Pizza & Wine: The Need for Educational Tools for Foundational Ontologies

Stefan SCHULZ[a,b,1], Martin BOEKER[b], José A. VERA RAMOS[a,c] and Ludger JANSEN[d]
[a] *Institute for Medical Informatics, Statistics and Documentation, Medical University of Graz, Austria*
[b] *Institute for Medical Biometry and Statistics, Faculty of Medicine and Medical Center - University of Freiburg, Germany*
[c] *Faculty of Biology, University of Murcia, Spain*
[d] *Institute of Philosophy, University of Rostock, Germany*

Abstract. The educational ontologies PIZZA and WINE&FOOD have been widely used to teach ontology methods and tools. However, the two ontologies have remained largely unchanged for many years, despite increasing awareness of foundational principles for good ontology design (GoodOD). Recognizing that the two ontologies are lagging behind such principles, we analysed and re-designed them by strictly adhering to established upper-level ontology constraints provided by the foundation ontology BioTopLite2 (BTL2). As a result, the redesign required clarifying the ontological commitment of the PIZZA and WINE&FOOD classes by assigning them to top-level classes. We expect the ongoing redesign effort to yield a harmonized PIZZA&WINE&FOOD ontology. The redesign reflects the change in requirements for educational ontologies, which have now to take into account the privileged role that should be granted to foundational ontologies in Applied Ontology education.

Keywords. Foundational Ontologies, Educational Ontologies

1. Introduction

Teaching ontology to a broad variety of users requires a representational domain that is both intuitively understandable and complex enough to demonstrate all important features of ontology languages and editors. This is the reason why the ontologies WINE&FOOD and PIZZA have found broad acceptance in educational settings which focus on Description Logics (DLs) and the Semantic Web. WINE&FOOD was originally developed for teaching CLASSIC [1], an early dialect of description logics (DL), and was later transformed into Protégé Frames, for which ontology-oriented tutorials existed [2]. In the last stage of its current development, it was transformed into OWL-DL, and examples taken from WINE&FOOD were used to explain syntax and semantics in the first OWL-DL guide [3]. PIZZA was developed for training courses at the University of Manchester with the goal to teach the Web Ontology Language (OWL) and the ontology editor Protégé [4], where examples from PIZZA are used in its printed documentation. The two ontologies are, hence, primarily tailored to demonstrate the power of DL representation and reasoning (set theory, restrictions and

[1] Corresponding author: Stefan Schulz, Medical University of Graz, Auenbruggerplatz 2/V, 8036 Graz (Austria), E-mail: stefan.schulz@medunigraz.at.

quantifications), and to acquaint the students with Protégé. However, both ontologies have not been updated for years, and they have not kept up with more recent tendencies in ontology development, such as the good practice in the new technical discipline of Applied Ontology [5] to employ principles of the philosophical discipline of Ontology when engineering ontology artefacts. Both educational ontologies that are under scrutiny here are agnostic regarding ontological upper-levels and naming conventions, as well as regarding the *de facto* adoption of the language profiles OWL-EL and OWL-DL as the ones supported by the most popular DL reasoners.

The objective of this paper is to evaluate the two educational ontologies on the background of these new developments. We first analyse how PIZZA and WINE&FOOD, in their current state, are positioned regarding quality criteria for good ontology design. Second, we test whether they can be redesigned to meet these criteria by aligning them with a pre-existing foundational ontology. Finally, we investigate whether the redesigned ontologies support two learning objectives, *viz.* (i) mastering OWL and description logics and (ii) understanding foundational ontologies.

This paper primarily describes the alignment of these two ontologies with a foundational ontology. But besides giving an insight into redesign aspects (which could be an educational goal *per se*), we will also discuss the goals of ontology education, possible methods, the role of foundational ontologies in this process and the usefulness of PIZZA and WINE&FOOD when seen in this light.

