Lecture Notes on Ontology Modeling Challenges and OntoClean Birzeit University, Palestine 2012

Advanced Topics in Ontology Engineering

Ontology Modeling Challenges and (OntoClean)

Dr. Mustafa Jarrar

Sina Institute, University of Birzeit

mjarrar@birzeit.edu www.jarrar.info





Guarino, Nicola and Chris Welty. 2002. **Evaluating Ontological Decisions with OntoClean**. Communications of the ACM. 45(2):61-65. New York: ACM Press. http://www.loa.istc.cnr.it/Papers/CACM2002.pdf

Most information based on source: Guarino

Jarrar © 2012

Ontology Modeling

- Building ontologies is still arcane art form.
- One maybe a good ontology engineer, but he does not know why!
- An ontology might be better than another, but it is difficult to know why!
- We need a methodology to guide us not only on what kinds of ontological decisions we should make, but on how these decisions can be evaluated.
 - OntoClean provides a set of modeling "guidelines" in this direction, but it is still not meant to be a comprehensive methodology for the all ontology modeling phases.



A methodology for ontology-driven conceptual analysis, developed by:



Nicola Guarino

CNR Institute for Cognitive Sciences and Technologies, Laboratory for Applied Ontology (LOA) in Trento, Italy



Chris Welty

Research Scientist at the IBM T.J. Watson Research Center in New York.

Modeling mistakes

- Many people misuse the subsumption relation, what they do is not subsumptions.
- Which are correct/wrong here?



Modeling mistakes

- Many people misuse the subsumption relation, what they do is not subsumptions.
- Which are correct/wrong here?

How to know that your modeling choices are right?

ite

Stuc • What makes your ontology better than my ontology?



Subsumption (The Subtype Relation)

- The subsumption (also called is-a relationship) forms a hierarchy
 - A subsumes B if all instances of B are necessarily instances of A
 - Attributes of a supertype are inherited by it subtypes along the hierarchy.

- The subsumption relation is often misused
 - People are typically confuses the subtype relation with the part-of and instance-of relations, and types with roles.

Subsumption misused with Instantiation



- Instances are sometimes confused with types!
- To distinguish between them you should ask what are the instances of "Mustafa"? Instances of "Human"? Instance of Species? Etc.
- How to distinguish between same/different instances of a type, two things are same entity (identity criteria)?

SubtypeOf

Engine

It is often difficult for beginners in ontological analysis to distinguish between the part-of and the subclass relation. This is due to the fact that subclass is analogous to subset, and a subset of a set is a part of it. This confusion can be overcomed when we realize the difference between the parts of a set and the parts of its members.

Engine

→ Among the essential properties of a car there are some functional properties, like being able to accommodate people. An engine has also certain functional properties as essential properties, like being able to crank and generate a rotational force. Since, however, the essential properties of cars do not apply to engines, one cannot subsume the other. The proper relationship here is part and subsumption is not part. [GW02].

Subsumption misused with Disjunction



To see how this is incorrect, rigidity analysis can be most useful. No instance of a car part is necessarily a car part (we could take an engine from a car and put it in a boat, making it no longer a car part but a boat part), so we have to make that class anti-rigid. The class engine itself is rigid, however, since we can't imagine an entity that is an engine becoming a non-engine. Being an engine is essential to it. An anti-rigid class, such as car part, can not subsume a rigid one, and so we have a conflict.

This type of mistake is particularly common in object-oriented, where value restrictions are an important part of modeling. These anti-rigid classes are created to satisfy a modeling need to represent disjunction, for instance, "any car part is either an engine, or a wheel, or a seat, …" It should be clear that this is different from saying "all engines are car parts," since, in fact, they are not. Of course, most modeling systems do not provide for disjunction, so modelers believe they are justified using these tricks, but if the intention is to make meaning as clear as possible then *subsumption is not disjunction*. [GW02]

Jarrar © 2012

Subsumption misused with Constitution



• An instance of Water is an amount of water

- An instance of Ocean is the "the Atlantic Ocean,"
- Oceans are made up of amounts of water



Subsumption misused with Polysemy

- A term may have multiple meanings (Polysemy),
- For example "Table" means:
 - A piece of furniture having a smooth flat top that is usually supported by one or more vertical legs.
 - A set of data arranged in rows and columns



A polysemous class might be placed below both meanings



OntoClean helps you to:

