University of Prishtina “Hasan Prishtina”

Faculty of Electrical and Computer Engineering

Project documentation

Subject: Semantic Web

Project title: Food Allergy Ontology

Professor/Assistant name	Name & surname / email address of students	
Prof. Dr. Ing. Lule AHMEDI Msc. Adrian YMERI	1. Erza Gashi	erza.gashi10@student.uni-pr.edu
	2. Mihrije Kadriu	mihrije.kadriu@student.uni-pr.edu
	3. Rrezearta Thaqi	rrezearta.thaqi@student.uni-pr.edu

Content

List of Figures

List of Abbreviations

OWL - Web Ontology Language

SPARQL - SPARQL Protocol and RDF Query Language

SWRL - Semantic Web Rule Language

1. Introduction

The Semantic Web, known as Web 3.0, is an extension of the World Wide Web, which is a mesh of data that are associated in such a way that they can easily be processed by machines instead of human operators [1]. An ontology is one of the components of a semantic web implementation, where ontology describes a technical term denoting an artifact that is designed for a purpose, which is to enable the modeling of knowledge about some domain and the relationships that hold between them. The role of ontologies in Semantic Web is to facilitate data organization and integration [2]. Food and drink can influence the risks of health conditions such as heart disease, diabetes, allergies, obesity, and others. By controlling what and how much to eat as well as what not to eat, we can maximize and decrease usage of unhealthy ingredients or compounds such as oxidants. The main objective of this project is the building an ontology for food classification based on different kinds of allergies, diets and origins.

2. Problem statement

The purpose of this project is to build an ontology through which classification of food is done based on different kinds of allergies, diets and their origin. The main purpose of this ontology is to suggest food/recipes for people that are allergic, that want to lose or gain weight and also to give different statistics about food.

3. Scope of the project

There are similar types of ontologies that were developed earlier about food, but our ontology is specialized for our purpose.
This ontology was developed especially for allergic people and for those that want to lose or gain weight. The definition of this ontology is based on different websites for food allergies, diets and different statistics that are online.

4. Ontology architecture

The development of the project is divided into four parts that will be described below as part of the project phases. There will also be a description of the concepts used.
As an overview, below (in Fig.1.) are the metrics of the ontology:

Ontology metrics:

Metrics

Axiom	5693
Logical axiom count	1091
Declaration axioms count	829
Class count	208
Object property count	338
Data property count	32
Individual count	117
Annotation Property count	138

Figure 1. Metrics of Food Allergy Ontology

5. First part

Building an ontology of the domain

5.1. The definition of classes hierarchy

The hierarchy of classes is defined based on data we have found on different websites for food allergies and diets and also based on our knowledge about food. In this section there will be described only some of the **main classes** of this ontology.

Agent - This class has three subclasses:

1. Person
2. SoftwareAgent
3. Organization

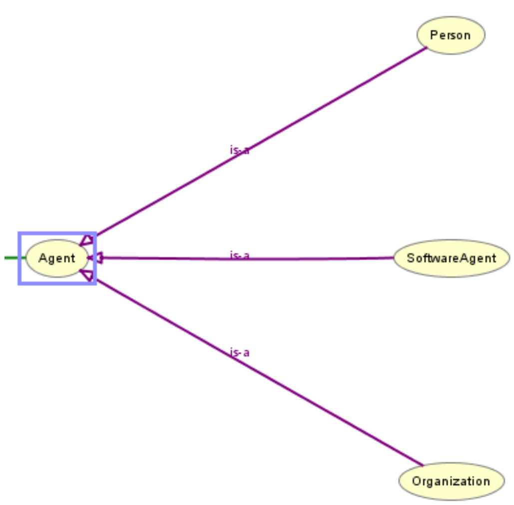


Figure 2. Agent class

characteristic - A food has two characteristics:
1. texture
2. flavor

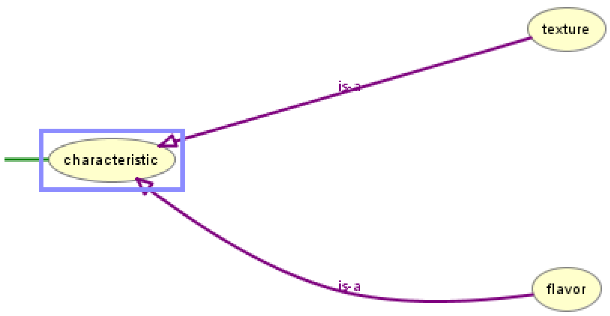


Figure 3. Characteristic class

Characteristic - This class has three subclasses:
1. Parameter
2. System Characteristic
3. User Characteristic

The User Characteristic subclass has two subclasses where each of them has four subclasses:

1. Positive User Characteristic
    1. Allowed Food Characteristic
    2. In Budget Characteristic
    3. In Diet Characteristic
    4. Liked Food Characteristic
2. Negative User Characteristic
    1. Allergic Food Characteristic
    2. Disliked Food Characteristic
    3. Forbidden Food Characteristic
    4. Out Budget Food Characteristic

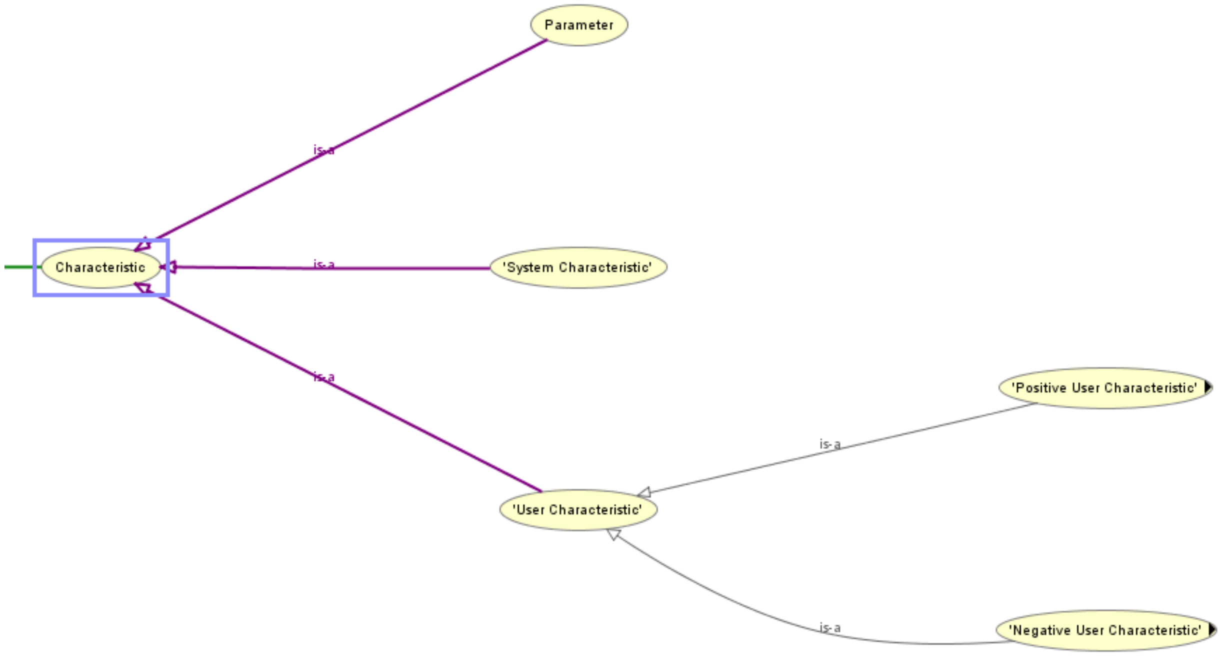


Figure 4. Characteristic class

Collection - This class has only one subclass:
1. Ordered Collection

Figure 5. Collection class

Diet - This class doesn't have any subclass. It contains individuals with different kinds of diets.

