorithm 1:
    1. bump up the association end names to DL roles
    2. type the relationships with:

$$\top \sqsubseteq \forall \texttt{has\_member.Affiliation} \sqcap \forall \texttt{has\_member}^{-}.\texttt{Person}$$

$$\top \sqsubseteq \forall \texttt{has.Person} \sqcap \forall \texttt{has}^{-}.\texttt{Affiliation}$$

    3. declare inverses, $\texttt{has\_member} \equiv \texttt{has}^{-}$

Repeat for each association in UML diagram

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p$+ Algorithm 1)
- Obtain set of OTs ({Person, ...}) and DTs ({Name, ...})
- For Relationships, use Algorithm 1:
  1. bump up the association end names to DL roles
  2. type the relationships with:

$$\top \sqsubseteq \forall \texttt{has\_member}.\texttt{Affiliation} \sqcap \forall \texttt{has\_member}^{-}.\texttt{Person}$$

$$\top \sqsubseteq \forall \texttt{has}.\texttt{Person} \sqcap \forall \texttt{has}^{-}.\texttt{Affiliation}$$

  3. declare inverses, $\texttt{has\_member} \equiv \texttt{has}^{-}$

  Repeat for each association in UML diagram
- Step 3 of $\mathcal{DC}_p$ definition: attributes. e.g., for Person's Name:

$$\texttt{Person} \sqsubseteq \exists \texttt{Name}.\texttt{String} \sqcap \leq 1 \ \texttt{Name}$$

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p$+ Algorithm 1)
- Obtain set of OTs ({`Person, ...`}) and DTs ({`Name, ...`})
- For Relationships, use Algorithm 1:
  1. bump up the association end names to DL roles
  2. type the relationships with:

$$\top \sqsubseteq \forall \texttt{has\_member}.\texttt{Affiliation} \sqcap \forall \texttt{has\_member}^-.\texttt{Person}$$

$$\top \sqsubseteq \forall \texttt{has}.\texttt{Person} \sqcap \forall \texttt{has}^-.\texttt{Affiliation}$$

  3. declare inverses, $\texttt{has\_member} \equiv \texttt{has}^-$

  Repeat for each association in UML diagram
- Step 3 of $\mathcal{DC}_p$ definition: attributes. e.g., for `Person`'s `Name`:

$$\texttt{Person} \sqsubseteq \exists \texttt{Name}.\texttt{String} \sqcap \leq 1\ \texttt{Name}$$

- Step 4: subsumptions; e.g., $\texttt{Popular\_science\_book} \sqsubseteq \texttt{Book}$

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p$+ Algorithm 1)
- Obtain set of OTs ({Person, ...}) and DTs ({Name, ...})
- For Relationships, use Algorithm 1:
  1. bump up the association end names to DL roles
  2. type the relationships with:

$$\top \sqsubseteq \forall \texttt{has\_member}.\texttt{Affiliation} \sqcap \forall \texttt{has\_member}^{-}.\texttt{Person}$$

$$\top \sqsubseteq \forall \texttt{has}.\texttt{Person} \sqcap \forall \texttt{has}^{-}.\texttt{Affiliation}$$

  3. declare inverses, $\texttt{has\_member} \equiv \texttt{has}^{-}$

  Repeat for each association in UML diagram
- Step 3 of $\mathcal{DC}_p$ definition: attributes. e.g., for Person's Name:

$$\texttt{Person} \sqsubseteq \exists \texttt{Name}.\texttt{String} \sqcap \leq 1\ \texttt{Name}$$

- Step 4: subsumptions; e.g., $\texttt{Popular\_science\_book} \sqsubseteq \texttt{Book}$
- Step 5 and 6: cardinalities. e.g. $\texttt{Affiliation} \sqsubseteq \geq 1\ \texttt{has\_member}$

# Steps UML diagram to $\mathcal{DC}_s$

- (Recall $\mathcal{DC}_s$ is obtained from $\mathcal{DC}_p$+ Algorithm 1)
- Obtain set of OTs ({Person, ...}) and DTs ({Name, ...})
- For Relationships, use Algorithm 1:
  1. bump up the association end names to DL roles
  2. type the relationships with:

$$\top \sqsubseteq \forall \texttt{has\_member}.\texttt{Affiliation} \sqcap \forall \texttt{has\_member}^{-}.\texttt{Person}$$

$$\top \sqsubseteq \forall \texttt{has}.\texttt{Person} \sqcap \forall \texttt{has}^{-}.\texttt{Affiliation}$$

  3. declare inverses, $\texttt{has\_member} \equiv \texttt{has}^{-}$

  Repeat for each association in UML diagram
- Step 3 of $\mathcal{DC}_p$ definition: attributes. e.g., for Person's Name:

$$\texttt{Person} \sqsubseteq \exists \texttt{Name}.\texttt{String} \sqcap \leq 1 \ \texttt{Name}$$

- Step 4: subsumptions; e.g., $\texttt{Popular\_science\_book} \sqsubseteq \texttt{Book}$
- Step 5 and 6: cardinalities. e.g. $\texttt{Affiliation} \sqsubseteq \geq 1 \ \texttt{has\_member}$
- Finally, identifiers; e.g. ISBN for Book, adding $\mathrm{id}\,\texttt{Book}\,\texttt{ISBN}$ to the $\mathcal{DC}_s$ knowledge base

# Profile comparison on language and complexity

| Profile | Main features | Approx. DL | Subsumption complexity |
|---|---|---|---|
| $\mathcal{DC}_p$ | positionalist, binary relationships, identifiers, cardinality constraints, attribute typing, mandatory attribute and its functionality | $\mathcal{DLR}$ | EXPTIME |
| $\mathcal{DC}_s$ | standard view, binary relationships, inverses | $\mathcal{ALNI}$ | P |
| $\mathcal{DC}_{UML}$ | relationship subsumption, attribute cardinality | $DL\text{-}Lite^{\mathcal{HN}}_{core}$ | NLOGSPACE |
| $\mathcal{DC}_{EER}$ | ternary relationships, attribute cardinality, external keys | $DL\text{-}Lite^{\mathcal{N}}_{core}$ | NLOGSPACE |
| | | $\mathcal{CFD}$ | P |
| $\mathcal{DC}_{ORM}$ | entity type disjunction, relationships complement, relationship subsumption, complex identifiers ('multi attribute keys') | $\mathcal{DLR}_{ifd}$ | EXPTIME |
| | | $\mathcal{CFDI}^{\forall-}_{nc}$ | P |