- evaluate misuses of subsumption and inconsistent modeling choices.
- provides a formal basis for why they're wrong.
- Focuses on *taxonomy*

A subsumes B iff for all x, x instance of B implies x instance of A

OntoClean

based on [Bechhofer]

- Considers general properties
 - being an apple
 - being a table
 - being a person
 - being red
- A Class is then the set of entities that exhibit a property in a possible world.
- Members of the Class are instances of the property.
- The terminology can be slightly confusing

- These are not properties in the sense of OWL properties.



based on [Bechhofer]

- *Metaproperties* describe particular characteristics of the properties.
 - Essence (صفة الجو هر)
 - Rigidity (صفة اللزوم)
 - Identity (صفة الهوية)
 - Unity (صفة الوحدة)
- Associating metaproperties with properties (i.e. classes) helps characterize aspects of the properties.
- *Constraints* on the metaproperties enforce restrictions on the taxonomy and help to highlight and evaluate the choices made.

(الجوهر) Essence

• A property (i.e. class) is essential to an entity if it must hold for it.

- Not just things that accidently happen to be true all the time.

تكون الصفة جوهرية لكينونة ما إذا كان إجبارية

تكون الصفة غير جو هرية الكينونة ما إذا كانت عرضية، لوقت معين وليس طوال الوقت.

(صفة اللزوم) Rigidity

- A <u>rigid</u> property is a property that is essential to all of its instances.
 تكون الصفة <u>لازمة</u> إذا كانت جو هرية لجميع حامليها، مثل صفة إنسان لأن من يحمل هذه الصفة لا يمكنه التخلي عنها في أي وقت أو ظرف.
 - Every entity that can exhibit the property must do so.
 - Every entity that is a person must be a person
 - There are no entities that can be a person but aren't.
- An <u>anti-rigid</u> property is one that is never essential
 تكون الصفة غير لازمة إذا لم تكن جو هرية لجميع حامليها، مثل صفة طالب لأن من يحمل هذه الصفة يمكنه التخلي عنها.
 - For example, every instance of student isn't necessarily a student students may cease to be students at some point without ceasing to exist or changing their identity.
- A <u>semi-rigid</u> property is one that is essential to some instances but not to others

تكون الصفة شبه لازمة إذا كانت جو هرية لبعض حامليها، مثل صفة قاسي فهي جو هرية للبعض مثل "مطرقة" وغير جو هرية للبعض الآخر مثل "الإسفنج" القاسي.

(صفة اللزوم) Rigidity

• A <u>rigid</u> property is a property that is essential to all of its instances. تكون الصفة لازمة إذا كانت جو هرية لجميع حمليها، مثل صفة إنسان لأن من يحمل هذه الصفة لا

بمكنه التخلي عنها في أي وقت أو ظرف

- Every
- Every
- There
- An <u>anti-r</u>i حمل هذه الصفة
 - For example
 studer
 or cha
- A <u>semi-ric</u> not to othe ية للبعض مثل

- Each concept in the Ontology should be labeled with Rigid (**+R**), Anti-rigid (**-R**), or Semi-rigid (**~R**).
 - As we will see later, these metaproperties impose constraints on the subsumption relation, which can be <u>used to check the ontological consistency</u> of taxonomic links.
 - One of these constraints is that <u>anti-rigid</u> properties cannot subsume rigid properties. Thus Student cannot subsume Person.

ist

(الهوية) Identity

Identity criteria allows us to recognize individual entities in the world.
 التفكير بالشروط التي تجعل الشيء ذاته تساعدنا على تمييز الأشياء.

Necessary and sufficient criteria in terms of definitions
 What is the difference between "Time Duration" and "Time Interval"?

- Time Duration
 - One hour, two hours etc.
- Time Interval
 - 11:00-12:00 on Tuesday 26th, 17:00-18:45 on Saturday 17th



When we say "all time intervals are time durations" we really mean "all time intervals have a time duration"; Jarrar © 2012 19

الهوية) Identity

- Identity refers to the problem of being able to recognize individual entities in the world as being the same (or different).
- Identity criteria are conditions used to determine equality (sufficient conditions) and that are entailed by equality (necessary conditions).
- For example, how do we recognize a person we know as the same person even though they may have changed?
- Thinking about concepts' identities, while building an ontology help us avoid/discover mistakes.
- For example: think again about this

 Time Duration
 Instances like: "One hour", "two hours", "one day" etc.