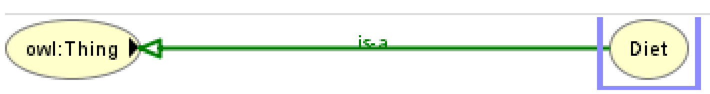


Figure 6. Diet class

Food - This class contains all the kinds of food as the subclasses below (19 subclasses):
1. Chemical food product
2. High Budget Food
3. High Calorie Food
4. High Carb Food
5. High Fat Food
6. High Folate Food
7. High Protein Food
8. Low Budget Food

9. Low Calorie Food
10. Low Carb Food
11. Low Fat Food
12. Low Protein Food
13. Low Sodium Food
14. Medium Budget Food
15. Primary food commodity of animal origin
    1. amphibian and reptile
    2. aquamatic animal product
    3. invertebrate animal
    4. mammalian product
    5. poultry product

These subclasses contain other subclasses that are food of that kind.

16. Primary food commodity of fungal origin
    1. mushroom
    2. yeast
17. Primary food commodity of plant origin
    1. fruit
    2. grass
    3. herb and spice
    4. nut and seed
    5. vegetable

These subclasses contain other subclasses that are food of that kind.

18. Processed food of animal origin
    1. derived edible product of animal origin
    2. manufactured food (multi-ingredient) of animal origin
    3. manufactured food (single-ingredient) of animal origin
    4. secondary food commodity of animal origin

These subclasses contain other subclasses that are food of that kind.

19. Processed food of plant origin
    1. derived product of plant origin
    2. manufactured food (multi-ingredient) of plant origin
    3. manufactured food (single-ingredient) of plant origin
    4. secondary food commodity of plant origin

These subclasses contain other subclasses that are food of that kind.

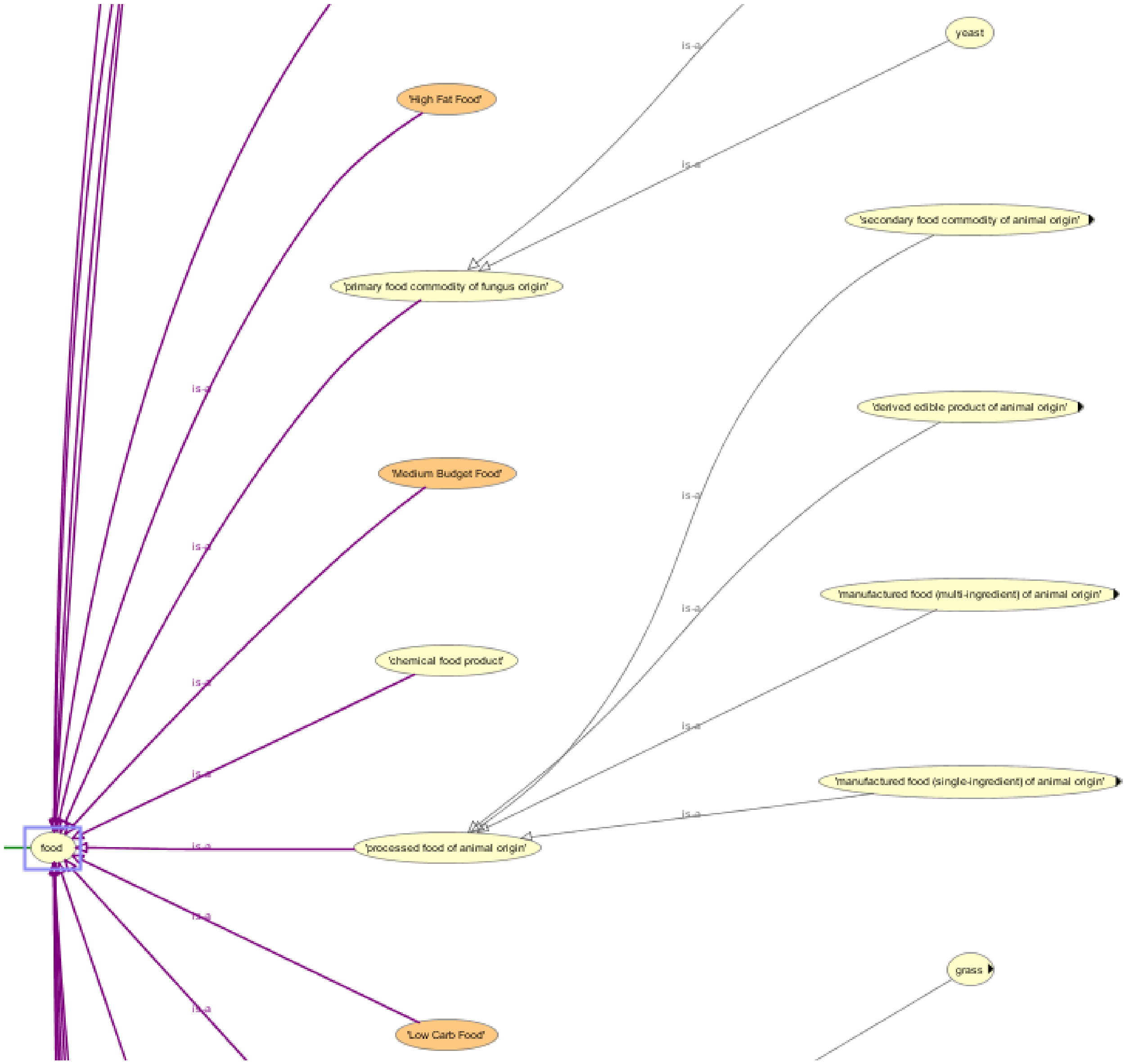


Figure 7. One part of food class hierarchy

<u>Goal</u> - This class doesn't have any subclass. It contains individuals with different kinds of goals that a user can have.

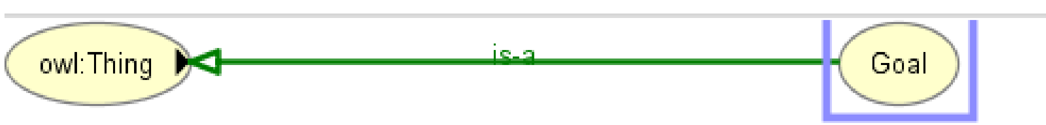


Figure 8. Goal class

ingredient - This class doesn't have any subclass. Ingredient is part of a food.

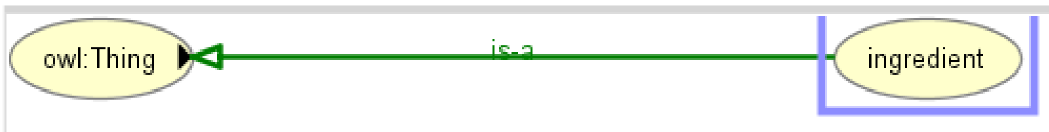


Figure 9. Ingredient class

meal - This class doesn't have any subclass. It contains individuals with different kinds of meals.

