Domain Formal Modelling

1 Scenario description

The motivation for creating the model in this project was to help people with different levels of background knowledge in the medical world to find the answers they would need. The initial idea was to let people look up their symptoms and get an overview of the possible diseases they could have. This developed in into much more when different personas were taken into account, as will be described in the following section. At the same time we decided to focus of infectious diseases to keep the model at a realistic size for the project. There are three main personas with different usage that the model in this project will be based on - a doctor, a pharmacist and a normal person with no professional background in the medical world. They each have different needs and backgrounds for using the model.

The doctor

Stefano is a 45 year old doctor with his own clinic in the center of Trento. His days are usually busy - especially in the morning where many people come unannounced for his drop-in service between 8 and 10 o'clock. To make sure that the people in line can get the medical advice they need, he cannot use too much time on each patient. This is why he would like an easy way to look up information that he is not able to remember by heart - what kind of sample to take to verify what infection the patient have and the names of drugs that can help the patient to recover or limit the symptoms. Sometimes Stefano also has patient who are going to another country for short or longer stays and need to recommend them vaccines that can protect them in the new environment.

The pharmacist

Anna just got her first job as a pharmacist after finishing her studies. She is 25 years old and is happy to finally go out and use all the things she learned. Even though she has a big amount of knowledge in the area, she would like to double check some information to make sure the customer gets the best possible service. She wants to check what kind of medicaments she can sell the customer for a specific symptom or disease and the price and name of the different brands that sell this medicament. By looking up this information she will be able to give better advice for the customers and keep track of the many different kinds of medicaments and find the one best suited for the customer.

The non-professional

Adrian is 55 year old businessman that lives in Trento with his wife Susanna and his daughter Lisa. Adrian has his own business and therefore he cannot afford to stay at home unless he is very sick or at risk of transmitting the disease to all his employees. This is why he would like to look up possible diseases when he is not feeling well to decide if he should stay at home, go to the doctor or if it is safe for him to go to work. If he decides that it is okay to go to work he might want to stop by a pharmacy to buy something to mitigate his symptoms.

2 Model formalization description

The different personas and their needs was used as a starting point for creating our model. To make the needs of the different personas more concrete, a number of queries they would be interested in asking was written down. These queries provided the foundation for the creation of an ER-model which was formalized into an ontology.

Classes

In the ER-model, some decisions were made about what concepts to make into entities and which were only attributes. Some of the entities that were in the original ER-model and became classes in the ontology was e.g. Infection, Symptom, Treatment and Pharmacy. But the conversion into a formal model also gave some difficulties, since classes such as Symptom and SideEffect could not be sister terms due to overlapping instances. The names of the classes also needed to be chosen with care to avoid subclasses that overlapped with the parent class but was not a subset.

According to our scenario, we are designing a online infectious diseases platform, with main classes such as Infection, Cause, Symptom and Treatment. For the doctor, who need to search the samples of specific infectious diseases, we added Diagnosis as a class. For staff working in pharmacy, who need to look up information about drugs, we added DrugTreatment as the subclass of Treatment. For the normal people who have no knowledge about the diseases, we added City, Prevention, Pharmacy, RiskFactors and RiskGroups.

For some classes we added subclasses. For example the class VirusInfection would contain all the individuals such as InfluenzaA, NoroVirus belonging to virus infection in our domain. Classes may be organised into a superclass-subclass hierarchy, which is also known as a taxonomy. Subclasses are subsumed by their superclasses. For example consider the classes Infection and VirusInfection: VirusInfection is a subclass of Infection (and Infection the superclass of VirusInfection). This says that, all members of the class VirusInfection are members of the class Infection , Being a VirusInfection implies that you're an Infection. Figure 1 shows the relationship between Infection and VirusInfection.

Figure 1: Subclasses of 'Infection' in our model.

Properties

We used two types of properties in the model: object properties and data properties. At first, we were confused about the differences between subclasses and properties. Figure 2 shows how the Address, Rating, Sections and Size was added as a subclass of Pharmacy in the very first draft of our model. This was changed, since they are not subclasses, but it was an important lesson for us. Subclasses can be seen as subset of the superclass, which is why e.g. 'Address' does not belong - it does not only belong to the class 'Pharmacy' and is therefore not a subset. Instead they were all properties that represented relationships in the model.