2. Materials

We use the PIZZA ontology version 2.0.0[2] and the WINE&FOOD ontology, which is provided as an annex to the 2004 OWL Web Ontology guide.[3] "P" is used as namespace prefix for PIZZA, "W" for WINE&FOOD. PIZZA contains 100 classes and five individuals plus five object properties with three corresponding inverses (Table 1). Most of its content is under *Food*, which branches into the three disjoint classes *Pizza*, *PizzaBase*, and *PizzaTopping*. *Pizza* has subclasses like: *CheeseyPizza*, *MeatyPizza*, *InterestingPizza*, *VegetarianPizza*, etc. These subclasses of *Pizza* classes are subsumed under the helper class *NamedPizza*. The class *PizzaTopping* has subclasses like *CheeseTopping*, *FishTopping*, *MeatTopping*, etc. All toppings are primitive classes apart from *SpicyTopping* and *VegetarianTopping*.

Axioms describe the composition of pizzas according to their base, toppings and country of origin of the recipe. Only the more general classes of the *Pizza* ontology are fully defined, e.g.:

P:VegetarianPizza equivalentTo *P:Pizza* and not (**P:hasTopping** some *P:MeatTopping*)
and not (**P:hasTopping** some *P:FishTopping*)

PIZZA uses role hierarchies, nominals, inverse roles, unqualified number restrictions and data types [6]. The ontology includes six individuals, five of which are used in the extensional definition of the class *P:Country*. Most classes have English labels as annotation properties, according to the W3C standard [3] and many also Brazilian Portuguese ones. The annotation property 'comment' is mostly left blank, in

[2] https://protege.stanford.edu/ontologies/pizza/pizza.owl
[3] https://www.w3.org/TR/owl-guide/wine.rdf

some cases it repeats the formal definition of a class; in other cases it has educational content.

Table 1. Object properties of PIZZA with algebraic properties, domain and range. Numbers indicate usage, as displayed by Protégé

Object property / Inverse	Algebraic properties	Domain	Range
P:hasCountryOfOrigin (14)	–	*owl:Thing*	*owl:Thing*
P:hasIngredient (17) / **P:isIngredientOf**(17)	transitive	*P:Food*	*P:Food*
P:hasTopping (301) / **P:isToppingOf** (15)	(inv) functional	*P:Pizza*	*P:PizzaTopping*
P:hasBase (22) / **P:isBaseOf** (16)	(inv) functional	*P:Pizza*	*P:PizzaBase*
P:hasSpiciness (77)	functional	*owl:Thing*	*P:Spiciness*

Table 2. Object properties of WINE&FOOD with algebraic properties, domain and range. Numbers indicate usage, as displayed by Protégé. Dashes are used to represent absence of algebraic properties

Object property / Inverse	Algebraic properties	Domain	Range
W:adjacentRegion(10)	symmetric	*W:Region*	*W:Region*
W:course (10)	–	*W:Meal*	*W:MealCourse*
W:hasDrink (132)	–	*W:MealCourse*	*W:PotableLiquid*
W:hasFood (56)	–	*W:MealCourse*	*W:EdibleThing*
W:hasMaker (116) / **W:producesWine** (6)	functional	*owl:Thing*	*owl:Thing*
W:hasVintageYear(12)	functional	*W:Vintage*	*W:VintageYear*
W:hasWineDescriptor(14)	–	*W:Wine*	*W:WineDescriptor*
W:hasBody (162)	functional	*owl:Thing*	*W:WineBody*
W:hasColor (94)	functional	*W:Wine*	*W:WineColor*
W:hasFlavor (170)	functional	*owl:Thing*	*W:WineFlavor*
W:hasSugar (168)	functional	*owl:Thing*	*W:WineSugar*
W:locatedIn (188)	transitive	*owl:Thing*	*W:Region*
W:madeFromFruit (10)	–	*W:ConsumableThing*	*W:Fruit*
W:madeFromGrape(92) / **W:madeIntoWine** (6)	–	*W:Wine*	*W:WineGrape*

For WINE&FOOD we use the namespace prefix "W". The ontology contains 138 classes, 16 object properties, one data type property and 206 individuals (e.g. with wine colours such as **W:Red**, **W:Rose** and **W:White** as individuals). The main classes are: *W:Wine*, *W:Region*, *W:Vintage*, *W:VintageYear*, *W:Winery,* and *W:WineDescriptor* with the subclasses *W:WineColor* and *W:WineTaste*. Table 2 shows the object properties. There is one datatype property, **W:yearValue**.