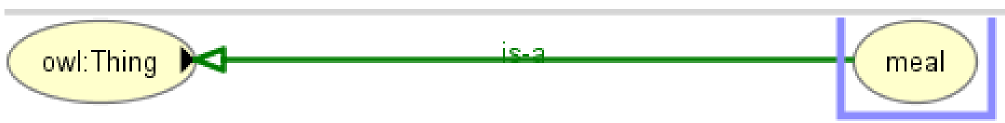


Figure 10. Meal class

measurement - Each food has nutrient values like calorie, fat, sugar, etc. This class has three subclasses:

1. Nutrient Measurement
2. Temperature measurement
3. Time measurement

The Nutrient Measurement subclass has six subclasses:

1. Calorie
2. Carbohydrate
3. Fat
4. Protein
5. Sodium
6. Sugar

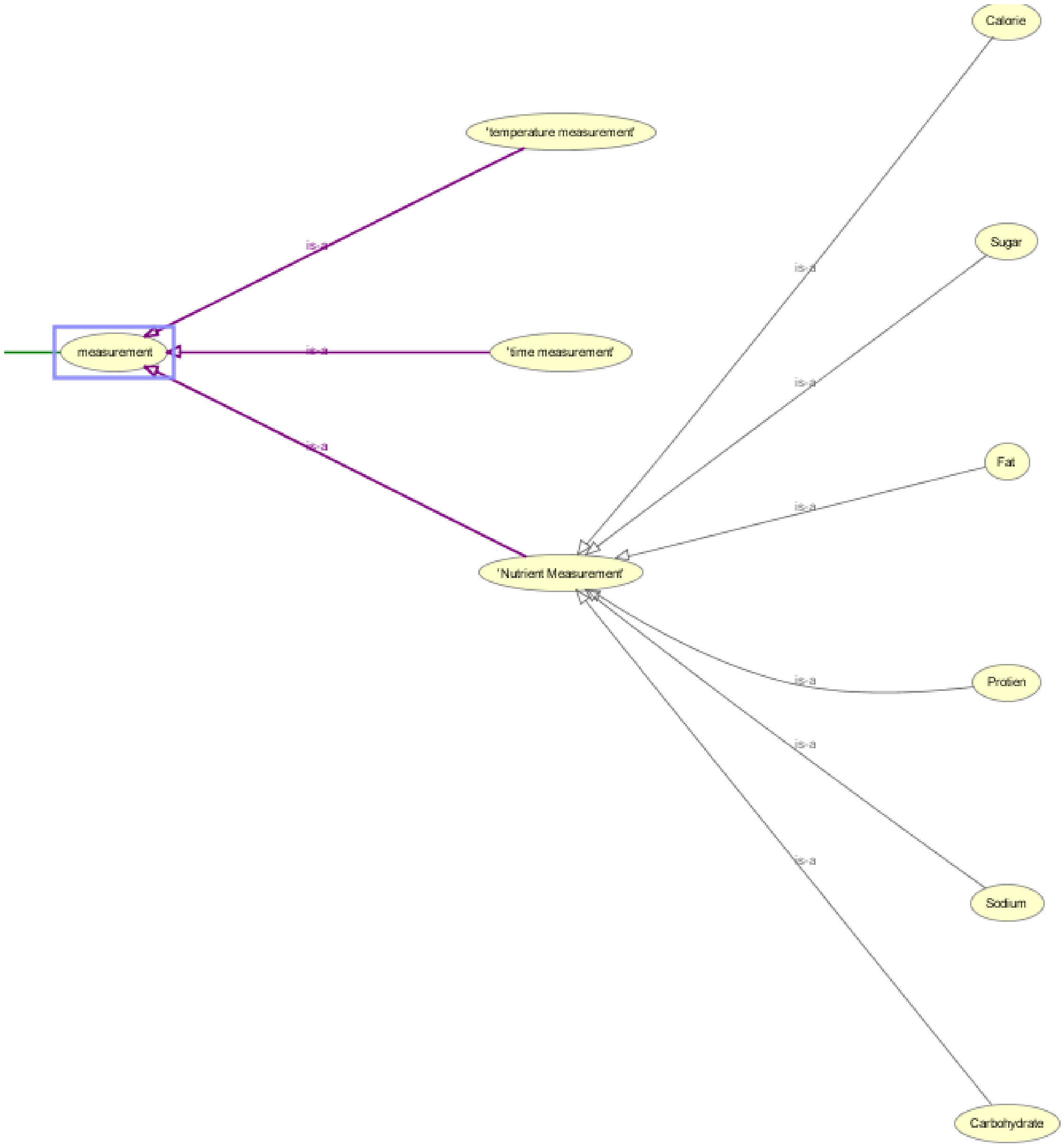


Figure 11. Measurement class hierarchy

recipe - For each user there can be a recipe recommended. This class has six subclasses:
1. Breakfast recipe
2. Dinner recipe
3. High glycemic recipe
4. Less than an hour recipe
5. Lunch recipe
6. Side recipe

