Project report

INFO116 Fall 2019

Group members

118

152

249

Introduction

In this report we will go through each of the assignment's tasks, explain how we solved them and the reasoning behind why we constructed the ontology and the way it turned out. The report will also describe the work progress, how we distributed the different tasks among the group members, our SPARQL queries and the markup of the annotated HTML-pages, as well as a conclusion to the assignment.

Ontology

Questions

With the ontology we wanted to answer some basic questions about the drinks at Vinmonopolet. Some of the questions the ontology can answer is:

- What kinds of wine or beer is good with different kinds of food?
- From which countries are the different kinds of drinks from and which districts are the different wines from?
- What is the price of a drink?
- What kind of characteristics does a drink have?
- What kind of packaging does the different kind of drinks have?
- What kind of raw material is a drink made of?

Classes

When we started making our ontology, we looked through the Vinmonopolet website to get an idea of how we would make our classes for the ontology.
We also got some ideas of what to name our object properties from the wine example ontology.

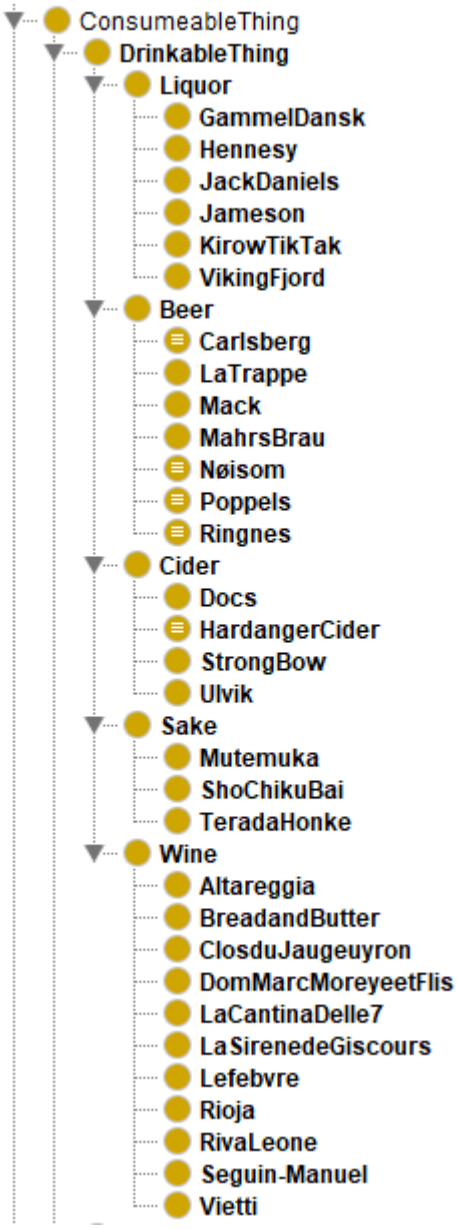

Figure 1:The classes that contains all the drinks

We have a superclass for all the drinks called DrinkableThing. The subclasses of DrinkableThing are all the different types of drinks which are Beer, Cider, Sake, Spirit and Wine. For the subclasses of all the types of drinks we decided to use the brands of the drinks as class names, where all the brands will have individuals of which is drinks of that brand. For example, Carlsberg is a beer brand which have all the different individuals of Carlsberg beers.

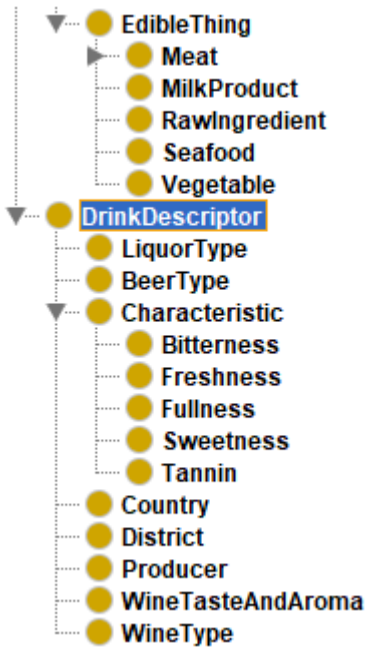

Figure 2:The classes of Descriptors and food

For all the different kinds of foods we made a class called EdibleThing where we have all the different kinds of food that goes well with all the drinks. The subclasses of EdibleThing are all the different types of food that goes well with the different kinds of drinks. Inside the food types we have the individual food items.