# Discussion

- ‘Uninteresting’ logics for automated reasoning over conceptual models
- But

# Discussion

- 'Uninteresting' logics for automated reasoning over conceptual models
- But
- assuming that also the reconstructions of $\mathcal{DC}_p$ and $\mathcal{DC}_{ORM}$ will be lower than EXPTIME (tbd),
- They're good/excellent for use of conceptual models during runtime; e.g.:

# Discussion

- 'Uninteresting' logics for automated reasoning over conceptual models
- But
- assuming that also the reconstructions of $\mathcal{DC}_p$ and $\mathcal{DC}_{ORM}$ will be lower than EXPTIME (tbd),
- They're good/excellent for use of conceptual models during runtime; e.g.:
  - Scalable test data generation [Smaragdakis et al.(2009)]
  - Designing [Bloesch and Halpin(1997)] and executing [Calvanese et al.(2010)] queries with the model's vocabulary
  - Querying databases during the stage of query compilation [Toman and Weddell(2011)]

# Language design

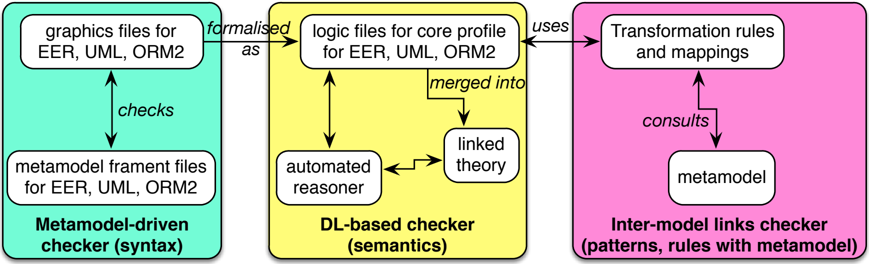
- First attempt to scope and structure the the logic design process, with ontological considerations
- Can do with a broader systematic investigation on alternative design choices and their consequences
- Identified alternate choices effectively addressed by multiple compatible profiles with algorithms for conversions

# Language design

- First attempt to scope and structure the the logic design process, with ontological considerations
- Can do with a broader systematic investigation on alternative design choices and their consequences
- Identified alternate choices effectively addressed by multiple compatible profiles with algorithms for conversions
- 'good' logic
  - matches the implicit ontological commitments
  - that fits needs here is 'less good' in precision
  - turns out to be a *family* of compatible logics + algorithms

# Toward applicability

- Profiles may be applied as back-end of CASE tool, OBDA
- Will allow modeller to model in their graphical notation of choice, yet be compatible with the rest
- Transformations and inter-model assertions of approximate entities and of modelling patterns [Fillottrani and Keet(2014), Khan et al.(2016), Fillottrani and Keet(2017)]



# Exercise: design your own (two options)

- Informal $\rightarrow$ formal
  - Take some graphical modelling language (e.g., flowcharts) or a CNL (e.g., Simplified English; see also CNL 2018 paper)
  - Examine the elements ontologically
  - Design a logic for it
- Requirements $\rightarrow$ language
  - Consider some task or thing (e.g., student enrolment process, event management)
  - Devise requirements for the language to be able to model such tasks/things
  - Design a language for it (logic/diagram notation/CNL)

# Outline

# Outline

# Examples

- CDM: 'RentalCar must be returned *before* Deposit is reimbursed'
- CDM: 'Employee will receive a bonus after 2 years of employment'

# Examples

- CDM: 'RentalCar must be returned *before* Deposit is reimbursed'
- CDM: 'Employee will receive a bonus after 2 years of employment'
- Domain ontology: 'Biopsy, *planned*' (in SNOMED CT)
- Domain ontology: 'HairLoss *during* the treatment Chemotherapy'; Butterfly is a transformation of (*used to be*) Caterpillar.

# Examples

- CDM: ‘RentalCar must be returned *before* Deposit is reimbursed’
- CDM: ‘Employee will receive a bonus after 2 years of employment’
- Domain ontology: ‘Biopsy, *planned*’ (in SNOMED CT)
- Domain ontology: ‘HairLoss *during* the treatment Chemotherapy’; Butterfly is a transformation of (*used to be*) Caterpillar.
- a brain is an essential part of a human (for the entire human’s lifetime)
- a boxer’s hands are essential parts of the boxer (for as long as he’s a boxer)

# What are the main choices regarding time?

1. Annotation model 'about time' vs reasoning over temporal knowledge
2. The main options ontologically w.r.t. the latter:

3. The representation

# What are the main choices regarding time?

1. Annotation model 'about time' vs reasoning over temporal knowledge
2. The main options ontologically w.r.t. the latter:
   - Chronons (successive points) vs dense time
   - Linear vs branching time
   - 3-dimensional objects + time vs 4-dimensional 'space-time worms'
3. The representation

# What are the main choices regarding time?

1. Annotation model 'about time' vs reasoning over temporal knowledge
2. The main options ontologically w.r.t. the latter:
   - Chronons (successive points) vs dense time
   - Linear vs branching time
   - 3-dimensional objects + time vs 4-dimensional 'space-time worms'
3. The representation
   - Add '$t$'; e.g., $R(x, y, t)$ "$R$ holds between $x$ and $y$ at time $t$" and 4-D fluents/$n$-ary approach
   - Temporal logic; include constructors in the language, e.g. $\Diamond^{+}$ "at some time in the future" (cf. "$\exists$ "some"), availing of $\mathcal{S}$ince and/or $\mathcal{U}$ntil operators