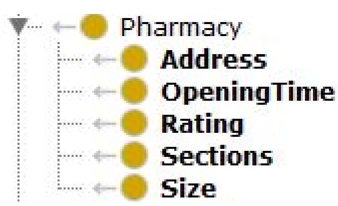

Figure 2: First version of our ontology

Object properties

Object properties join two different individuals together. Figure 3 shows some of the relationships among classes in our model.

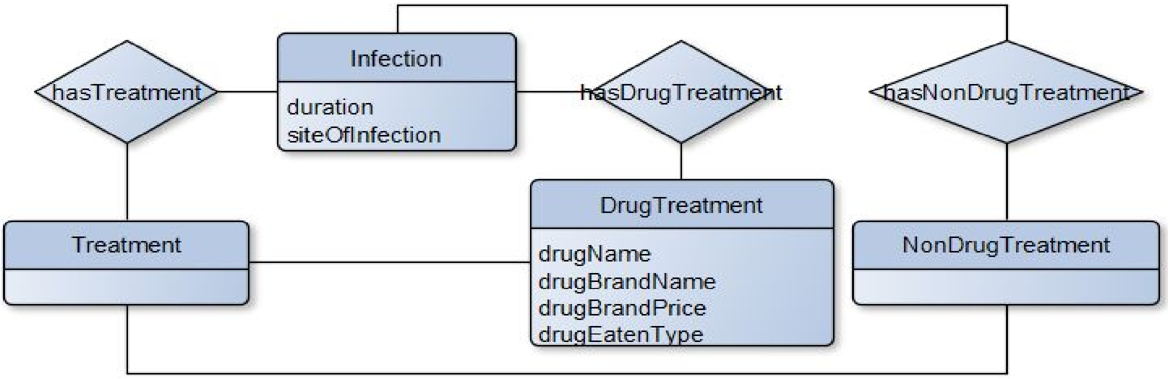

Figure 3: This figure shows the process of creating our object properties.

Let's look at the relationship between the class Infection and the class Treatment. To treat an infection there are different procedures - for some infections you might need to take a specific drug, whereas other infections can be treated by just keeping hydrated with water and electrolytes. Both drinking water and the specific drug treatment are individuals of the class Treatment with the relation 'hasTreatment' between Infection and Treatment, but to distinguish between drugs and changes in behaviour we decided to create the subclasses DrugTreatment and NonDrugTreatment for the superclass Treatment. To connect these subclasses to the Infection class we needed to introduce the object properties hasDrugTreatment and hasNonDrugTreatment as seen in figure 4.

Figure 4: Subclasses of the object property 'hasTreatment'

Data properties

The data properties link individuals to data values. We can use the class Pharmacy as an example to explain how we build our data properties. As the scenario mentioned before, if a normal people Adrian who lives in Trento wants to find a pharmacy to buy some drugs, he need to know the opening time, detailed address and the name of pharmacies. These attributes are obviously used to describe values, so we created the data properties addressDetails, endTime, PharmacyName, and openTime to describe a specific pharmacy. How this was implemented in Protégé can be seen in figure 5.

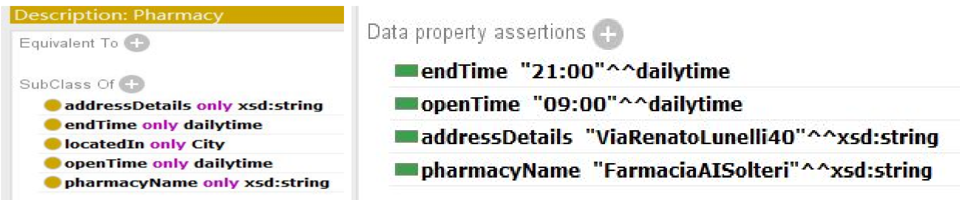

Figure 5: Data properties for the class Pharmacy

Formal model

The total number of classes, properties and individuals added in our model are shown in figure 6 and the full list of the classes, properties and data types are shown in table 1 and 2.

Class count	20
Object property count	27
Data property count	12
Individual count	73

Figure 6: The count of the classes, properties and individuals in our final model.