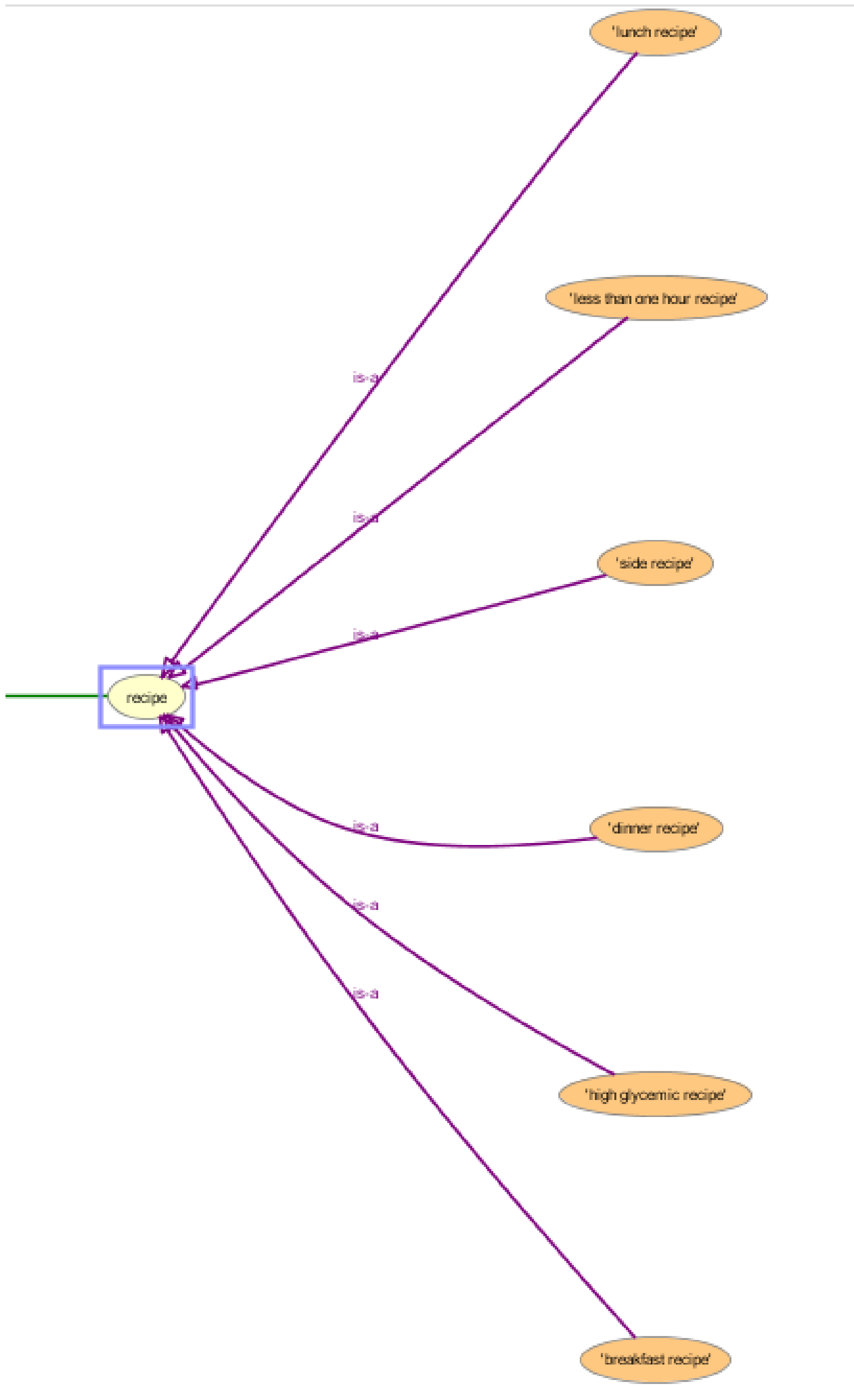


Figure 12. Recipe class hierarchy

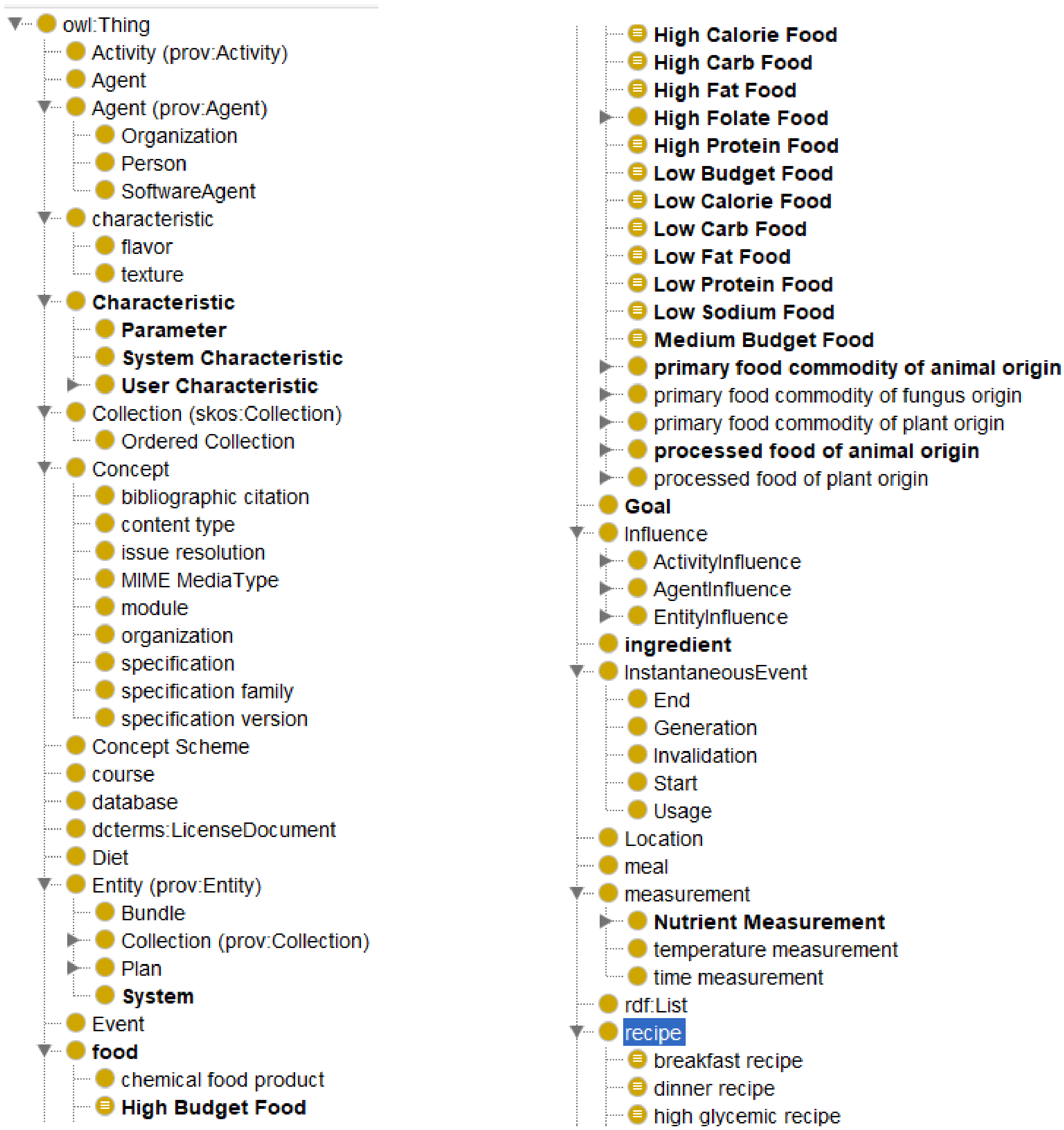
The metrics of the classes defined in our ontology are presented below:

Class axioms

SubClassOf	224
EquivalentClasses	21
DisjointClasses	8
GCI count	0
Hidden GCI Count	15

Figure 13. The metrics of the classes

The hierarchy of all classes is shown in the figures below:



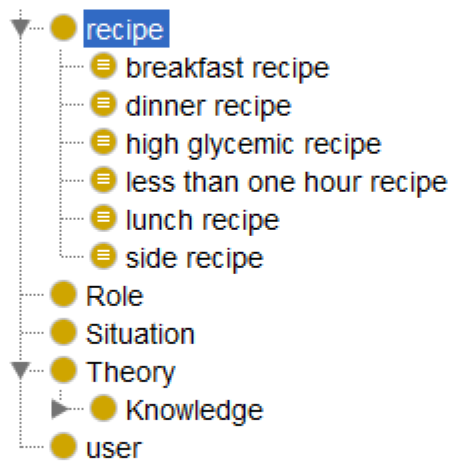


Figure 14. The hierarchy of all classes of the ontology

5.1.1. Disjoint classes

Two classes in an ontology are disjoint if they cannot share an instance, regardless of how the classes are interpreted. The cases where the disjointed classes are used, are shown below:

Figure 15. Disjoint classes

5.2. Object and data properties of classes

Properties are binary relations on individuals, properties link two individuals together.

There are two types of properties:

1. **Object properties** – that link two objects and
2. **Data properties** – that link an object with atomic data.

Metrics of the properties of objects and their characteristics that are used are presented below:

Object property axioms

SubObjectPropertyOf	**292**
EquivalentObjectProperties	0
InverseObjectProperties	**104**
DisjointObjectProperties	**1**
FunctionalObjectProperty	**15**
InverseFunctionalObjectProperty	**8**
TransitiveObjectProperty	**34**
SymmetricObjectProperty	**42**
AsymmetricObjectProperty	**2**
ReflexiveObjectProperty	**3**
IrrefexiveObjectProperty	**3**
ObjectPropertyDomain	**58**
ObjectPropertyRange	**61**
SubPropertyChainOf	**16**

Figure 16. Metrics of object properties

There are seven main characteristics that object properties can have:

1. **Functional** - the property is *Functional*, if that for any given individual, the property can have at most one value [3].
2. **Inverse Functional** - the property is *InverseFunctional*, if the inverse property of the selected property (whether it is explicitly declared or not) is Functional. If multiple individuals are specified as incoming values for the property then these values will be inferred to denote the same object [3].
3. **Transitive** -the property is *Transitive* if individual **x** is related to individual **y**, and individual **y** is related to individual **z**, then individual **x** will be related to individual **z**. [3]
4. **Symmetric** - the property is *Symmetric* which means that the property has itself as an inverse, so if individual **x** is related to individual **y** then individual **y** must also be related to individual **x** along the same property [3].
5. **Asymmetric** - the property is *Asymmetric*, which means that if individual **x** is related to individual **y** then individual **y** is not related to individual **x** along the same property.
6. **Reflexive** - the property is *Reflexive*, which means this property causes every single individual to be related to itself via that property [3].
7. **Irreflexive** - the property is *Irreflexive,* which means that an individual cannot be related to itself via that property [3].

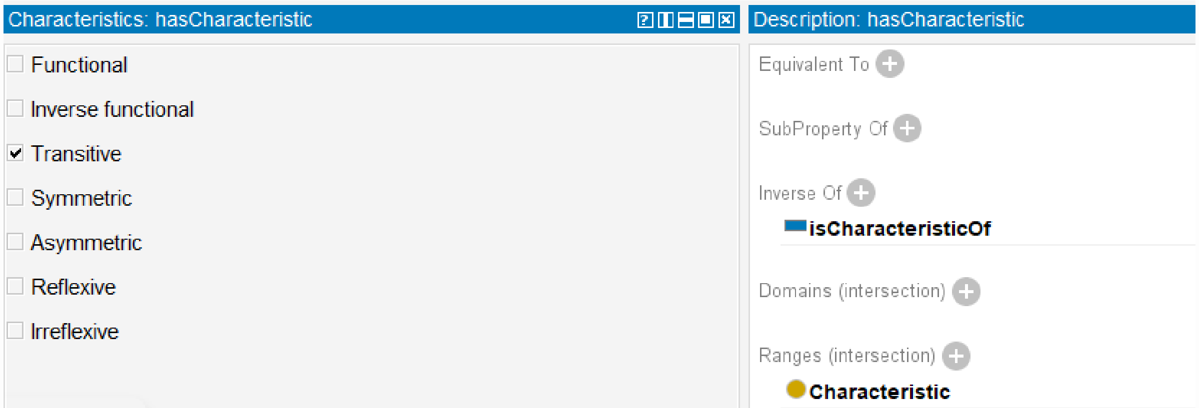


Figure 17. Transitive object property

From the Figure 17 above it can be seen that the object property *hasCharacteristic* is transitive and it's the inverse of *isCharacteristicOf*. This property is used for foods.