# Additions to CDMLs

- Mostly linear time (makes more sense in a database setting, cf CTL in formal methods)
- Chronons (fits more easily with snapshots of databases)
- 3D most popular, with a few extensions, notably to ER and EER

# Outline

# Example: TREND with $\mathcal{DLR}_{\mathcal{US}}$

- Approach the same as before:
  - Graphical notation
  - Pick a logic or design one
  - Do the logic-based reconstruction
- The very expressive (well, undecidable) $\mathcal{DLR}_{\mathcal{US}}$ [Artale et al.(2002)]
- $ER_{VT}$ extended into $EER_{VT}^{++}$ and now $\mathrm{TREND}$

# $\mathcal{DLR}_{\mathcal{US}}$ (the essence of it)

- $\mathcal{DLR}_{\mathcal{US}}$ [Artale et al.(2002)] combines the PTL with the *Since* and *Until* and the DL $\mathcal{DLR}$ [Calvanese and De Giacomo(2003)], i.e., a expressive fragment of $L^{\{\textbf{since, until}\}}$
    - Classes, $n$-ary relations ($n > 2$), role components
    - Binary constructors ($\sqcap, \sqcup, \mathcal{U}, \mathcal{S}$) for relations of the same arity, and all boolean constructors for both class and relation expressions
    - For both classes and relations: temporal operators $\Diamond^+$, $\oplus$ , and their past counterparts can be defined via $\mathcal{U}$ and $\mathcal{S}$: $\Diamond^+ C \equiv \top \, \mathcal{U} \, C$, $\oplus \, C \equiv \bot \, \mathcal{U} \, C$, etc; $\Box^+$ and $\Box^-$ as $\Box^+ C \equiv \neg \Diamond^+ \neg C$ and $\Box^- C \equiv \neg \Diamond^- \neg C$. $\Diamond^*$ and $\Box^*$ as $\Diamond^* C \equiv C \sqcup \Diamond^+ C \sqcup \Diamond^- C$ and $\Box^* C \equiv C \sqcap \Box^+ C \sqcap \Box^- C$.
- Interpreted in *temporal models* over $\mathcal{T}$ (where $\mathcal{T} = \langle \mathcal{T}_p, < \rangle$), which are triples of the form $\mathcal{I} \doteq \langle \mathcal{T}, \Delta, \cdot^{\mathcal{I}(t)} \rangle$, where $\Delta$ is the *domain* of $\mathcal{I}$ and $\cdot^{\mathcal{I}(t)}$ an *interpretation function* s.t., for every $t \in \mathcal{T}$, every $C$, and $R$, we have $C^{\mathcal{I}(t)} \subseteq \Delta$ and $R^{\mathcal{I}(t)} \subseteq (\Delta)^n$.

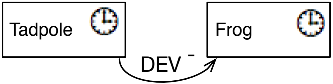
# TREND: Temporal EER

- For each TREND conceptual data model, there is an equi-satisfiable $\mathcal{DLR}_{\mathcal{US}}$ knowledge base
- Given the set-theoretic semantics for TREND, modelling notions such as satisfiability, subsumption, and derivation of new constraints have been defined (as for $ER_{VT}$ in [Artale et al.(2007a)])
- Textual and a graphical syntax along with a model-theoretic semantics as a temporal extension of the EER semantics
- TREND [Keet and Berman(2017)] supports timestamping for classes, attributes, and relationships
- Status classes [Artale et al.(2007a)] and Status relations [Artale et al.(2008)] constrain evolution of an instance's (relation's) membership in a class (relationship) along its lifespan

# Example TREND and logic and text-based notations

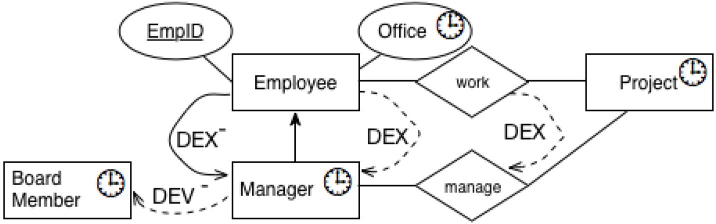
($\textsc{DevM}^{-}$) Mandatory dynamic evolution, past; e.g., Frog and the Tadpole it used to be.

- $o \in \mathit{Frog}^{\mathcal{I}(t)} \rightarrow \exists t' < t.o \in \textsc{Dev}^{\mathcal{I}(t')}_{\mathit{Tadpole},\mathit{Frog}}$
- Frog $\sqsubseteq \Diamond^{-}\textsc{Dev}_{\text{Tadpole,Frog}}$
- Diagram:




- *Each* Frog *was a(n)* Tadpole *before, but is not a(n)* Tadpole *now*.

# Example TREND diagram



# Examples

- CDM: 'RentalCar must be returned *before* Deposit is reimbursed'; e.g., $\mathit{reimbursement} \sqsubseteq \Diamond^{-} \mathit{return}$
- Domain ontology: 'Biopsy, *planned*' (in SNOMED CT); with $\Diamond^{+} \mathit{Biopsy}$

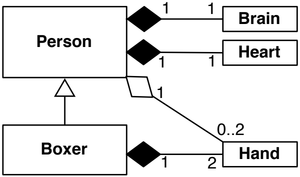
# Examples

- CDM: 'RentalCar must be returned *before* Deposit is reimbursed'; e.g., $\mathit{reimbursement} \sqsubseteq \Diamond^{-} \mathit{return}$
- Domain ontology: 'Biopsy, *planned*' (in SNOMED CT); with $\Diamond^{+} \mathit{Biopsy}$
- brain, hands, and boxer: next slides

# Outline

# Resolving the brain, hands, and boxer

- Need to represent difference between essential vs mandatory vs immutable parts and wholes, but cannot un 'plain' UML (or EER or ORM)