Classes	Object properties
Cause City Diagnosis Infection BacterialInfection VirusInfection Pharmacy Prevention LifeStylePrevention VaccinePrevention RiskFactors RiskGroups Symptom SideEffect Transmission Treatment DrugTreatment AntibioticsDrugTreatment AntiviralsDrugTreatment NonDrugTreatment	farmacyHasDrug hasDiagnosis hasDiseaseCause hasPrevention hasLifeStylePrevention hasVaccinePrevention hasRiskFactor hasRiskGroup hasSideEffect hasSymptom hasTransmission hasTreatment hasDrugTreatment hasNonDrugTreatment isDiagnosisOf isDiseaseCauseOf isPreventionOf isRiskFactorOf isRiskGroupOf isSideEffectOf isSymptomOf isTransmissionOf isTreatmentFor isDrugTreatmentFor isNonDrugTreatmentFor locatedIn

Table 1: The full list of classes and object properties used in our ontology.

Data properties	Data types
owl:topDataProperty pharmacyName drugEatenType addressDetails siteOfInfection buyWithPrescription drugBrandName drugBrandPrice drugName duration endTime openTime	dailytime rdf:PlainLiteral xsd:boolean xsd:integer xsd:string

Table 2: The full list of data properties and data types used in our ontology.

3 Lexical information upload description

To make sure that the created model can be interpreted by other people who might be interested, it is important to remove ambiguity of any words used. To do this WordNet was used - here the different meanings of a specific word are described and they each have their own ID to distinguish them. Some of the classes in the ontology were easy to find, some were a bit more complicated and some were missing since they were specific for the medical domain. In the following sections the considerations will be described for each class in the ontology.

Infection
Looking up 'infection' on WordNet resulted in 7 different uses and the right one for our model was 'the pathological state resulting from the invasion of the body by pathogenic microorganisms'.

> **BacterialInfection**
> The concept 'bacterial infection' was not found on WordNet even though both 'infection' and 'bacteria' was there. These concepts were however not specific enough for this model since the meaning was not only a bacteria but 'infection by a bacteria that is pathogenic to humans' - therefore a new term was created.

> **VirusInfection**
> On WordNet the term 'virus infection' was there, and the description 'infection by a virus that is pathogenic to humans' fitted well with the use in our model.

Symptom
For the word 'symptom' it was important to use the medical definition 'any sensation or change in bodily function that is experienced by a patient and is associated with a particular disease'.

> **SideEffect**
> The term 'side effect' was in the beginning a sister term of symptoms, but since these two groups overlap (a side effect of a drug can also be a symptom of a disease) one needed to belong to the other. Looking at WordNet for the inherited hypernym of 'side effect' showed that it belonged to the parent term 'symptom' which we then used in our model.

Transmission
The word 'transmission' has several meanings - in this context the definition was 'an incident in which an infectious disease is transmitted'. The other meanings were terms outside the medical area.

Cause
Looking up the word 'cause' gives many hits on WordNet and in this model, what we were looking for was a term describing why a disease occurred. What matched best was the definition 'any entity that produces an effect or is responsible for events or results'.

Treatment
For the word 'treatment' the medical definition 'care provided to improve a situation (especially medical procedures or applications that are intended to relieve illness or injury)' was used since this was the most fitting description.

DrugTreatment
When searching on WordNet and looking at the full hyponym of ‘treatment’ (same as above) there was no term close to the meaning we needed. Instead a new term with the definition ‘treatment of illness with pharmaceutical drugs’ was used.

AntibioticDrugTreatment
For the ‘antibiotic drug treatment’ the synset ID of the term ‘antibiotic drug’ was used with the definition ‘a chemical substance derivable from a mold or bacterium that can kill microorganisms and cure bacterial infections’.

AntiviralsDrugTreatment
As with the antibiotics we also found ‘antiviral drug’ in WordNet with the definition ‘any drug that destroys viruses’.

NonDrugTreatment
This class is a class we defined as ‘treatment of illness without pharmaceutical drugs’.

Prevention
When looking up ‘prevention’ on WordNet there was only one hit and this was a possible description of the term used in the model. When looking at the full hyponym however a more fitting word showed up which was ‘prophylaxis’ described as ‘the prevention of disease’ instead of just ‘the act of preventing’. The ID for this word was used as synset ID for the word ‘prevention’ in our model to make it clear that the meaning was prevention in a medical setting.

LifestylePrevention
This class was not found on WordNet, but instead defined as ‘the act of preventing by changing one’s lifestyle’.