3. Methods

We evaluate the ontologies in the light of the quality criteria stated in the GoodOD guideline for good ontology design of OWL ontologies with Protégé [7]. On the assumption that usability, maintainability and sustainable evolution of ontologies are important desiderata, the GoodOD guideline and its recommendations aim at optimising user-friendliness and interoperability of DL-based ontologies. GoodOD takes into account established best practice rules in ontology design [8] and class labelling [9], and reflects good practice of many previously published OWL ontologies. For the present study, we use the following quality requirements in particular:

- DL semantics should not be compromised: without exception, class-level axioms must hold true for all members of a class.
- There should be a robust and interoperable top-level ontology whose classes cover the domain exhaustively. For better usability, the top level is designed in a way that all domain classes can be uniquely attributed to one top-level class.
- Naming and annotation conventions should be consistently followed.
- Classes should always correspond to 'repeatable' features of the domain. E.g., the class *P:Pizza* can be instantiated by many particular pizzas; and *P:Spicy* is repeated in every particular spiciness.
- The division between individuals and classes should reflect the ontological or set-theoretic interpretation of ontology classes and not be motivated by mere subjective criteria for particular use cases.
- The object properties provided by the upper level ontology should be largely sufficient for the domain ontology. For new predicates, preference should be given to express them by means of process classes rather than as new *ad hoc* object properties (reification).
- The interpretation of the meaning of classes and object properties should be clear and understandable.

We checked the two ontologies against these quality criteria (Sect. 4.1), from which we then derived principles for redesign. The two ontologies have then been redesigned accordingly while preserving all entities (Sect. 4.2). The resulting artefacts are then assessed against the questions stated in section 1 (Sect. 5).

4. Results

4.1. Quality analysis

A positive point of PIZZA is the availability of an up-to-date, extensive and didactically optimized tutorial. We found, however, the following shortcomings:

- No reference to any foundational ontology.
- The top-level bipartition into the classes labelled "Domain concepts" and "Value partitions" seems to be purely navigational, its ontological significance remains unclear.
- Use of non-rigid classes [10], e.g. *P:Food*.
- Extensional class definitions: *P:Country* defined as having exactly five members, which is objectively wrong.
- Idiosyncratic object properties, with their meaning restricted to the Pizza domain, partly unclear such as **P:hasCountryOfOrigin**.
- Naming lacks precision, e.g. *P:TabascoPepperSauce* is surprisingly a descendent of *P:PizzaTopping*, because it means topping with tabasco pepper.

In the WINE&FOOD ontology, we found the following shortcomings:

- No reference to any foundational ontology.
- Very domain-specific top-level partition into consumable and non-consumables, regions, vintages, wineries and wine descriptors.
- Idiosyncratic object properties: e.g. highly domain-specific as in PIZZA.
- Numerous potentially ambiguous labels, e.g. *W:Loire* intended to mean wine from the Loire region and not the river Loire.
- Unprincipled instance/class division: *W:Chianti* is a class, whereas **`W:ChiantiClassico`** is an individual.
- Extensional definitions, e.g. *W:WineSugar* defined as {**`W:Dry`**, **`W:OffDry`**, **`W:Sweet`**}.
- The ontology is completely devoid of metadata.
- In the tutorial [2], numerous references still point to the old Protégé frames version, using frame terminology, like "concept", "slot" etc.

