[redacted]

Coursework Project (Part 1): Ontology Modelling (Task OWL)

[redacted]

Member 1: Kai Sun Anson Lam (Anson)

[redacted]

Task OWL.A: Ontology modelling in Protégé

Classes

To delineate our ontology, we begin with four primary classes, each outlined as follows:

Location: This encompasses subclasses Address, City, Country, Postcode, and State, in where each restaurant is located. Additionally, it includes the subclass Restaurant since a restaurant is considered as a type of location. Within this subclass, we have two further subclasses: NamedRestaurant, allowing for differentiation based on names, and RestaurantCategory, which includes subclasses like AmericanRestaurant, ItalianRestaurant, etc. (We omitted various categories to maintain a compact ontology for the coursework.)

MenuItem: This class involves a subclass called Food, (Each food item is regarded as a type of menu item in our consideration.) specifically focusing on pizzas for this coursework. It includes subclasses like NamedPizza, with examples such as American and Margherita, and blank nodes like VegetarianPizza and NonVegetarianPizza to demonstrate the disjoint functionality between vegetarian and non-vegetarian pizzas. At the same level as the Pizza class, we have PizzaBase and PizzaTopping subclasses. For instance, PizzaBase further divides into DeepPanBase and ThinAndCrispyBase, while PizzaTopping includes different toppings such as CheeseTopping, exemplified by MozzarellaTopping. (and of course many other toppings.)

ValuePartition: This class contains the Size subclass, reflecting the various sizes available for pizzas in the CSV file, including LargeSize, MediumSize, and SmallSize. In fact, Size can also be applied to other types of food added in the future.

ItemDetails: This class demonstrates attributes such as Currency, ItemDescription, and ItemValue.

In terms of annotations, labels have been assigned to most classes, for example, ItemValue is labeled as "ItemValue," "PriceValue," and "Value." Similarly, the Location class is labeled as “Location” and "Place," among others.

Properties and Property Restrictions

In the Object Properties section, seven main properties and some sub-properties are defined:

hasIngredient: hasIngredient is the inverse of the isIngredientOf and includes sub-properties hasPizzaBase and hasPizzaTopping. Both the domain and range of hasIngredient are the Food class, reflecting that the subject and object of this property must be food. Also, hasIngredient exhibits a transitive characteristic. Both hasPizzaBase and hasPizzaTopping sub-properties have the same domain as the Pizza class. However, while the range of hasPizzaBase is the PizzaBase class, the range of hasPizzaTopping is the PizzaTopping class.

isIngredientOf: isIngredientOf is the inverse of the hasIngredient and includes sub-properties isPizzaBaseOf and isPizzaToppingof. Both the domain and range of isIngredientOf are the Food class, reflecting that the subject and object of this property must be food. Also, isIngredientOf exhibits a transitive characteristic. Both isPizzaBaseOf and isPizzaToppingof sub-properties have the same range as the Pizza class. However, while the domain of isPizzaBaseOf is the PizzaBase class, the range of isPizzaToppingof is the PizzaTopping class.

hasItemDetails: hasItemDetails signifies that each MenuItem (domain) includes certain ItemDetails (range), comprising three sub-properties: hasCurrency, hasItemDescription, and hasItemValue. While all these three sub-properties also have MenuItem as domain, the ranges are either Currency, ItemDescription or ItemValue depending on each sub-property.

hasMenuItem: hasMenuItem has the domain NamedRestaurant and range MenuItem. While the range is obvious based on the property name, NamedRestaurant is considered as the domain, instead of Restaurant or CategorisedRestaurant. For instance, it makes more sense that PizzaExpress (an instance not in the current ontology) has PizzaDaAS as MenuItem, because it might be too general to say that that ItalianRestaurant (i.e. all restaurants classified as this type) has PizzaDaAS as MenuItem.

hasRestaurantCategory: NamedRestaurant as domain and CategorisedRestaurant as ranges suggests that the subject and object of the property must be the two classes respectively.

hasSize: hasSize only has Size as range but does not have a domain. This implies that different type of food or even things can use this property, but the object of it is limited to Size as suggested by the property name.

isLocatedIn: With Location as both domain and range, a location can be location in another location. There is also the transitive property. For example, a restaurant can be located in an address; an address can be located in a city; a city can be located in a country.

New Restaurant and Pizza Concepts

For the new restaurant concept, "PizzeriaDaAS" falls under the "CategorisedRestaurant" subclass. For the new pizza concept, "PizzaDaAS" is categorized under the "NamedPizza" subclass, customized to include only toppings such as BBQ Sauce, Ham, Mozzarella, Pepperoni, Pineapple, or Tomato.

Task OWL.B: Ontology modelling on “Paper”

For subtasks of Task OWL.B, axioms are followed by descriptions in natural language.

Subtask OWL.B.1: An axiom with an atomic subsumption.

- PremierLeagueTeam ⊑ FootballTeam
- All Premier League teams are football teams.

Subtask OWL.B.2: An axiom with an universal restriction.

- PremierLeagueTeam ⊑ ∀hasPlayer.PremierLeaguePlayer
- A Premier League team is consisted of only Premier League players.

Subtask OWL.B.3: An axiom with an existential restriction.

- BookedPlayer ⊑ ∃isGivenCard
- A booked player is given some cards.

Subtask OWL.B.4: An axiom with a union.

- MatchResult ⊑ Win ⊔ Draw ⊔ Loss
- A match result is either win, draw or loss.

Subtask OWL.B.5: An axiom with an intersection.

- CentralAttackingMidfielder ⊑ CentralMidfielder ⊓ AttackingMidfielder
- A central attacking midfielder is both central and attacking midfielder.

Subtask OWL.B.6: An equivalence axioms with a cardinality restriction.

- HattrickPlayer ≡ $\geq_3$hasGoal.GoalScored
- A hattrick player is a player with at least 3 goals scored.

Subtask OWL.B.7: An axiom stating two concepts cannot have a common instance.

- Win ⊓ Loss ⊑ ⊥
- A team cannot win and lose a match at the same time.

Subtask OWL.B.8: A property chain axiom.

- ownsClub ◦ consistedOfPlayer ⊑ employsPlayer
- If a person owns a football club which is consisted of players, the person employs the players.

Subtask OWL.B.9: A role assertion axiom and a valid inverse.

- plays_for(arsenal_players, arsenal)
  signs(arsenal, arsenal_players)
- Arsenal players plays for Arsenal means that Arsenal signs the Arsenal players.

Subtask OWL.B.10: A class assertion axiom where the class is complex.

- locatedInLondon ⊓ ∀hasPlayer.PremierLeaguePlayer (arsenal)
- Arsenal is football team, which is located in London and has only Premier League players.

Subtask OWL.B.11: (optional) A combination of axioms that makes the ontology to be outside OWL 2.