VaccinePrevention
For the ‘vaccine prevention’ the definition of a ‘vaccine’ was used: ‘immunogen consisting of a suspension of weakened or dead pathogenic cells injected in order to stimulate the production of antibodies

Diagnosis
The word ‘diagnosis’ was found on WordNet with the description ‘identifying the nature or cause of some phenomenon’.

Pharmacy
For the ‘pharmacy’ a synset ID with the description ‘a retail shop where medicine and other articles are sold’ was used.

City
The ‘city’ used in our model is a place, so the description fitting was ‘a large and densely populated urban area’.

RiskFactors
The term ‘risk factors’ is used in the medical manner where it means anything that can increase the risk of getting in contact with a disease - therefore the we chose the synset ID of the term ‘risk of exposure’ instead of the term ‘risk of infection’ (where exposure already happened).

4 Top-level grounding

To better understand the model we made, a top-level grounding was performed. The model was mapped to a light version of the CSK and the different considerations are described in the sections below.

Infection
The infection was placed as a ‘situation’ in the CSK, since it is can be described as a situation where a pathogenic microorganism is invading the body.

Cause
The cause was mapped to ‘event’ since it happens at a specific time and at a specific location of the body.

Symptom
For the term symptom, a mapping to the category ‘event’ was also most fitting, since a symptom takes place in a specific period of time.

Transmission
Transmission was mapped to ‘serve’ because there are two objects affecting each other - the sick person transmitting the disease to another person.

Treatment
For the treatment a mapping to ‘event’ was done, since a treatment is something that is done only when sick and therefore specific for a give time and setting.

Prevention
Prevention was mapped to ‘destroy’ since the definition ‘the action by which an object disappears because of losing its identity’ was the best match we could find, since prevention of a disease make the disease disappear/not happen.

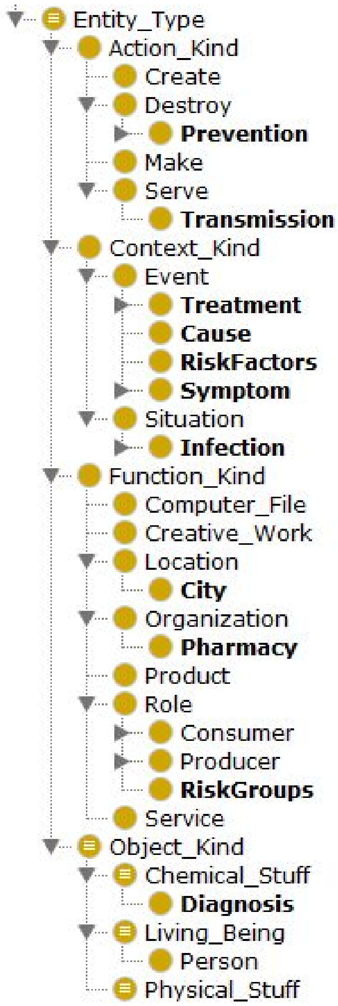

Figure 7: Top-level grounding with CSK

Diagnosis
The diagnosis class in our model has different instances with the type of sample the doctor needs to take from the patient. This is the reason why we mapped the diagnosis to the ‘chemical_stuff’ since the sample is biological (chemical) matter taken from the patient.

Pharmacy
A pharmacy is a company, which is why it was mapped to the ‘organization’ in the CSK.

City
The city belongs to ‘location’ since it is a fixed region of a space (a country).

RiskFactors
The risk factors has instances that are consumption of products or human interaction that can increase the risk of getting infected, which is why the class belong to ‘event’ since the consumption/interaction happens at a given time.

Top-level grounding model

To show the mapping of our model to the CSK in a more visual way, Protégé was used to produce tree structures of the different subclasses in the CSK as seen in figure 8, 9, 10 & 11.

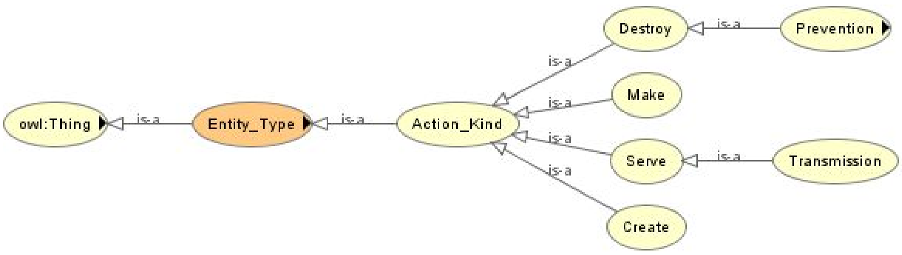

Figure 8: The tree structure of the subclass 'action_kind' with prevention mapped to 'destroy' and transmission mapped to 'serve'.