4.2. Redesign

Applying the recommendations of the GoodOD guideline, we refined the two educational ontologies into the re-modelled counterparts PIZZA$^+$ and WINE&FOOD$^+$, with the namespace identifiers "P$^+$" and "W$^+$". The new ontologies import the upper-level ontology BioTopLite2 (BTL2, namespace "btl2") [11]. Accordingly, in PIZZA$^+$, *P$^+$:Food* is a subclass of *btl2:PolyMolecularCompositeEntity*. Since an instance of this class is required to have clear unity and identity criteria [10], we changed its name to "FoodItem". The class is subclass of the top-level class *btl2:MaterialObject*, defined as having one mass and one volume at a time. Also, the object properties of BTL2 are intended to be exhaustive. In accordance with the GoodOD criteria we strived to fully

express the meaning of the PIZZA$^+$ and WINE&FOOD$^+$ axioms by using exclusively BTL2 object properties.

4.2.1. PIZZA$^+$

The re-implementation of PIZZA under BTL2 required numerous changes. First, we replaced all ontology-specific object properties by BTL2 object properties, e.g. **P:hasTopping** by **btl2:hasComponentPart,** a non-transitive subproperty of **btl2:hasPart**, which relates components with a compound. Components strictly partition the compound, and the compound is the mereological sum of its components. A loss of some component affects the integrity of the compound and can change the type it instantiates; e.g., it can change from a complete to a defective organism. The relation **btl2:isBearerOf** (inverse: **btl2:inheresIn**) relates realizable entities (qualities, roles, functions, dispositions, or information objects) with the physical entity they depend on. In BTL2, general localization is expressed by the object property **btl2:includes** (inverse **btl2:isIncludedIn**). It is transitive and relates a place with an entity which occurs, inheres, or is part of it. E.g., the original PIZZA included the following closure axioms:

P:AmericanPizza subClassOf *P:NamedPizza* and **P:hasTopping** only
(*P:MozzarellaTopping* or *P:PeperoniSausageTopping* or *P:TomatoTopping*)

P:AmericanPizza subClassOf *P:NamedPizza* and
P:hasTopping some *P:MozzarellaTopping* and
P:hasTopping some *P:PeperoniSausageTopping* and
P:hasTopping some *P:TomatoTopping*

These axioms were transformed into *PIZZA*$^+$ axioms by using only BTL2 object properties:

P$^+$*:AmericanPizza* subClassOf *P*$^+$*:NamedPizza* and
(**btl2:hasComponentPart** only (*P*$^+$*:MozzarellaTopping* or
P$^+$*:PeperoniSausageTopping* or *P*$^+$*:TomatoTopping* or *P*$^+$*:PizzaBase*)) and
(**btl2:hasComponentPart** some *P*$^+$*:TomatoTopping*) and
(**btl2:hasComponentPart** some *P*$^+$*:MozzarellaTopping*) and
(**btl2:hasComponentPart** some *P*$^+$*:PeperoniSausageTopping*)

We had to add some classes that were not included in the original ontologies in order to replace old *ad hoc* object properties by more precise constructs. The object property **P:hasCountryOfOrigin** had remained undefined in PIZZA. Common sense suggests that **P:hasCountryOfOrigin** is not meant to relate an individual pizza with the country where it was produced. In contrast to the wine and fruit, pizzas are not shipped around between countries. Therefore, we assume that an *P*$^+$*:AmericanPizza* is a *P*$^+$*:Pizza* that has been produced according to a recipe which was created in America. This leads us to create a class *P*$^+$*:Recipe*, which is a *btl2:Plan* and therefore a *btl2:InformationObject*. *btl2:Plan* represents a thing in which several targeted steps (processes) are to be performed to reach a planned goal. A *btl2:Plan* can only be realized by some *btl2:Process*. It is a piece of information that exists independently of a particular material carrier. *P*$^+$*:Recipe* has the subclass *P*$^+$*:PizzaRecipe* with following definition:

P$^+$*:PizzaRecipe* isEquivalentTo *P*$^+$*:Recipe* and (**btl2:hasRealization** only
(*btl2:Action* and (**btl2:hasOutcome** some *P*$^+$*:Pizza*)))

In BTL2 an action is a process that has an agent [12]. Preparing food or creating a recipe should therefore be placed under the class *btl2:Action:*

btl2:Action isEquivalentTo *btl2:Process* and **btl2:hasAgent** some *owl:Thing*

Accordingly:

P⁺*:PreparingFoodAction* subClassOf *btl2:Action* and **btl2:hasOutcome** some *P*⁺*:Food*

P⁺*:PreparingPizza* EquivalentTo
P⁺*:PreparingFoodAction* and **btl2:hasOutcome** some *P*⁺*:Pizza*

P⁺*:AmericanPizzaRecipe* EquivalentTo *P*⁺*:PizzaRecipe* and (**btl2:isOutcomeOf** some
(*P*⁺*:CreatingRecipe* and (**btl2:isIncludedIn** value **P⁺:NorthAmerica**)))

By these means we can redefine *P*⁺*:AmericanPizza* as follows:

P⁺*:AmericanPizza* EquivalentTo *P*⁺*:Pizza* and
btl2:hasComponentPart only (*P*⁺*:MozzarellaTopping* or
P⁺*:PeperoniSausageTopping* or *P*⁺*:PizzaBase* or *P*⁺*:TomatoTopping*) and
btl2:hasComponentPart some *P*⁺*:MozzarellaTopping* and
btl2:hasComponentPart some *P*⁺*:PeperoniSausageTopping* and
btl2:hasComponentPart some *P*⁺*:TomatoTopping* and
btl2:isOutcomeOf some
(*P*⁺*:PreparingPizza* and (**btl2:isRealizationOf** some *P*⁺*:AmericanPizzaRecipe*))

The class *P*⁺*:Spiciness* in PIZZA⁺ describes a value partition for the classes *P*⁺*:Hot*, *P*⁺*:Medium* or *P*⁺*:Mild*. It is a *btl2:ObjectQuality* with the superclass *btl2:Quality*.

4.2.2 WINE&FOOD⁺

To redesign *WINE&FOOD* under BTL2, we aligned its class structure to BTL2 and replaced the *WINE&FOOD* object properties by BTL2 ones. E.g., the relations that indicate the qualities of a wine, *viz.* **W:hasBody**, **W:hasColor**, **W:hasFlavor** and **W:hasSugar** (in the sense of having a sweet flavour) were replaced with **btl2:isBearerOf**. Furthermore, new labels were added to several *W*⁺*:AnimalOrganismPart* classes to avoid ambiguity in an out-of-context situation. The class *W*⁺*:ConsumableThing* with the subclasses *W*⁺*:EdibleThing*, *W*⁺*:Meal*, *W*⁺*:MealCourse*, *W*⁺*:PotableLiquid* were defined as subclasses of *btl2:PolyMolecularCompositeEntity*.
In WINE&FOOD some classes were subsumed by *W:EdibleThing*, e.g., *W:Fowl*, *W:Meat* and *W:Seafood*. The re-modelling requires changes in view of the classification, e.g., the classes *W*⁺*:Fowl*, *W*⁺*:Meat* and *W*⁺*:Seafood* are now subclasses of both *W*⁺*:AnimalOrganismPart* and *W*⁺*:EdibleThing*, the latter being defined as follows:

W⁺*:EdibleThing* subClassOf *W*⁺*:ConsumableThing* and
(**btl2:isBearerOf** some (*btl2:Disposition* and
(**btl2:hasRealization** only *W*⁺*:BeingEatenByHumans*)))

In order to distinguish between countable and mass entities, *W:Fruit* was re-labelled *W*⁺*:FruitUnit* to denote a countable thing; also its classification was changed:

W⁺*:FruitUnit* subClassOf *btl2:PolyMolecularCompositeEntity*

Not all fruits are edible, therefore we added:

W⁺*:EdibleFruitUnit* equivalentTo *W*⁺*:EdibleThing and W*⁺*:FruitUnit*

W⁺*:NonSweetFruitUnit* subClassOf *W*⁺*:EdibleFruitUnit*

W⁺:SweetFruitUnit subClassOf *W⁺:EdibleFruitUnit*

W⁺:MixedFruitUnit subClassOf *W⁺:EdibleFruitUnit*

W:WineDescriptor is re-modelled as subclass of *btl2:Quality: W⁺:WineDescriptor* with the subclasses *W⁺:WineColor* and *W⁺:WineTaste* :

W⁺:WineDescriptor subclassOf **btl2:inheresIn** some *W⁺:Wine*

W⁺:WineDescriptor subclassOf **btl2:inheresIn** only *W⁺:Wine*

The object property **W:madeFromGrape** was dropped and reified by a new subclass of *btl2:Action*, *viz. W⁺:WineMaking*:, a class representing the process of which wine is an outcome:

W⁺:WineMaking isEquivalentTo *btl2:Action* and
(**btl2:hasOutcome** some *W⁺:Wine*) and **btl2:hasPatient** some *W⁺:Grape*)

W⁺:Wine subClassOf (**btl2:isOutcomeOf** some *W⁺:WineMaking*)

The meaning of **W:madeFromFruit** was represented similarly by *W⁺:FruitJuiceMaking,* defined as a subclass of *btl2:Action*:

W⁺:FruitJuiceMaking isEquivalentTo *btl2:Action* and
(**btl2:hasOutcome** some *W⁺:FruitJuice*) and (**btl2:hasPatient** some *W⁺:Fruit*)

Hence:

W⁺:FruitJuice isEquivalentTo *W⁺:PotableLiquid* and
btl2:isOutcomeOf some *W⁺:FruitJuiceMaking*

Individuals like **`W:CabernetFrancGrape`**, **`W:GamayGrape`**, **`W:MalbecGrape`**, etc. are now typed as *W⁺:WineGrape*, whereas geographical regions are represented as classes like *W⁺:AlsaceRegion*, *W⁺:AustralianRegion*. The underlying definitions are of the following type:

W⁺:AlsaceRegion isEquivalentTo *W⁺:GeographicRegion* and
btl2:isIncludedIn value **`W⁺:Alsace`**

Wine attributes like **`W:Red`**, **`W:Rose`**, **`W:Full`**, **`W:Delicate`**, **`W:Dry`**, etc. are classes in W⁺ and descendants of *btl2:Quality*.

5. Discussion

The first aim of this paper was to evaluate the two popular educational ontologies PIZZA and WINE&FOOD with respect to common quality criteria. We have found that the two ontologies were not designed for interoperability and did not make any effort to align their classes with any foundational ontology.

Whereas the PIZZA creators took care in good documentation, which partly elucidates the design decisions taken, WINE&FOOD reflects a pure computer-science approach towards ontologies as knowledge-representation devices for certain purposes, reflecting the notion of ontologies as "conceptualisations" [13], which characterised the discourse in the 1990s. This is concretised by ontology specialists' recommendation that the decision "class or instance" is just a matter of convenience [2], to be driven by what the application requires. This postmodern flavour was then heavily attacked by proponents of "reality representation" like Barry Smith [14], who also were fierce advocates of foundational ontologies. In the meanwhile, this controversy has cooled down, mainly because interoperability-driven standardisation gained more importance

in Applied Ontology, which prohibits overly idiosyncratic approaches anyway. The creation of BTL2 and the development of GoodOD were driven by the pragmatic motivation to simplify ontology construction, predominantly in the realm of biology and medicine. This is also the reason for its underlying naïve realism, which set it in close proximity to the Basic Formal Ontology (BFO) and the OBO Relation Ontology (RO).