  - Brain is an `essential part` of Human
  - Heart is a `mandatory part` of Human but a heart can be transplanted
  - Hand is an `immutable part` of Boxer but a human can do without hands
- More generally: the life cycle semantics of parts and wholes

# Defining participation in the relation

- Two criteria: (i) nature of the dependence relationship between the classes and (ii) strength of the participation

# Defining participation in the relation

- Two criteria: (i) nature of the dependence relationship between the classes and (ii) strength of the participation
  1. Generic Dependence – Mandatory Part. The whole must have a part at each instant of its lifetime. Thus, the presence of the part is mandatory, but it can be replaced over time (e.g., the human heart example).
  2. Unconditional Specific Dependence – Essential Part. The part is mandatory, but it cannot be replaced without destroying the whole (e.g., the human brain example).
  3. Conditional Specific Dependence – Immutable Part (also called *conditionally essential part*). The part is mandatory and cannot be replaced, but only as long as the whole belongs to the class that describes it (e.g., the boxer's hand example).

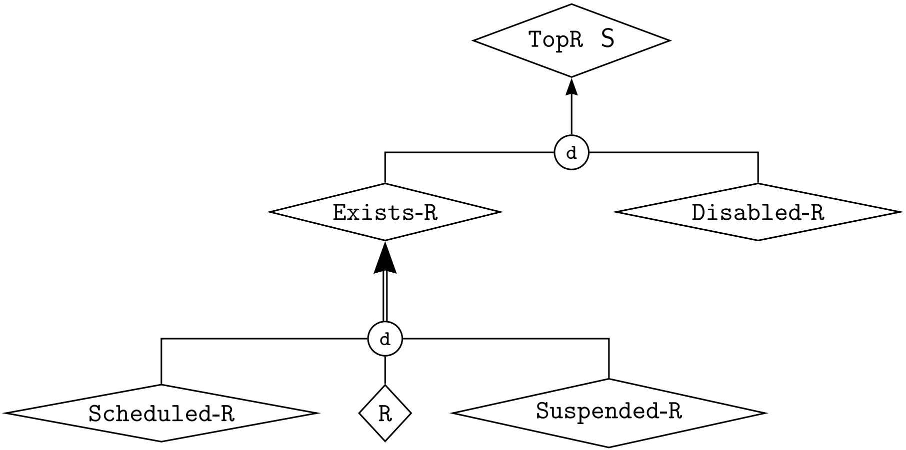
# Status relations (included in TREND)

- **Scheduled**: a relation is scheduled if its instantiation is known but its membership will only become effective some time later. e.g., a new pillar for the Sagrada Familia's interior is scheduled to become part of that church.
- **Active**: the status of a relation is active if the particular relation fully instantiates the type-level relation and only active classes can participate into an active relation; e.g., the Mont Blanc mountain is part of the Alps mountain range

# Status relations (included in TREND)

- **Suspended**: to capture a temporarily inactive relation; e.g., an instance of a CarEngine is removed from the instance of a Car it is part of for purpose of maintenance.
- **Disabled**: to model expired relations that never again can be used; e.g., to represent the donor of an organ who has donated that organ and one wants to keep track of who donated what to whom.

# Status relations



# Constraints and logical implications

**Proposition (Status Relations: Logical Implications)**

*Given the set of axioms $\Sigma_{st}$ (REXISTS-RSCH2), an n-ary relation (where $n \geq 2$) $R \sqsubseteq U_1 : C_1 \sqcap \ldots \sqcap U_n : C_n$, the following logical implications hold:*

(RACT) *Active will possible evolve into Suspended or Disabled.*
$\Sigma_{st} \models \texttt{R} \sqsubseteq \Box^{+}(\texttt{R} \sqcup \texttt{Suspended-R} \sqcup \texttt{Disabled-R})$

(RDISAB3) *Disabled will never become active anymore.*
$\Sigma_{st} \models \texttt{Disabled-R} \sqsubseteq \Box^{+}\neg\texttt{R}$

(RDISAB4) *Disabled classes can participate only in disabled relations.*
$\Sigma_{st} \models \texttt{Disabled-C}_\texttt{i} \sqcap \Diamond^{-}\exists[\texttt{U}_\texttt{i}]\texttt{R} \sqsubseteq \exists[\texttt{U}_\texttt{i}]\texttt{Disabled-R}$

# Constraints and logical implications

PROPOSITION (Status Relations: Logical Implications–*cont'd*)

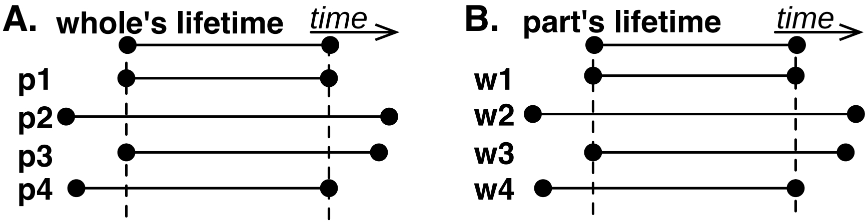
(RDISAB5) *Disabled relations involve active, suspended, or disabled classes.*
$\texttt{Disabled-R} \sqsubseteq \texttt{U}_i\texttt{:}(\texttt{C}_i \sqcup \texttt{Suspended-C}_i \sqcup \texttt{Disabled-C}_i)$, *for all* $i = 1, \ldots, n$.

(RSCH3) *Scheduled persists until active.*
$\Sigma_{st} \models \texttt{Scheduled-R} \sqsubseteq \texttt{Scheduled-R}\,\mathcal{U}\,\texttt{R}$

(RSCH4) *Scheduled cannot evolve directly to Disabled.*
$\Sigma_{st} \models \texttt{Scheduled-R} \sqsubseteq \oplus\neg\texttt{Disabled-R}$

(RSCH5) *Scheduled relations do not involve disabled classes.*
$\texttt{Scheduled-R} \sqsubseteq \texttt{U}_\texttt{i} : \neg\texttt{Disabled-C}_\texttt{i}$, *for all* $i = 1, \ldots, n$.