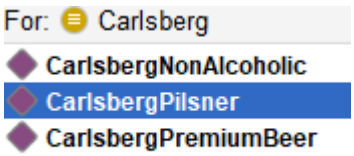

Figure 3: The individuals of Carlsberg

Lastly, we have the Descriptor class. The Descriptor class contains all the different types of individuals that describes a drink. Things like characteristics, countries, beer types and wine types are in the descriptor class.

Object Properties

All object properties that describes a drink is subproperties of the hasDescriptor object property. The reason for this is because they all have a range that is a subclass of the descriptor class.

We also chose to make object properties like hasProducer, fromCountry and fromDistrict. Which describes where the drinks are made, and which producer made them. The rest of the object properties are shown in the image to the right.

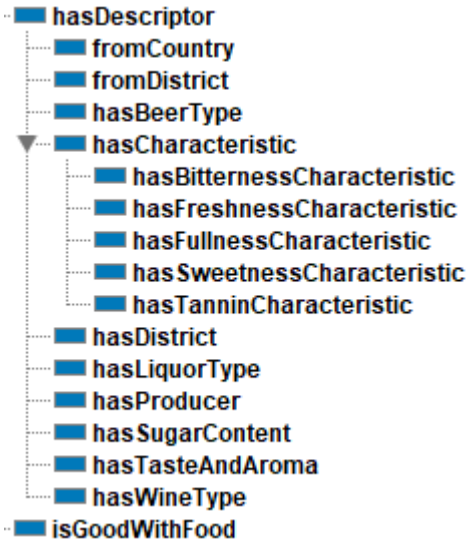

Figure 4: The object properties of the ontology.

There are some drinks that does not have certain object properties. The fromDistrict property have not been used on all the drinks because some of the drinks does not have a district listed on Vinmonopolet and most of the liquors does not have any food listed either.

Data Properties

Our data properties are just a selection of data which describes the product listed on the website. As you can see, we have characteristics like sweetness, fullness and freshness in our data properties. There is a difference between the hasCharacteristic object properties and the data properties however, the hasCharacteristic object properties can be used to find every drink that has a specific characteristic. The data properties however focus on the specific value of the characteristics.

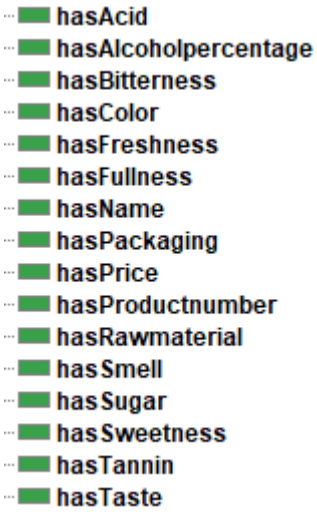

Figure 5: The dataproperties of the ontology.

For the different kinds of data on the website the data properties have three different data types. xsd:string, xsd:decimal and xsd:integer. In the image to the right you can see an example of how we have used the object and data properties to describe one of the wines in the ontology.