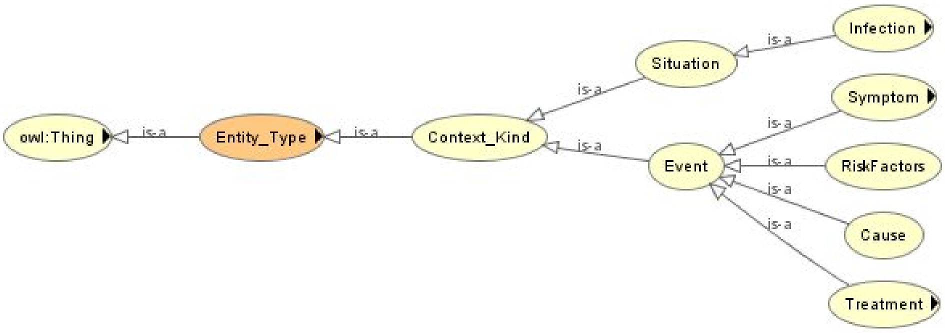

Figure 9: The tree structure of the subclass 'context_kind' with symptoms, risk factors, cause and treatment mapped to the 'event' and infection mapped to 'situation'.

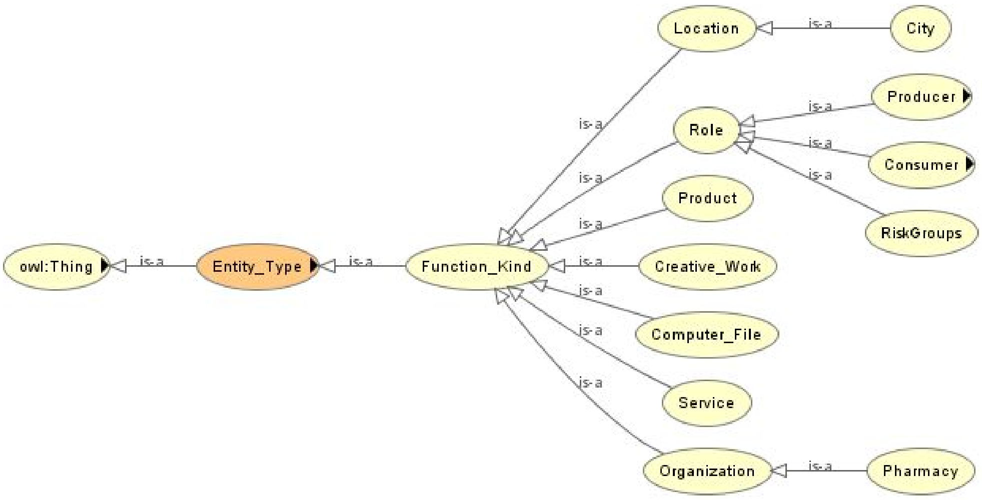

Figure 10: The tree structure of the subclass 'function_kind' with city mapped to 'location', risk groups mapped to 'role' and pharmacy mapped to 'organization'.

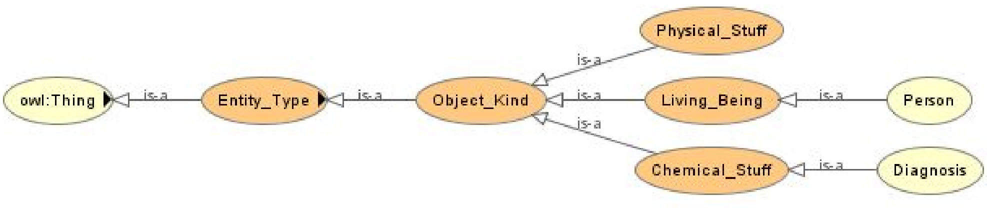

Figure 11: The tree structure of the subclass 'object_kind' with diagnosis mapped to 'chemical_stuff'.

5 Model visualization

To show the ontology created in this project, we used different visualization tools such as OWLViz in Protégé and GraphDB. The different classes and their relationships are shown in figure 12 and an example of one of the relationships between individuals (InfluenzaA and drug02) is shown in figure 13.