# Life cycles


A. whole's lifetime
time
p1
p2
p3
p4
B. part's lifetime
time
w1
w2
w3
w4

# Mandatory & Exclusive

| | | |
|---|---|---|
| (ManP) | $\mathtt{W} \sqsubseteq \exists[\mathtt{whole}]\mathtt{PartWhole}$ | *Mandatory Part* |
| (ManW) | $\mathtt{P} \sqsubseteq \exists[\mathtt{part}]\mathtt{PartWhole}$ | *Mandatory Whole* |
| (ExlP) | $\mathtt{P} \sqsubseteq \exists^{\leq 1}[\mathtt{part}]\mathtt{PartWhole}$ | *Exclusive Part* |
| (ExlW) | $\mathtt{W} \sqsubseteq \exists^{\leq 1}[\mathtt{whole}]\mathtt{PartWhole}$ | *Exclusive Whole* |

# Rigidity

### Definition (Rigid (+R))

A *rigid* property $\phi$ is a property that is essential to *all* its instances, i.e., $\forall x \phi(x) \to \Box \phi(x)$

### Definition (Anti-Rigid (∼R))

An *anti-rigid* property $\phi$ is a property that is not essential to *all* its instances, i.e., $\forall x \phi(x) \to \neg \Box \phi(x)$

$$
\begin{array}{ll}
(\textsc{Rigid}) & \mathtt{C} \sqsubseteq \Box^{*}\mathtt{C} \\
(\text{A-}\textsc{Rigid}) & \mathtt{C} \sqsubseteq \Diamond^{*}\neg\mathtt{C} \\
(\text{A-}\textsc{sub}\text{-R}) & \mathtt{C_A} \sqsubseteq \mathtt{C_R}
\end{array}
$$

# Essential parts and wholes

- *Essential parts* are global properties of rigid wholes that can be formalized in $\mathcal{DLR}_{\mathcal{US}}$ with:

| | | |
|---|---|---|
| $(\textsc{RigidW})$ | $\mathtt{W} \sqsubseteq \Box^* \mathtt{W}$ | *Rigid Whole* |
| $(\textsc{EssP})$ | $\mathtt{W} \sqsubseteq \exists[\mathtt{whole}]\Box^* \mathtt{PartWhole}$ | *Essential Part* |

- Likewise for *essential whole*

| | | |
|---|---|---|
| $(\textsc{RigidP})$ | $\mathtt{P} \sqsubseteq \Box^* \mathtt{P}$ | *Rigid Part* |
| $(\textsc{EssW})$ | $\mathtt{P} \sqsubseteq \exists[\mathtt{part}]\Box^* \mathtt{PartWhole}$ | *Essential Whole* |

# Additional axioms for Immutable

(SusW) $\texttt{Suspended-PartWhole} \sqsubseteq \texttt{whole} : \texttt{Suspended-W}$
*Suspended Whole*

(SusP) $\texttt{Suspended-PartWhole} \sqsubseteq \texttt{part} : \texttt{Suspended-P}$
*Suspended Part*

(DisP) $\texttt{Disabled-PartWhole} \sqsubseteq \texttt{part} : \texttt{Disabled-P}$
*Disabled Part*

(DisW) $\texttt{Disabled-PartWhole} \sqsubseteq \texttt{whole} : \texttt{Disabled-W}$
*Disabled Whole*

(SchPW) $\texttt{PartWhole} \sqsubseteq \Diamond^{-}\texttt{Scheduled-PartWhole}$
*Scheduled Part-Whole*

(SchP) $\texttt{Scheduled-PartWhole} \sqsubseteq \texttt{part} : \texttt{Scheduled-P}$
*Scheduled Part*

(SchW) $\texttt{Scheduled-PartWhole} \sqsubseteq \texttt{whole} : \texttt{Scheduled-W}$
*Scheduled Whole*

# Immutable part

**Theorem (Immutable Parts)**

*Let* $\mathtt{W_R}$ *be a rigid class (i.e.,* $\mathtt{W_R} \sqsubseteq \Box^* \mathtt{W_R}$*),* $\mathtt{W}$ *be an anti-rigid class (i.e.,* $\mathtt{W} \sqsubseteq \Diamond^* \neg \mathtt{W}$*) s.t.* $\mathtt{W} \sqsubseteq \mathtt{W_R}$*, and* $\mathtt{PartWhole} \sqsubseteq \mathtt{part} : \mathtt{P} \sqcap \mathtt{whole} : \mathtt{W}$ *be a generic part-whole relation satisfying* $\Sigma_{st}$*. Then, for each whole,* $o_w$*, of type* $\mathtt{W}$ *there exists an immutable part,* $o_p$*, of type* $\mathtt{P}$ *that is temporally related to* $o_w$ *with the relation:*

- $\mathtt{p2}$ *holds if* $(\textsc{ManP})$, $(\textsc{SusW})$, $(\textsc{DisW})$ *hold.*
- $\mathtt{p4}$ *holds if* $(\textsc{ManP})$, $(\textsc{SusW})$, $(\textsc{DisW})$, $(\textsc{DisP})$ *hold.*
- $\mathtt{p3}$ *holds if* $(\textsc{ManP})$, $(\textsc{SusW})$, $(\textsc{DisW})$, $(\textsc{SchPW})$, $(\textsc{SchP})$ *hold.*
- $\mathtt{p1}$ *holds if* $(\textsc{ManP})$, $(\textsc{SusW})$, $(\textsc{DisW})$, $(\textsc{DisP})$, $(\textsc{SchPW})$, $(\textsc{SchP})$ *hold.*