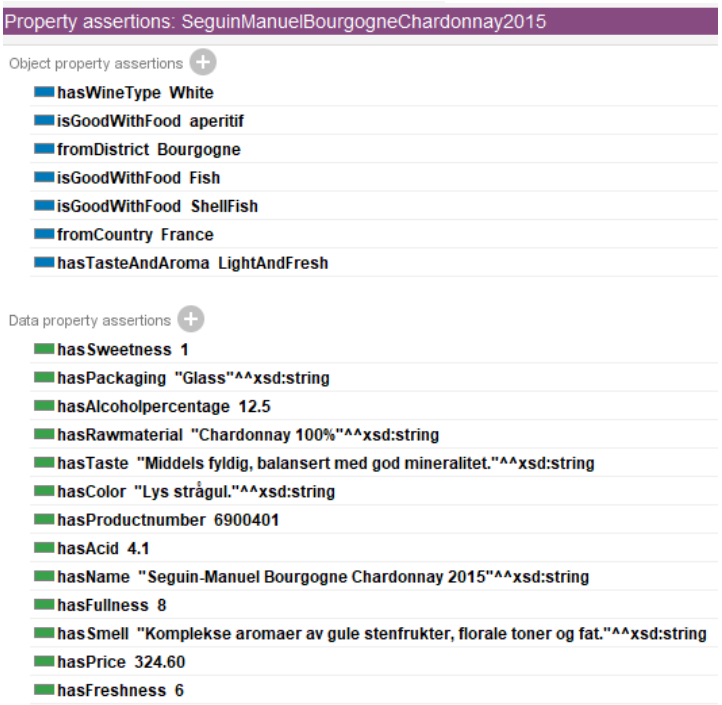

Figure 6: An individual in the ontology, with all the object and data properties that describes it.

Markup

When we were done with the ontology, we started with the markup of the html pages. After playing around with json-LD playground and RDFa/play we started to understand how the json-LD and RDFa-lite worked. For the markup we annotated the country page of Italy and the product page of the prosecco “Altareggia Treviso Prosecco Brut”, both from Vinmonopolet.no. The annotations are in JSON-LD and RDFa Lite.

For the Product page we annotated some of the data about the drink using both schema.org markup and our own ontology markup. We tried to use schema.org to annotate most of the data. We used our own ontology to markup the rest of the data that did not work with schema.org.

While we knew pretty much from the start what to annotate on the product page, the country page was a little bit harder. There were not that much to annotate on the page. Secondly there were not much we could use our ontology for to annotate on the page. We mostly used schema markup for the country page and used our ontology to markup all the districts the country has.