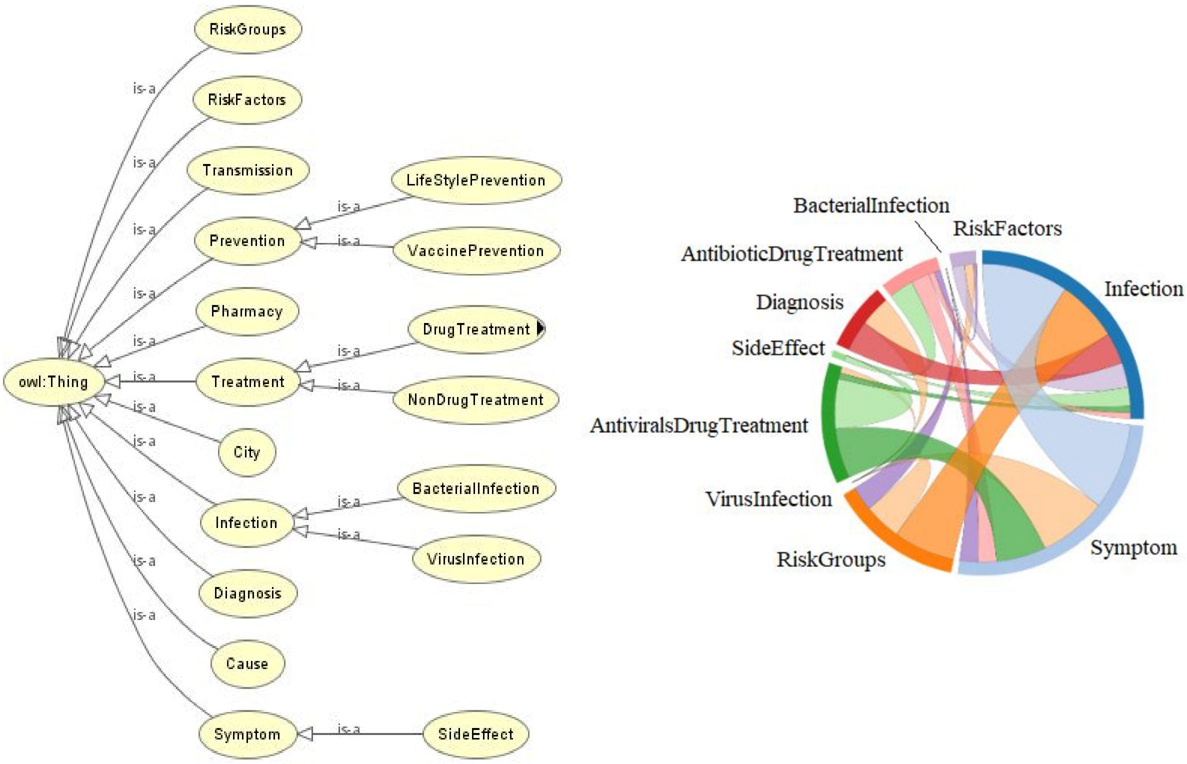

Figure 12: The graph on the left shows all classes in the model with the subclasses. The figure on the right shows the relation among all the classes.

Let's look a little closer on one of the individuals we added - InfluenzaA. In table 3 you can see that InfluenzaA is related to other individuals with the relationships shown in the first column. The individuals also belong to a specific class shown in column 3.

Relationship	**Individuals**	**Class**
hasRiskGroup	OlderPeople,SickPeople,PregnantWomen,Children	RiskGroup
hasTransmission	Airborne	Transmission
hasSymptoms	Headache, SoreThroat, Fever,Chills, RunningNose, Fatigue, MuscleAche	Symptom
hasVaccinePrevention	InfluenzaVaccine	VaccinePrevention
hasRiskFactor	WithInfectedPeople	RiskFactor
hasDiseaseCause	ContactWithInfluenzaVirus	Cause
hasDiagnosis	ThroatSwabs	Diagnosis

Table 3: The relationships between InfluenzaA and other individual belonging to different classes.

Another individual is the one called ‘drug02’. In table 4 you can see the relations drug02 has with other individuals in the model.