Metrics of data properties and their characteristics that are used are presented below:

Data property axioms

SubDataPropertyOf	5
EquivalentDataProperties	0
DisjointDataProperties	0
FunctionalDataProperty	1
DataPropertyDomain	11
DataPropertyRange	19

Figure 18. Metrics of data properties

Data properties can only be functional so that a property has only one value.

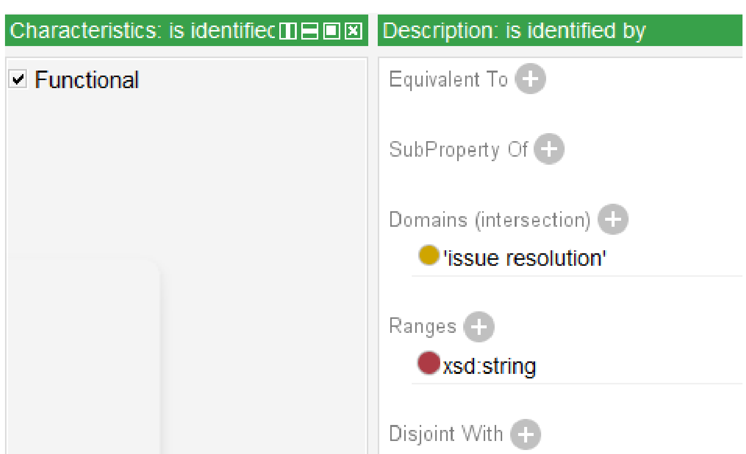


Figure 19. Functional data property

An issue can only have one identifier.

Object Property	Func	Sym	Inv Func	Trans	ASym	Refl	Irrefl	Domain	Range	Inverse
owl:topObjectProperty										
has conversion		✔								
alternateOf								Entity	Entity	
isBasedOn								Explanation	Theory	
isAccomplishedBy										
isBudgetFor										hasBudget
hasCharacteristic				✔					Characteristic	isCharacteristicOf
is version of								'specification version'	specification	'has specification version'
isDietFor										
hadUsage								Derivation	Usage	
hadGeneration								Derivation	Generation	
hadActivity								Delegation or Derivation or End or Start, Infl...	Activity	
is recommended for meal										
interacts		✔								
isParameterOfType										
is reflected in								'issue resolution'	'specification version'	reflects
componentOf										
forbiddenBy										forbids
influenced										wasInfluencedBy
wasInfluencedBy								Activity or Agent or Entity	Activity or Agent or Entity	influenced
isCharacteristicOf										hasCharacteristic
inBudgetFor										
hasExplanans								Explanation	Event	
substitutes for		✔								
reflects								'specification version'	'issue resolution'	'is reflected in'
is related to		✔								
hasExplanandum								Explanation	Event	
has flavor										
isInSeason										
opposes										isOpposedBy
isConceptualizedBy								Explanation	Agent	
hadRole								Association or InstantaneousEvent, Influence	Role	
recommendedBy										recommends
supports										isSupportedBy
hasSupportFor										
is transitively related to		✔		✔						
influencer								Influence	owl:Thing	
hasExplanationType										
isOpposedBy				✔						opposes
accomplishedGoal										
isAbout										
hasDiet										
atLocation								Activity or Agent or Entity or Instantaneous...	Location	
isAllergenFor								food		'is allergic to'
likedBy								food		likes
pertainsTo										
askedBy										
hasSetting								Event	Situation	
has member (skos:membe								Collection	Collection or Concept	
is in semantic relation with								Concept	Concept	
has specification version								specification	'specification version'	'is version of'
hasCondition								Explanation	Situation	
dislikedBy								food		dislikes
is recommended for cours										
isSupportedBy				✔						supports
has top concept								'Concept Scheme'	Concept	'is top concept in scheme'
hasBudget										isBudgetFor
has texture										
qualifiedInfluence								Activity or Agent or Entity	Influence	
hadPlan								Association	Plan	
has saved										
allowedBy										allows
hasGoal										

Figure 20. Object properties with their domain, range and reverse

Data Property	Func	Domain	Range
owl:topDataProperty	☐		
has cooking temperature	☐		
notation	☐		
has issue reference	☐	'issue resolution'	xsd:anyURI
value	☐	Entity	
has gluten	☐		
has description	☐	'issue resolution'	xsd:string
isLowProtein	☐		xsd:boolean
isLowCarb	☐		xsd:boolean
serves	☐		
hasPrice	☐		xsd:decimal
isLowSodium	☐		xsd:boolean
endedAtTime	☐	Activity	xsd:dateTime
invalidatedAtTime	☐	Entity	xsd:dateTime
has cook time	☐		
generatedAtTime	☐	Entity	xsd:dateTime
hasGrams	☐		
isLowCalorie	☐		xsd:boolean
requiresFact	☐		xsd:boolean
isLowFat	☐		xsd:boolean
isSupportive	☐		xsd:boolean
hasNutrientValue	☐	Characteristic	xsd:int
atTime	☐	InstantaneousEvent	xsd:dateTime
startedAtTime	☐	Activity	xsd:dateTime
isInternal	☐		xsd:boolean
has glycemic index	☐		
hasCalories	☐	Characteristic	xsd:int
is identified by	☑	'issue resolution'	xsd:string

Figure 21. Data properties with their domain and range

6. OWL2.0 Features

6.1. OWL2.0 - equivalent to, inverse of, property chain, data ranges, qualified cardinality restrictions, disjoint properties

OWL 2 adds new functionality with respect to OWL 1. Some of the new features are syntactic sugar (e.g., disjoint union of classes) while others offer new expressivity, including:
• equivalent to,
• inverse of,
• property chains,
• richer datatypes, data ranges,
• qualified cardinality restrictions,
• disjoint properties [4].

Equivalent to
Classes in our vocabulary may effectively refer to the same sets, and OWL provides a mechanism by which they are considered to be semantically equivalent [4]. For example, we use the term Person and Human interchangeably, meaning that every instance of the class Person is also an instance of class Human, and vice versa. Two classes are considered equivalent if they contain exactly the same individuals. The following example states that the class *Foil* is equivalent to the class *Negative User Characteristic*.

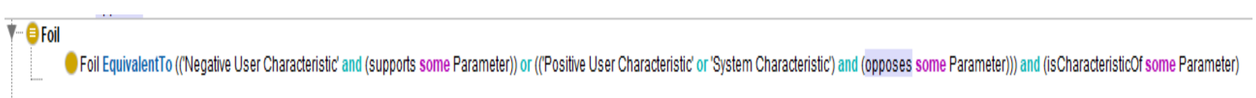


Figure 22. Equivalent to - Class Foil with Negative User Characteristic

Inverse of
Sometimes one property can be obtained by taking another property and changing its direction, i.e. inverting it. For example, the property hasParent can be defined as the inverse property of hasChild. This would for example allow to deduce for arbitrary individuals A and B, where A is linked to B by the *hasChild* property, that B and A are also interlinked by the *hasParent* property [4]:
In this sample property *dislikes* is InverseOf *dislikedBy*.