Second, though PIZZA and WINE&FOOD fared badly in this evaluation, we found that they can be re-designed such that their content is aligned with a pre-existing foundational ontology. Our redesigned ontologies demonstrate a successful proof of concept for such a redesign. In particular, the inventory of object properties contained in BTL2 proved to be sufficient for the domains covered. Idiosyncratic object properties in the source ontologies could be either replaced by existing BTL2 ones, or reconstructed by introducing reifying process classes. A considerable benefit of the avoidance of introducing new primitive object properties is the need to re-think the meaning of ambiguous and possibly fuzzy predicates like **hasCountryOfOrigin**, an object property that has turned out to have different meanings when applied to pizzas and to wines, respectively, which requires a thorough elucidation exceeding the expressivity of a single binary OWL object property.

Thirdly, we intended to compare the educational value of the original and redesigned ontologies. PIZZA and WINE&FOOD describe material things, their constitution, qualities and processes they are involved in. In the terms of Karl Popper [15], these are "first-world" entities for which there is a large consensus that a realistic approach is adequate. Due to everyone's familiarity with food, this subject matter constitutes an easy and straightforward entry point to ontology engineering for a broad range of learners. Domains that require to model mental, social, or legal entities would spark more controversy about upper-level divisions, provide less clear criteria for distinguishing between individuals and repeatables, and would therefore not be well served by a foundational ontology like BTL2.

Regarding the two main learning objectives of (i) mastering OWL and description logics, on the one hand, and (ii) understanding foundational ontologies, on the other hand, the answers are clearly split. In the original Manchester tutorial for PIZZA, the focus was set (besides developing skills in using Protégé) on understanding the whole breadth of description logics including all technical intricacies connected with automated reasoning. This scope is clearly narrowed down by our redesigned PIZZA and WINE&FOOD ontologies, because the very point of using a highly constraining foundational ontology like BTL2 is to limit modellers' degrees of freedom. This means for instance, that the manipulation of the "R-Box", i.e., the creation and customisation of object properties, with their algebraic features, hierarchies, domain and range restrictions and CGIs is no longer admissible (nor necessary, as long as the premise holds that all necessary ontological relations are already provided by the upper-level ontology).

On the other hand, with PIZZA$^+$ and WINE&FOOD$^+$ the ontology engineers' curriculum now includes an in-depth understanding of a foundational ontology. This prevents naïve modelling decisions such as introducing *ad hoc* primitives like **hasCountryOfOrigin**. An ontologically proper representation of such predications is, however, not trivial and requires at least as many in-depth skills and knowledge in Formal Ontology as logic knowledge is needed for mastering the syntax and semantics of the representational language.

Taken singly as well in combination, these ontologies can be applied for educational purposes, accompanied by a new tutorial which follows a clear educational strategy: After an introduction to tools (editor, reasoner), language, upper-level ontology and naming conventions, the modelling tasks are guided by textual characterisations of the entities and their relationships. This could be performed in two parallel groups, one working on the pizza domain and one working on the wine domain. The curriculum can then be concluded by a fusion of these two ontologies, demonstrating the interoperability of the two ontologies and thus proving the value of principled ontology design for interoperability of separately created ontology artefacts.

6. Conclusion

The educational ontologies PIZZA and WINE&FOOD have been widely used to teach ontology methods and tools. However, the two ontologies have remained largely unchanged for many years, despite increasing awareness of foundational principles (GoodOD) for good ontology design. Recognizing that the two popular educational ontologies PIZZA and WINE&FOOD were lagging behind design principles and do not commit to any foundational ontology, we analyzed and re-designed them by adhering to upper level ontology constraints provided by the foundation ontology BTL2. As a result, the redesign required clarifying the ontological commitment of the PIZZA and WINE&FOOD classes by assigning them to top-level ontology classes. The ongoing redesign effort[4] is expected to yield a harmonized combined PIZZA&WINE&FOOD^{+} ontology.