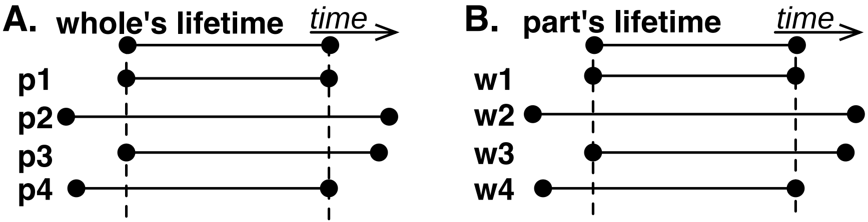
# Immutable whole

## Theorem (Immutable Wholes)

*Let* $\mathtt{P_R}$ *be a rigid class (i.e.,* $\mathtt{P_R} \sqsubseteq \Box^* \mathtt{P_R}$*),* $\mathtt{P}$ *be an anti-rigid class (i.e.,* $\mathtt{P} \sqsubseteq \Diamond^* \neg \mathtt{P}$*) s.t.* $\mathtt{P} \sqsubseteq \mathtt{P_R}$*, and* $\mathtt{PartWhole} \sqsubseteq \mathtt{part} : \mathtt{P} \sqcap \mathtt{whole} : \mathtt{W}$ *be a generic part-whole relation satisfying* $\Sigma_{st}$*. Then, for each part,* $o_p$*, of type* $\mathtt{P}$ *there exists an immutable whole,* $o_w$*, of type* $\mathtt{W}$ *that is temporally related to* $o_p$ *with the relation:*

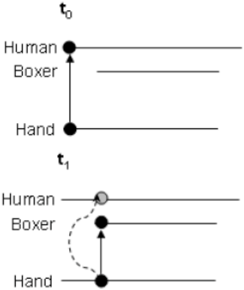
- w2 *holds if* (ManW), (SusP), (DisP) *hold.*
- w4 *holds if* (ManW), (SusP), (DisP), (DisW) *hold.*
- w3 *holds if* (ManW), (SusP), (DisP), (SchPW), (SchW) *hold.*
- w1 *holds if* (ManW), (SusP), (DisP), (DisW), (SchPW), (SchW) *hold.*

# Life cycles



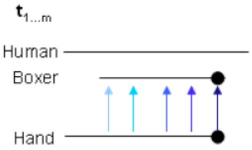

p4 holds if (ManP), (SusW), (DisW), (DisP) hold.

# The Boxer's hand (with p4)



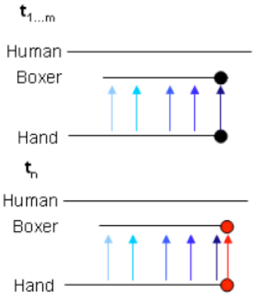

| | | |
|---|---|---|
| (1) | $\mathtt{Human} \sqsubseteq \Box^* \mathtt{Human}$ | (RIGID) |
| (2) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{PartWhole}$ | |
| (3) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Hand} \sqcap \mathtt{whole} : \mathtt{Human}$ | |
| (4) | $\mathtt{Boxer} \sqsubseteq \Diamond^* \neg \mathtt{Boxer}$ | (A-RIGID) |
| (5) | $\mathtt{Boxer} \sqsubseteq \mathtt{Human}$ | (A-SUB-R) |
| (6) | $\mathtt{Boxer} \sqsubseteq \exists^{=2}[\mathtt{whole}]\mathtt{HumanHandPW}$ | (MANP) |
| (7) | $\mathtt{Suspended\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Suspended\text{-}Boxer}$ | (SUSW) |
| (8) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Disabled\text{-}Hand}$ | (DISP) |
| (9) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Disabled\text{-}Boxer}$ | (DISW) |
| (10) | $\Sigma_{st} \models \mathtt{R} \sqsubseteq \Box^+(\mathtt{R} \sqcup \mathtt{Suspended\text{-}R} \sqcup \mathtt{Disabled\text{-}R})$ | (RACT) |
| (11) | $\Sigma_{st} \models \mathtt{Disabled\text{-}C_i} \sqcap \Diamond^- \exists[\mathtt{U_i}]\mathtt{R} \sqsubseteq \exists[\mathtt{U_i}]\mathtt{Disabled\text{-}R}$ | (RDISAB4) |

# The Boxer's hand (with p4)




| | | |
|---|---|---|
| (1) | $\mathtt{Human} \sqsubseteq \Box^{*}\mathtt{Human}$ | (RIGID) |
| (2) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{PartWhole}$ | |
| (3) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Hand} \sqcap \mathtt{whole} : \mathtt{Human}$ | |
| (4) | $\mathtt{Boxer} \sqsubseteq \Diamond^{*}\neg\mathtt{Boxer}$ | (A-RIGID) |
| (5) | $\mathtt{Boxer} \sqsubseteq \mathtt{Human}$ | (A-SUB-R) |
| (6) | $\mathtt{Boxer} \sqsubseteq \exists^{=2}[\mathtt{whole}]\mathtt{HumanHandPW}$ | (MANP) |
| (7) | $\mathtt{Suspended\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Suspended\text{-}Boxer}$ | (SUSW) |
| (8) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Disabled\text{-}Hand}$ | (DISP) |
| (9) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Disabled\text{-}Boxer}$ | (DISW) |
| (10) | $\Sigma_{st} \models \mathtt{R} \sqsubseteq \Box^{+}(\mathtt{R} \sqcup \mathtt{Suspended\text{-}R} \sqcup \mathtt{Disabled\text{-}R})$ | (RACT) |
| (11) | $\Sigma_{st} \models \mathtt{Disabled\text{-}C_i} \sqcap \Diamond^{-}\exists[\mathtt{U_i}]\mathtt{R} \sqsubseteq \exists[\mathtt{U_i}]\mathtt{Disabled\text{-}R}$ | (RDISAB4) |