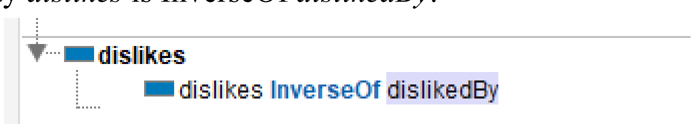


Figure 23. InverseOf

Property chain

Sometimes, we do wish to have some transfer along properties of different kinds. This effect is achieved by using **Role Chains** [4]. A role chain is best explained using an example. By creating a role chain of two properties we can make an inference.

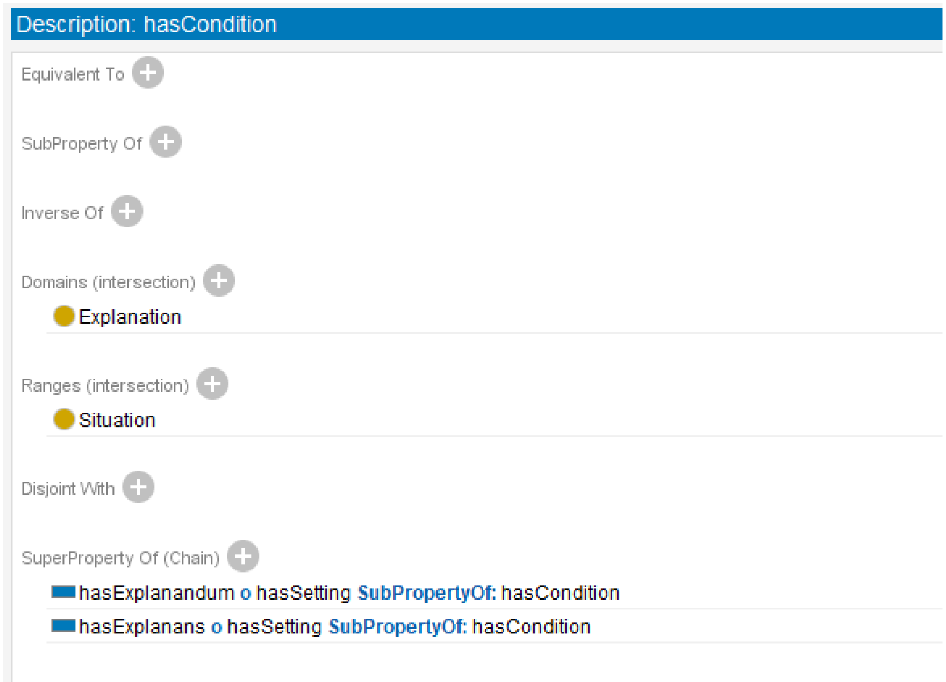


Figure 24. Property Chain

Data ranges

Data ranges define a range to which a value can belong.
In our example *High Budget Food* can be between *3.9595* and *1000.*

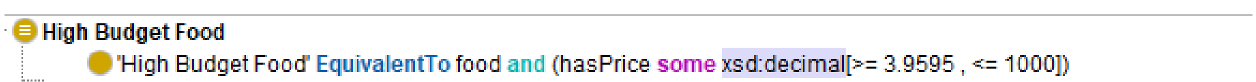


Figure 25. Data ranges for High Budget Food

Disjoint with

It turns out that it also make sense to transfer the notion of class disjointness to properties: two properties are disjoint if there are no two individuals that are interlinked by both properties [4]. Following common law, we can thus state that parent-child marriages cannot occur, as in our example we show that allows and forbids as in real life they are disjoint properties.

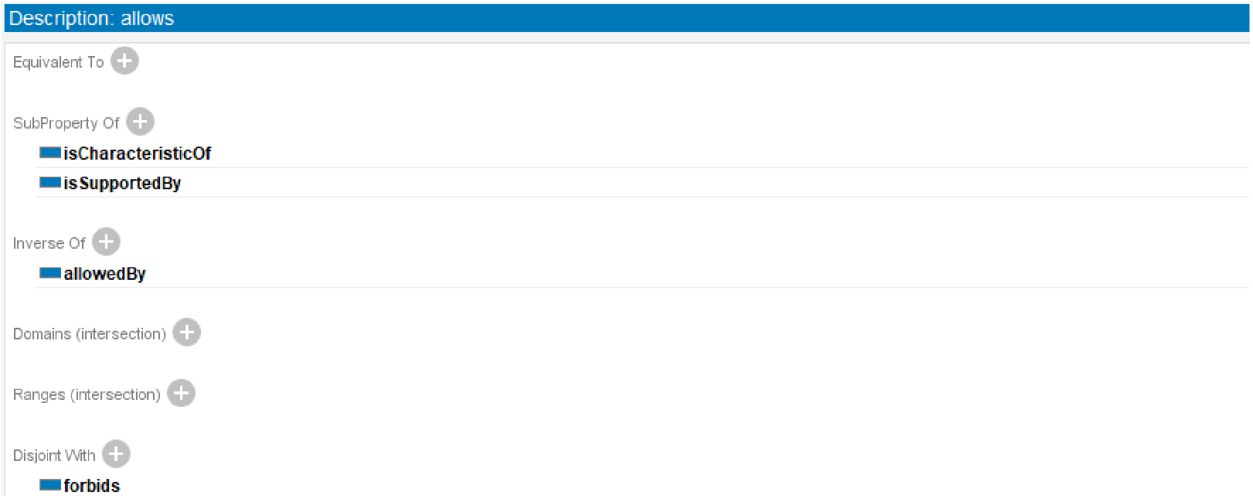
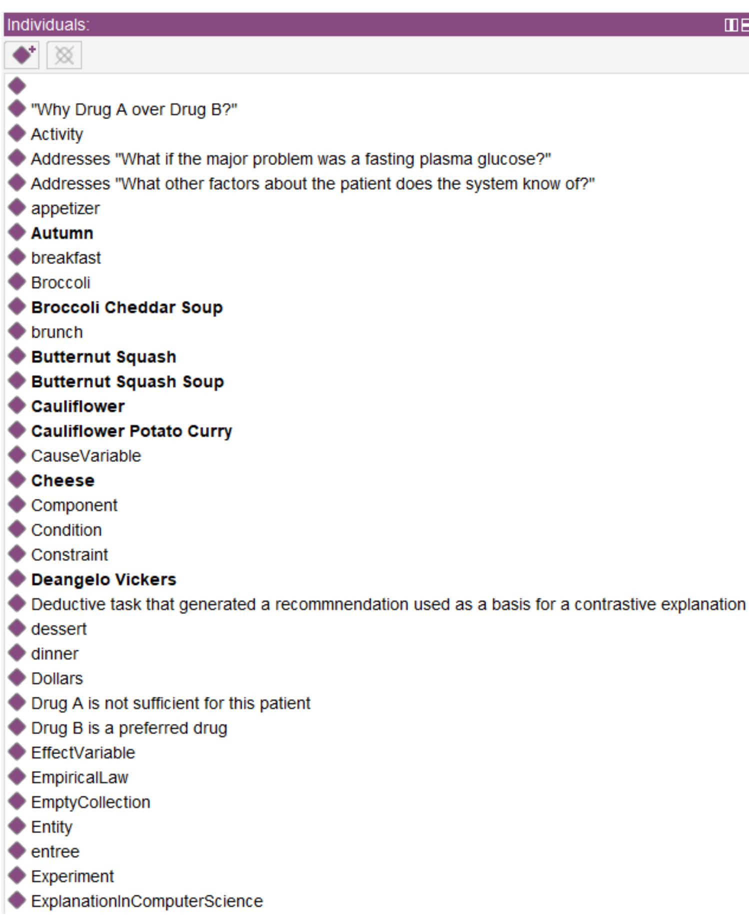


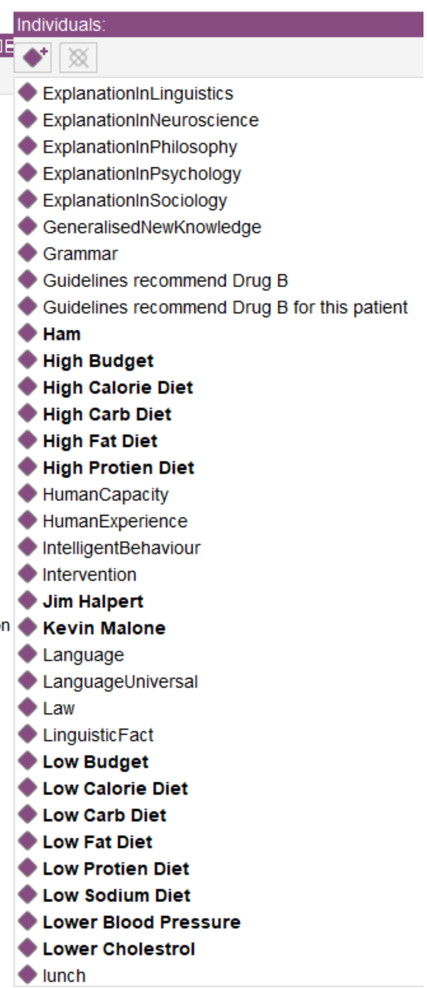
Figure 26. Disjoint of allows with - forbids

7. Second part

Population of the domain ontology with data – at least 30 instances/individuals

The ontology is populated with 117 instances/individuals.