Relationship	Individuals	Class
hasSideEffect	Nausea,LossOfAppetite, Drowsiness, Dizziness, Lightheadedness, DryMouth, Constipation, TroubleSleeping	SideEffect
farmacyHasDrug	farmacy01, farmacy03	Pharmacy

Table 4: The relationships between drug02 and other individual belonging to different classes.

To show the relation between the individual ‘InfluenzaA’ and the individual ‘drug02’, we used GraphDB. This relationship is shown in figure 13, where one interesting point is that they are both connected to the individual ‘Headache’ - for the InfluenzaA this individual is a symptom, whereas for the drug02 it is a side effect. The direct connection between the two individuals is through the ‘hasDrugTreatment’ object property.

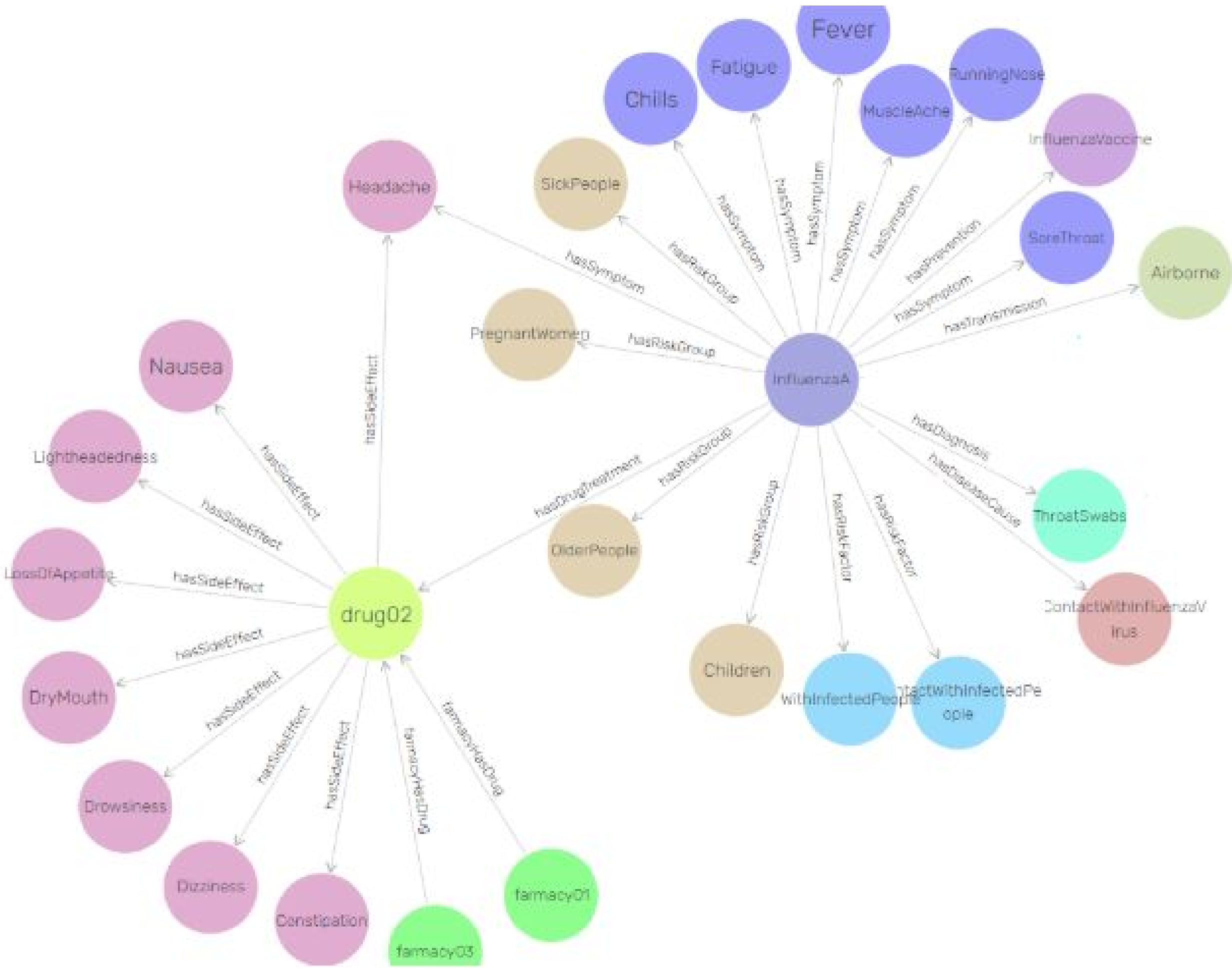

Figure 13: We use graphDB to show the relationship between the individuals InfluenzaA and drug02.