# The Boxer's hand (with p4)




| | | |
|---|---|---|
| (1) | $\mathtt{Human} \sqsubseteq \Box^* \mathtt{Human}$ | (RIGID) |
| (2) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{PartWhole}$ | |
| (3) | $\mathtt{HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Hand} \sqcap \mathtt{whole} : \mathtt{Human}$ | |
| (4) | $\mathtt{Boxer} \sqsubseteq \Diamond^* \neg \mathtt{Boxer}$ | (A-RIGID) |
| (5) | $\mathtt{Boxer} \sqsubseteq \mathtt{Human}$ | (A-SUB-R) |
| (6) | $\mathtt{Boxer} \sqsubseteq \exists^{=2}[\mathtt{whole}]\mathtt{HumanHandPW}$ | (MANP) |
| (7) | $\mathtt{Suspended\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Suspended\text{-}Boxer}$ | (SUSW) |
| (8) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{part} : \mathtt{Disabled\text{-}Hand}$ | (DISP) |
| (9) | $\mathtt{Disabled\text{-}HumanHandPW} \sqsubseteq \mathtt{whole} : \mathtt{Disabled\text{-}Boxer}$ | (DISW) |
| (10) | $\Sigma_{st} \models \mathtt{R} \sqsubseteq \Box^+(\mathtt{R} \sqcup \mathtt{Suspended\text{-}R} \sqcup \mathtt{Disabled\text{-}R})$ | (RACT) |
| (11) | $\Sigma_{st} \models \mathtt{Disabled\text{-}C_i} \sqcap \Diamond^- \exists[\mathtt{U_i}]\mathtt{R} \sqsubseteq \exists[\mathtt{U_i}]\mathtt{Disabled\text{-}R}$ | (RDISAB4) |

# Summary

- Semantics of elements in conceptual modelling languages
  - Relations (standard view vs. positionalism)
  - Attributions
- Ontology-informed language design process
  - Principal design choices
  - Profiles
- Temporal conceptual models

# References I


A. Artale, E. Franconi, F. Wolter, and M. Zakharyaschev.
A temporal description logic for reasoning about conceptual schemas and queries.
In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer Verlag, 2002.

A. Artale, C. Parent, and S. Spaccapietra.
Evolving objects in temporal information systems.
*Annals of Mathematics and Artificial Intelligence*, 50(1-2):5–38, 2007a.

A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev.
The DL-Lite family and relations.
*Journal of Artificial Intelligence Research*, 36:1–69, 2009.

Alessandro Artale, Diego Calvanese, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev.
Reasoning over extended ER models.
In Christine Parent, Klaus-Dieter Schewe, Veda C. Storey, and Bernhard Thalheim, editors, *Proceedings of the 26th International Conference on Conceptual Modeling (ER'07)*, volume 4801 of *LNCS*, pages 277–292. Springer, 2007b.
Auckland, New Zealand, November 5-9, 2007.

Alessandro Artale, Nicola Guarino, and C. Maria Keet.
Formalising temporal constraints on part-whole relations.
In Gerhard Brewka and Jerome Lang, editors, *11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 673–683. AAAI Press, 2008.
Sydney, Australia, September 16-19, 2008.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors.
*The Description Logics Handbook – Theory and Applications*.
Cambridge University Press, 2 edition, 2008.

# References II


D. Berardi, D. Calvanese, and G. De Giacomo.
Reasoning on UML class diagrams.
*Artificial Intelligence*, 168(1-2):70–118, 2005.

A. C. Bloesch and T. A. Halpin.
Conceptual Queries using ConQuer-II.
In *Proceedings of ER'97: 16th International Conference on Conceptual Modeling*, volume 1331 of *LNCS*, pages 113–126. Springer, 1997.

Bernardo F. B. Braga, João Paulo A. Almeida, Giancarlo Guizzardi, and Alessander Botti Benevides.
Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal methods.
*Innovations in Systems and Software Engineering*, 6(1-2):55–63, 2010.

Jordi Cabot, Robert Clarisó, and Daniel Riera.
Verification of UML/OCL class diagrams using constraint programming.
In *Model Driven Engineering, Verification, and Validation: Integrating Verification and Validation in MDE (MoDeVVA 2008)*, 2008.

D. Calvanese and G. De Giacomo.
*The DL Handbook: Theory, Implementation and Applications*, chapter Expressive description logics, pages 178–218.
Cambridge University Press, 2003.

Diego Calvanese, C. Maria Keet, Werner Nutt, Mariano Rodríguez-Muro, and Giorgio Stefanoni.
Web-based graphical querying of databases through an ontology: the WONDER system.
In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung, editors, *Proceedings of ACM Symposium on Applied Computing (ACM SAC'10)*, pages 1389–1396. ACM, 2010.
March 22-26 2010, Sierre, Switzerland.

# References III


Diego Calvanese, Domenico Carbotta, and Magdalena Ortiz.
A practical automata-based technique for reasoning in expressive description logics.
In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11)*, 2011.

Charles B. Cross.
Armstrong and the problem of converse relations.
*Erkenntnis*, 56:215–227, 2002.

F. Donini, M. Lenzerini, D. Nardi, and W. Nutt.
Tractable concept languages.
In *Proc.of IJCAI'91*, volume 91, pages 458–463, 1991.

P. R. Fillottrani and C. M. Keet.
Conceptual model interoperability: a metamodel-driven approach.
In A. Bikakis et al., editors, *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, volume 8620 of *LNCS*, pages 52–66. Springer, 2014.
August 18-20, 2014, Prague, Czech Republic.

Pablo R. Fillottrani and C. Maria Keet.
Patterns for heterogeneous tbox mappings to bridge different modelling decisions.
In E. Blomqvist et al., editors, *Proceeding of the 14th Extended Semantic Web Conference (ESWC'17)*, volume 10249 of *LNCS*, pages 371–386. Springer, 2017.
30 May - 1 June 2017, Portoroz, Slovenia.

# References IV


Pablo Rubén Fillottrani and C. Maria Keet.
Evidence-based languages for conceptual data modelling profiles.
In T. Morzy et al., editors, *19th Conference on Advances in Databases and Information Systems (ADBIS'15)*, volume 9282 of *LNCS*, pages 215–229. Springer, 2015.
8-11 Sept, 2015, Poitiers, France.

Kit Fine.
Neutral relations.
*The Philosophical Review*, 109(1):1–33, 2000.