 SubtypeOf?
 Instances like: "1:00–2:00 next Tuesday", "2:00–3:00 next Wednesday", etc.

 Time Interval
 Instances like: "1:00–2:00 next Tuesday", "2:00–3:00 next Wednesday", etc.

→ since all Time Intervals are also Time Durations.

Jarrar © 2012

(الهوية) Identity

According to the identity criteria for time durations, two durations of the same length are the same duration. In other words, all one hour time durations are identical—they are the same duration and therefore there is only one "one hour" time duration.

According to the identity criteria for time intervals, two intervals occurring at the same time are the same, but two intervals occurring at different times, even if they are the same length, are different. Therefore, the two example intervals given would be different intervals, but the same duration.

<u>This creates a contradiction</u>: if all instances of time interval are also instances of time duration (as implied by the subclass relationship), how can they be two instances under one class and a single instance under another?



→ since all Time Intervals are also Time Durations.

Jarrar © 2012

(الهوية) Identity

When we say "<u>all time intervals are time durations</u>" we really mean "<u>all time</u> <u>intervals have a time duration</u>"; the duration is a component of an interval, but it is not the interval itself. Therefore, we cannot model the relationship as subclass.



22

(الذاتية) Identity

When we say "<u>all time intervals are time durations</u>" we really mean "<u>all time</u> <u>intervals have a time duration</u>"; the duration is a component of an interval, but it is not the interval itself. Therefore, we cannot model the relationship as subclass.

- A property will inherit identity criteria from parents.
- A property may also supply its own additional identity criteria.



(الهوية) Unity

- <u>Unity</u> refers to the problem of describing the way the parts of an object are bound together.
- <u>Unity criteria</u> identify whether or not a property is intended to describe whole objects.
- For some classes, all their instances are wholes (like Car), for others none of their instances are wholes (like Oil).
- A property **carries unity** if all its instances exhibit a common unity criterion (e.g., Ocean)
- A property **carries no unity** if its instances are all wholes, but with different unity criteria (Legal Entity, as it includes companies and people
- A property **carries anti-unity** if all of its instances are not necessarily whole.



 Thinking about concepts' identities, while building an ontology helps us avoid/discover mistakes. For example:



Instance is an **amount of water**, but it is not a whole, since it is not recognizable as an isolated entity.

Instances like the "Atlantic Ocean", is recognizable as a single entity

If we claim that instances of the latter are not wholes, and instances of the former always are, then we have a <u>contradiction</u>. Problems like this again stem from the ambiguity of natural language, oceans are not "kinds of "water", they are composed of water.



If we claim that instances of the latter are not wholes, and instances of the former always are, then we have a <u>contradiction</u>. Problems like this again stem from the ambiguity of natural language, oceans are not "kinds of "water", they are composed of water.

OntoClean's support!

- Recognizing identity and unity criteria is typically very difficult, and needs philosophical/analytical skill.
- Well, of course good ontologies are not easy and need deep thinking!
- These difficulties are not simplified much by OntoClean, but there is no better methodology!
- More examples and practice help you become a good ontology modeler! → then your salary will be very high! ;-)

How to apply OntoClean?

based on [Bechhofer]

Given two classes A and B, where B subsumes A, the following rules must hold:

SubtypeOf A

Rules

- 1. If B is anti-rigid, then A must be anti-rigid
- 2. If B carries an identity criterion, then A must carry the same criterion
- 3. If B carries a unity criterion, then A must carry the same criterion
- 4. If B has anti-unity, then A must also have anti-unity
- 5. If B is externally dependent, then A must be
- Analysis of a hierarchy using these rules can help identify problematic modeling choices.
- Conceptual backbone of rigid properties

Jarrar © 2012



based on [Bechhofer]

- OntoClean helps a modeler to justify and analyze the choices made in defining a subsumption hierarchy.
- Metaproperties :
 - Rigidity
 - Identity
 - Unity
 - Dependent
- Application of constraints on metaproperties
- Highlights potential inconsistencies in the modelling



Guarino, Nicola and Chris Welty. 2002. **Evaluating Ontological Decisions with OntoClean**. Communications of the ACM. 45(2):61-65. New York: ACM Press.

Sean Bechhofer: OntoClean. COMP30412. University of Manchester http://www.cs.man.ac.uk/~seanb/teaching/COMP30412/OntoClean.pdf