6 Generalized queries

To show how the model works, this section will give example of different queries the personas described in the scenario description could ask. The general queries, the SPARQL code and the output will be provided for each case.

The doctor

The doctor would like to check the different kinds of samples he would need to take based on the type of infection. He also wants to see the different drugs that can be used to treat the infections. This query is seen in table 5.

General Query	give me all diagnosis and drugs treatments of all infections.

SPARQL	SELECT ?Infection ?Diagnosis ?Drug
	WHERE {
	?Infection kx:hasDiagnosis ?Diagnosis.
	?Infection kx:hasDrugTreatment ?Drug.
	}

Table 5: The query performed by the doctor.

The output table from the query performed by the doctor is shown in figure 14. Here we see that for the Escherichia Coli infection he would need to take a urine test and for the Influenza A he would need to get a throat swab instead. The output also shows which drugs to use for the treatment of these infections.

	Infection	Diagnosis	Drug
1	kx:EscherichiaColi	kx:UrineTest	kx:drug01
2	kx:InfluenzaA	kx:ThroatSwabs	kx:drug02

Figure 14: The output of the query performed by the doctor.

The pharmacist

The pharmacist wants to check what kind of medicaments she can sell the customer for diseases and the price and name of the different brands that sell this medicament. The query is shown in table 6.

General Query	Give me all drugs brand and their price which are used to treat infection.
SPARQL	SELECT ?Infection ?Drug ?BrandName ?Price WHERE { ?Infection kx:hasDrugTreatment ?DrugTreatment. ?DrugTreatment kx:drugName ?Drug. ?DrugTreatment kx:drugBrandName ?BrandName. ?DrugTreatment kx:drugBrandPrice ?Price}

Table 6: The query performed by the pharmacist.

The output of the pharmacist's query is shown in figure 15. Because our data is limited, the output only has one brand for each drug, but with more data the price could be compared and e.g. the cheapest one found.

	Infection	Drug	BrandName	Price
1	kx:EscherichiaColi	Vancomycin	Firvanq	"8"^^xsd:integer
2	kx:InfluenzaA	Amantadine	Gocovri	"10"^^xsd:integer

Figure 15: The output of the query performed by the pharmacist.

The non-professional

The non professional person wants to find a pharmacy to buy medicine. He wants to find a pharmacy that closes late, since he wants to buy it after work. The query is shown in table 7.

General Query	Give me the information about the pharmacies, where the pharmacy which is open until latest needs to be ranked first.

SPARQL	SELECT ?PharmacyName ?Address ?EndTime
	WHERE {
	?Pharmacy kx:farmacyName ?PharmacyName.
	?Pharmacy kx:AddressDetails ?Address.
	?Pharmacy kx:endTime ?EndTime.
	}
	ORDER BY DESC(?EndTime)

Table 7: The query performed by the non-professional

The output of the query is seen in figure 16 and shows the address and name of the different pharmacies, and they are ranked by the data property endTime.

	PharmacyName	Address	EndTime
1	FarmaciaComunaleClarina	ViaAlcideDegasperi94	"21:30"^^<http://www.semanticweb.org/kathrine/ontologies/2018/11/untitled-ontology-13#dailytime>
2	FarmaciaAlSolteri	ViaRenatoLunelli40	"21:00"^^<http://www.semanticweb.org/kathrine/ontologies/2018/11/untitled-ontology-13#dailytime>
3	FarmaciaAllaMadonna	ViaGianantonioManci42	"20:30"^^<http://www.semanticweb.org/kathrine/ontologies/2018/11/untitled-ontology-13#dailytime>

Figure 16: The output of the query performed by the non-professional.

7 Final considerations and open issues

Our project shows how it is possible to model real life problem through the methodology learned in this course. We started with an idea of a tool that could fulfill a need we saw ourselves – understanding diseases no matter what you background in the medical world is. We learned a lot by making mistakes and fixing them. The outcome of this project is a model of infectious diseases with some example individuals to prove our concept. To make the model useful for all the scenarios we described it would be necessary to add more data – and even expand the model to work also for other kinds of diseases.