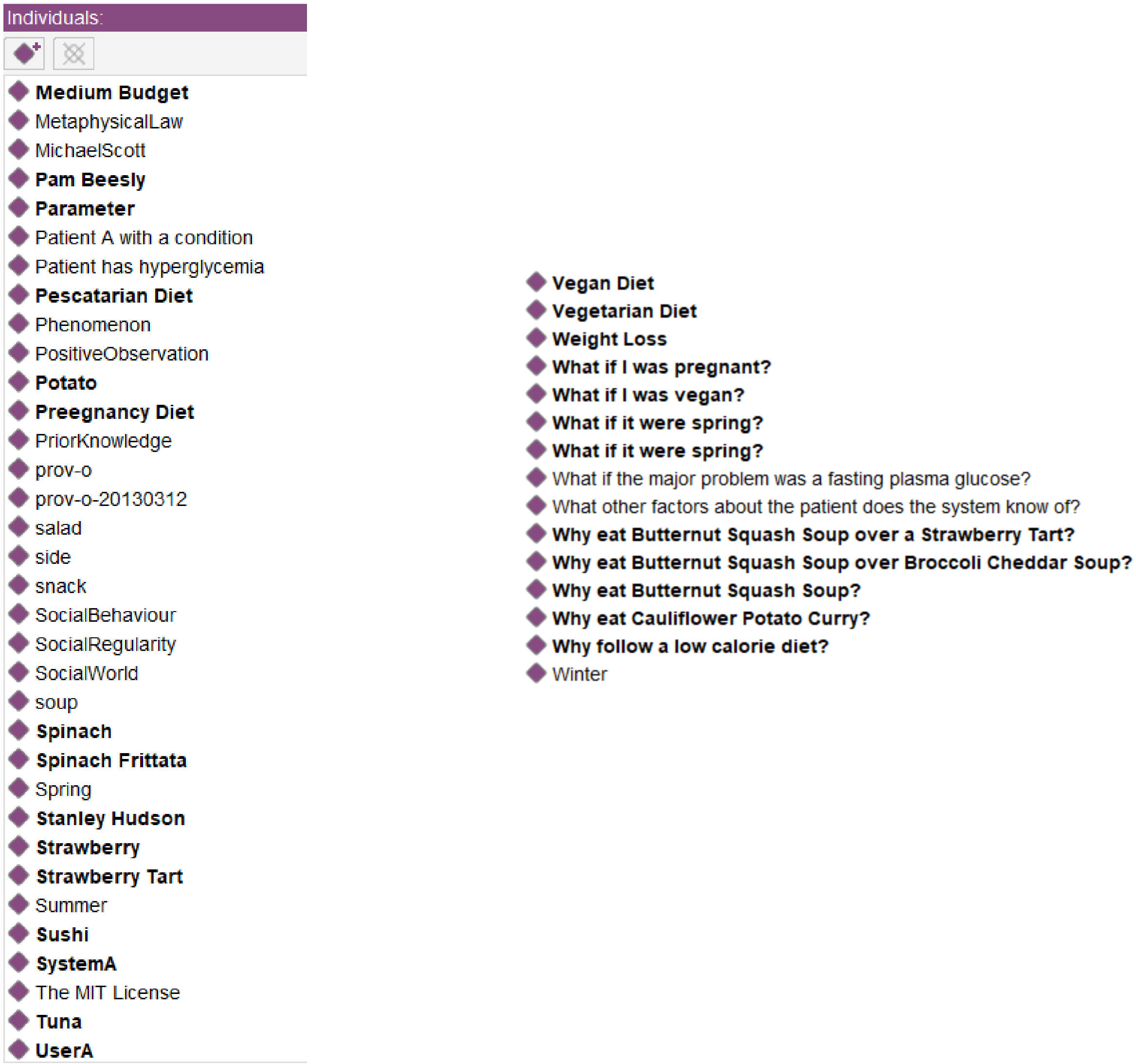


Figure 27. Domain Ontology with data - 117 instances

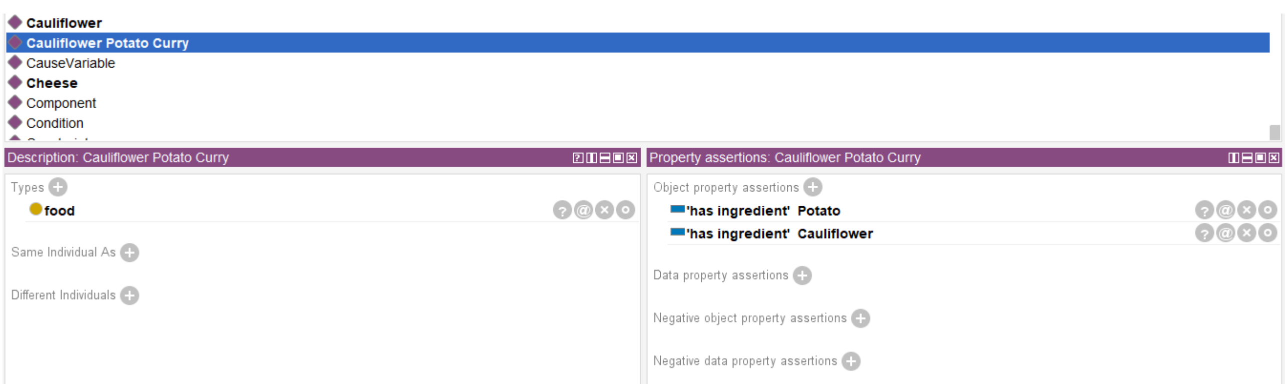


Figure 28. Description of the individuals (in this case for the individual Cauliflower Potato Curry)

8. Third part

Write 5 SPARQL queries which argue the usability of your system

8.1. Show which food instances to which food class/subclass they belong:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?characteristic ?classes
WHERE{

  ?characteristic a ?classes .
  ?classes rdfs:subClassOf ?food .