Ulrich Frank.
Domain-specific modeling languages - requirements analysis and design guidelines.
In I. Reinhartz-Berger, A. Sturm, T. Clark, J. Bettin, and S. Cohen, editors, *Domain Engineering: Product Lines, Conceptual Models, and Languages*, pages 133–157. Springer, 2013.

Amir Jahangard Rafsanjani and Seyed-Hassan Mirian-Hosseinabadi.
A Z Approach to Formalization and Validation of ORM Models.
In Ezendu Ariwa and Eyas El-Qawasmeh, editors, *Digital Enterprise and Information Systems*, volume 194 of *CCIS*, pages 513–526. Springer, 2011.

C. M. Keet and L. Khumalo.
On the ontology of part-whole relations in Zulu language and culture.
In S. Borgo and P. Hitzler, editors, *10th International Conference on Formal Ontology in Information Systems 2018 (FOIS'18)*, page (in print). IOS Press, 2018.
17-21 September, 2018, Cape Town, South Africa.

# References V


C. Maria Keet.
Factors affecting ontology development in ecology.
In B Ludäscher and L. Raschid, editors, *Data Integration in the Life Sciences 2005 (DILS2005)*, volume 3615 of *LNBI*, pages 46–62. Springer Verlag, 2005.
San Diego, USA, 20-22 July 2005.

C. Maria Keet.
Positionalism of relations and its consequences for fact-oriented modelling.
In R. Meersman, P. Herrero, and Dillon T., editors, *OTM Workshops, International Workshop on Fact-Oriented Modeling (ORM'09)*, volume 5872 of *LNCS*, pages 735–744. Springer, 2009.
Vilamoura, Portugal, November 4-6, 2009.

C. Maria Keet.
Transforming semi-structured life science diagrams into meaningful domain ontologies with DiDOn.
*Journal of Biomedical Informatics*, 45:482–494, 2012.
doi: doi:10.1016/j.jbi.2012.01.004.

C. Maria Keet and Alessandro Artale.
Representing and reasoning over a taxonomy of part-whole relations.
*Applied Ontology*, 3(1-2):91–110, 2008.

C. Maria Keet and Sonia Berman.
Determining the preferred representation of temporal constraints in conceptual models.
In H.C. Mayr et al., editors, *36th International Conference on Conceptual Modeling (ER'17)*, volume 10650 of *LNCS*, pages 437–450. Springer, 2017.
6-9 Nov 2017, Valencia, Spain.

# References VI


C. Maria Keet and Pablo Rubén Fillottrani.
Toward an ontology-driven unifying metamodel for UML class diagrams, EER, and ORM2.
In W. Ng, V. C. Storey, and J. Trujillo, editors, *32nd International Conference on Conceptual Modeling (ER'13)*, volume 8217 of *LNCS*, pages 313–326. Springer, 2013.
11-13 November, 2013, Hong Kong.

C. Maria Keet and Pablo Rubén Fillottrani.
An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2.
*Data & Knowledge Engineering*, 98:30–53, 2015.
doi: 0.1016/j.datak.2015.07.004.

Zubeida Khan and C. Maria Keet.
ONSET: Automated foundational ontology selection and explanation.
In A. ten Teije et al., editors, *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW'12)*, volume 7603 of *LNAI*, pages 237–251. Springer, 2012.
Oct 8-12, Galway, Ireland.

Zubeida C. Khan, C. Maria Keet, Pablo R. Fillottrani, and Karina Cenci.
Experimentally motivated transformations for intermodel links between conceptual models.
In J. Pokorný et al., editors, *20th Conference on Advances in Databases and Information Systems (ADBIS'16)*, volume 9809 of *LNCS*, pages 104–118. Springer, 2016.
August 28-31, Prague, Czech Republic.

Joop Leo.
Modeling relations.
*Journal of Philosophical Logic*, 37:353–385, 2008.

# References VII


Matthew Nizol, Laura K. Dillon, and R. E. K. Stirewalt.
Toward tractable instantiation of conceptual data models using non-semantics-preserving model transformations.
In *Proceedings of the 6th International Workshop on Modeling in Software Engineering (MiSE'14)*, pages 13–18. ACM Conference Proceedings, 2014.
Hyderabad, India, June 02-03, 2014.

Wen-Lin Pan and Da-xin Liu.
Mapping object role modeling into common logic interchange format.
In *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE'10)*, volume 2, pages 104–109. IEEE Computer Society, 2010.

Anna Queralt, Alessandro Artale, Diego Calvanese, and Ernest Teniente.
OCL-Lite: Finite reasoning on UML/OCL conceptual schemas.
*Data & Knowledge Engineering*, 73:1–22, 2012.

Y. Smaragdakis, C. Csallner, and R. Subramanian.
Scalable satisfiability checking and test data generation from modeling diagrams.
*Automation in Software Engineering*, 16:73–99, 2009.

Ahmet Soylu, Evgeny Kharlamov, Dimitry Zheleznyakov, Ernesto Jimenez Ruiz, Martin Giese, Martin G. Skjaeveland, Dag Hovland, Rudolf Schlatte, Sebastian Brandt, Hallstein Lie, and Ian Horrocks.
OptiqueVQS: a visual query system over ontologies for industry.
*Semantic Web Journal*, page in press, 2017.

David Toman and Grant E. Weddell.
*Fundamentals of Physical Design and Query Compilation*.
Synthesis Lectures on Data Management. Morgan & Claypool, 2011.

# References VIII


Peter van Inwagen.
Names for relations.
*Philosophical perspectives*, 20(1):453–477, 2006.

Heba M. Wagih, Doaa S. El Zanfaly, and Mohamed M. Kouta.
Mapping Object Role Modeling 2 schemes into $\mathcal{SROIQ}(d)$ description logic.
*International Journal of Computer Theory and Engineering*, 5(2):232–237, 2013.

Timothy Williamson.
Converse relations.
*The Philosophical Review*, 94(2):249–262, 1985.

# Thank you!

For more information, papers, data sets, presentations and other files, please visit
`http://www.meteck.org/`