As much as we acknowledge the value of educational ontologies, especially of the well-documented PIZZA, we are convinced that the education of ontologists must not be limited to training in the use of tools and the understanding of their logical foundations. Foundational ontologies ought to play a central role in the future of Applied Ontology as an engineering discipline; and this needs to be addressed by appropriate educational ontologies. The redesigned PIZZA^{+} and WINE&FOOD^{+} might be a first step in this direction. A detailed investigation of the reasoning results of the original and redesigned ontologies, their assessment against competency questions as well as benchmarking with common DL reasoners will be described elsewhere.

[4] Documented in the GitHub repository: http://purl.org/biotop

References

[1] R.J. Brachman, D.L. McGuinness, P.F. Patel-Schneider, L.A. Resnick, A. Borgida, Living with CLASSIC: When and how to use a KL-ONE-like language, *Principles of semantic networks* (1991), 401–456.

[2] N.F. Noy, D.L. McGuinness, Ontology development 101: A guide to creating your first ontology, Stanford knowledge systems laboratory technical report KSL-01-05; *Stanford medical informatics technical report* SMI-2001-0880, 2001. http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

[3] M.K. Smith, C. Welty, D.L. McGuinness, OWL Web ontology language guide, *W3C Recommendation*, 10 February 2004. http://www.w3.org/TR/owl-guide/.

[4] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, C. Wroe, OWL Pizzas: Practical experience of teaching OWL-DL: common errors & common patterns, *Lecture Notes in Computer Science* **3257** (2004), 63–81.

[5] B. Smith, Applied ontology: A new discipline is born, *Philosophy Today* **12**. 29 (1998), 5–6.

[6] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*, 2nd Edition, Cambridge University Press, 2007.

[7] S. Schulz, D. Seddig-Raufie, N. Grewe, J. Röhl, D. Schober, M. Boeker, L. Jansen, *Guideline on Developing Good Ontologies in the Biomedical Domain with Description Logics*, University Medical Center, Freiburg, 2012. http://www.purl.org/goodod/guideline

[8] L. Jansen, S. Schulz, The ten commandments of ontological engineering. H. Herre et al. *Ontologies in Biomedicine and the Life Sciences* (OBML 2011), 2011/01: IMISE-Reports, Universität Leipzig, 2011.

[9] D. Schober, B. Smith, S.E. Lewis, W. Kusnierczyk, J. Lomax, C. Mungall, C.F. Taylor, P. Rocca-Serra, S.A. Sansone, Survey-based naming conventions for use in OBO Foundry ontology development, *BMC Bioinformatics* **10.1** (2011),125.

[10] N. Guarino, C.A. Welty, An Overview of OntoClean. In S. Staab and R. Studer (eds) *Handbook on Ontologies*, 201–220. Springer, Berlin/Heidelberg, 2009.

[11] S. Schulz, M. Boeker, C. Martinez-Costa, The BioTop family of upper level ontological resources for biomedicine, *Studies of Health Technology and Informatics* **235** (2017), 441–445.

[12] D. Seddig-Raufie, J. Jansen, D. Schober, M. Boeker, N. Grewe, S. Schulz, Proposed actions are no actions: re-modeling an ontology design pattern with a realist top-level ontology, *Journal of Biomedical Semantics* **3 Suppl. 2** (2012) S2.

[13] T.R. Gruber, Towards principles for the design of ontologies used for knowledge sharing, In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Dordrecht, 1993.

[14] B. Smith, Beyond concepts: ontology as reality representation, *Proceedings of the third international conference on formal ontology in information systems* (FOIS 2004), 73–84. IOS Press, Amsterdam, 2004.

[15] K. Popper, Three worlds, *The Tanner lecture on human values*, Univ. Michigan, 1978. http://tannerlectures.utah.edu/_documents/a-to-z/p/popper80.pdf.