}
```

characteristic	classes
Spinach Frittata	food
Sushi	food
Cauliflower Potato Curry	food
Butternut Squash Soup	food
Strawberry Tart	food
Broccoli Cheddar Soup	food
Spinach	High Folate Food
Tuna	aquatic animal product
High Budget	Budget
Medium Budget	Budget
Low Budget	Budget
SystemA	System
Ham	primary food commodity of animal origin
Cheese	manufactured milk product (single ingredient) of animal origin
Strawberry	primary food commodity of plant origin
Spinach	primary food commodity of plant origin
Cauliflower	primary food commodity of plant origin

Execute

Figure 29. Result of the first SPARQL query

8.2. Find the highest price of food:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?food ?price
WHERE { ?food feo:hasPrice ?price }
ORDER BY DESC(?price) LIMIT 1
```

food	price
Butternut Squash Soup	"12"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 30. Result of the second SPARQL query

8.3. Show the list of foods with their main ingredient and in which season they are available in ascending order:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX food: <http://purl.org/heals/food/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?food ?ingredient ?seasonAvailable
WHERE {

?food food:hasIngredient ?ingredient .
OPTIONAL { ?ingredient feo:isAvailibleIn ?seasonAvailable }

}

ORDER BY ASC (?food)
```

food	ingredient	seasonAvailable
Broccoli Cheddar Soup	Broccoli	
Butternut Squash Soup	Butternut Squash	Autumn
Cauliflower Potato Curry	Cauliflower	Autumn
Cauliflower Potato Curry	Potato	
Spinach Frittata	Spinach	
Strawberry Tart	Strawberry	Spring
Sushi	Tuna	

Figure 31. Result of the third SPARQL query

8.4. For each user show the number of goals accomplished and the number of goals left:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT Distinct ?user (COUNT(?accomplished) as ?completedGoals)  (COUNT(?goal) as ?leftGoals)

WHERE{
{
 ?user feo:hasGoal ?goal
}
UNION{
 ?user feo:accomplishedGoal ?accomplished
}
}

GROUP BY ?accomplished ?user
ORDER BY ASC (?count)
```

user	completedGoals	leftGoals
Kevin Malone	"0"^^<http://www.w3.org/2001/XMLSchema#integer>	"2"^^<http://www.w3.org/2001/XMLSchema#integer>
Pam Beesly	"1"^^<http://www.w3.org/2001/XMLSchema#integer>	"0"^^<http://www.w3.org/2001/XMLSchema#integer>
Deangelo Vickers	"1"^^<http://www.w3.org/2001/XMLSchema#integer>	"0"^^<http://www.w3.org/2001/XMLSchema#integer>
Stanley Hudson	"1"^^<http://www.w3.org/2001/XMLSchema#integer>	"0"^^<http://www.w3.org/2001/XMLSchema#integer>
Jim Halpert	"1"^^<http://www.w3.org/2001/XMLSchema#integer>	"0"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 32. Result of the fourth SPARQL query

8.5. Select all users that are allergic to a food and the food they are allergic to:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX food: <http://purl.org/heals/food/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?user ?allergicIngredient
WHERE{
{
 ?user food:isAllergicTo ?allergicIngredient
}
UNION
{
 ?allergicIngredient food:isAllergenFor ?user
}
}
```

user	allergicFood
UserA	Broccoli

Figure 33. Result of the fifth SPARQL query

8.6. Query that tells why we should follow a low calorie diet:

SPARQL query:

```
PREFIX feo: <http://www.semanticweb.org/admin/ontologies/2022/4/untitled-ontology-63/>
PREFIX eo: <http://purl.org/heals/eo#>
SELECT DISTINCT ?goal ((Count( distinct ?accomplishedUsers))/(Count( distinct ?allOtherUsers)) as ?percentAccomplished)
WHERE{
 ?WhyFollowLowCalorieDiet feo:hasParameter ?diet.
 ?WhyFollowLowCalorieDiet feo:askedBy ?mainUser.
 ?mainUser feo:hasGoal ?goal.
 ?allOtherUsers feo:hasDiet ?diet .
 FILTER ( ?allOtherUsers != ?mainUser ).
 ?accomplishedUsers feo:hasDiet ?diet .
 ?accomplishedUsers feo:accomplishedGoal ?goal.
}
GROUP BY ?goal
Having ((Count( distinct ?accomplishedUsers))/(Count( distinct ?allOtherUsers)) >= .5)
```

goal	percentAccomplished
Weight Loss	"0.6666666666666666666666667"^^<http://www.w3.org/2001/XMLSchema#decimal>

Figure 34. Result of the sixth SPARQL query

As we see from Fig.34., the result is goal: Weight Loss, so the goal is to lose weight and percentAccomplished that tells that 66.67% of the users that followed the low calorie diet have lost weight which is the goal.

9. Fourth part

Write 5 SWRL rules which extend the usability of your system

9.1. Show the number of diets:

Name	Query
First SWRL Rule	feo:Diet(?d) -> sqwrl:count(?d)

SQWRL Queries | OWL 2 RL | First SWRL Rule

count(?d)
"13"^^xsd:int

Figure 35. Result of the first SWRL rule

9.2. Find the foods that have the price smaller than 30:

Name	Query
Second SWRL rule	food:Food(?f) ^ feo:hasPrice(?f, ?price) ^ swrlb:lessThan(?price, 30) -> sqwrl:select(?f, ?price)

SQWRL Queries | OWL 2 RL | Second SWRL rule

f	
feo:Butternut Squash Soup	12

Figure 36. Result of the second SWRL rule

9.3. The transitivity rule on *hasCharacteristic* property:

```
feo:hasCharacteristic(?c1, ?c2) ^ feo:hasCharacteristic(?c2, ?c3) -> feo:hasCharacteristic(?c1, ?c3)
```

9.4. Show diets in a descending order:

Name	
Fourth SWRL rule	feo:Diet(?d) -> sqwrl:select(?d) ^ sqwrl:orderByDescending(?d)

SQWRL Queries | OWL 2 RL | Fourth SWRL rule

feo:VegetarianDiet
feo:VeganDiet
feo:PregnancyDiet
feo:PescatarianDiet
feo:LowSodiumDiet
feo:LowProteinDiet
feo:LowFatDiet
feo:LowCarbDiet
feo:LowCalorieDiet
feo:HighProteinDiet
feo:HighFatDiet
feo:HighCarbDiet
feo:HighCalorieDiet

Figure 37. Result of the fourth SWRL rule

9.5. List users that have a diet and still have a/some goal/goals to accomplish:

Name	Query
Fifth SWRL rule	owl:Thing(?name) ^ feo:Diet(?diet) ^ feo:hasDiet(?name, ?diet) ^ feo:hasGoal(?name, ?goal) º sqwrl:makeSet(?s, ?name) -> sqwrl:select(?name, ?goal, ?diet)

SQWRL Queries | OWL 2 RL | Fifth SWRL rule

name	goal	
feo:KevinMalone	feo:LowerBloodPressure	feo:LowCalorieDiet
feo:KevinMalone	feo:WeightLoss	feo:LowCalorieDiet

Figure 38. Result of the fifth SWRL rule

9.6. Show the high folate foods that are recommended to consume in a diet:

Name	Query
Sixth SWRL rule	feo:HighFolateFood(?recommendedFood) ^ feo:Diet(?diet) ^ feo:recommendedBy(?recommendedFood, ?diet) º sqwrl:makeBag(?hff, ?recommendedFood) -> sqwrl:select(?recommendedFood, ?diet)

SQWRL Queries | OWL 2 RL | S1

recommendedFood	diet
feo:Spinach	feo:PregnancyDiet

Figure 39. Result of the sixth SWRL rule

10. Tool used

10.1. Protégé

Protégé is a free, open source ontology editor and a knowledge management system [5]. The Protégé meta-tool was first built by Mark Musen in 1987 and has since been developed by a team at Stanford University [5]. The software is the most popular and widely used ontology editor in the world [5].
Protégé provides a graphic user interface to define ontologies [5]. It also includes deductive classifiers to validate that models are consistent and to infer new information based on the analysis of an ontology [5].

11. References

[1] Techopedia. (2017, March 31). *What is Semantic Web? - Definition from Techopedia*. Techopedia. Retrieved: June 8, 2022, from https://www.techopedia.com/definition/27961/semantic-web

[2] Ontotext. (n.d.). *What are Ontologies?* Ontotext. Retrieved: June 9, 2022, from https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/

[3] Protégé 5 Documentation. (n.d.). Object Property Characteristics. Retrieved: June 10, 2022, from http://protegeproject.github.io/protege/views/object-property-characteristics/

[4] W3S. (2012, December 11). *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C. Retrieved: June 10, 2022, from https://www.w3.org/TR/owl2-primer/

[5] Wikipedia. (n.d.). *Protégé (software)*. Wikipedia. Retrieved: June 10, 2022, from https://en.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(software)