JSON-LD

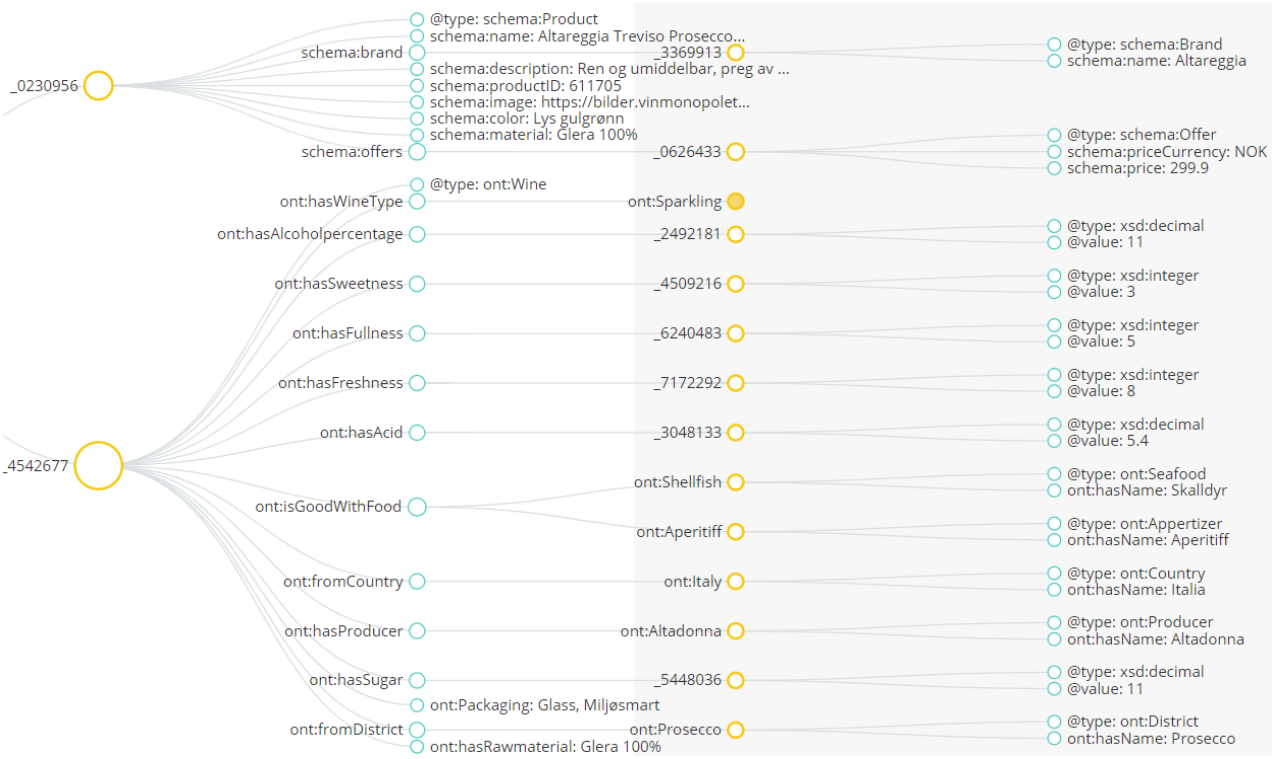

Figure 7: The graph of the json-ld markup shown using json-ld playground

On the image above you can see the visualization of the JSON-LD markup. We used schema.org together with our own ontology when writing the markup. To make sure our code was correct and to visualize it, we used JSON-LD Playground and Google's structured-data testing tool.

Here you can see the vocabulary used in the JSON-LD markup. We used schema.org's Product-schema, with the use of properties like color, brand, productID and more. This is declared as the default vocabulary "@vocab". Our ontology is defined as '"ont". We decided to split our JSON-LD in two, with "@graph", where we used schema for the first part of the code and our ontology for the second part.

```
"@context":
    {
        "schema":"http://schema.org/",
        "ont": "http://www.semanticweb.org/116assignment#",
        "xsd": "http://www.w3.org/2001/XMLSchema#",
        "label": {
                "@id": "http://www.w3.org/2000/01/rdf-schema#label",
                "@container": "@language"
        }
    },
"@graph":[{
"@type":"schema:Product",
"schema:name": "Altareggia Treviso Prosecco Brut",
"schema:brand":{
"@type":"schema:Brand",
"schema:name":"Altareggia"
},
"schema:description": "Ren og umiddelbar, preg av pære og litt urter.",
"schema:productID": 611705,
"schema:image": "https://bilder.vinmonopolet.no/cache/300x300-0/611705-1.jpg",
"schema:color": "Lys gulgrønn",
"schema:material": "Glera 100%",
"schema:offers": {
    "@type": "schema:Offer",
    "schema:priceCurrency": "NOK",
    "schema:price": "299.9"
```

Figure 9:The Markup using schema.org

```
"@type":"ont:Wine",
"ont:hasWineType":{
    "@type":"ont:WineType",
    "@id":"ont:Sparkling",
    "ont:hasName":"Musserende Vin"
} ,
"ont:hasAlcoholpercentage": {
    "@type":"xsd:decimal",
    "@value": 11
},
"ont:hasSweetness": {
    "@type":"xsd:integer",
    "@value": 3
},
"ont:hasFullness":{
    "@type":"xsd:integer",
    "@value": 5
},
"ont:hasFreshness": {
    "@type":"xsd:integer",
    "@value": 8
},
"ont:hasAcid": {
    "@type":"xsd:decimal",
    "@value": 5.4
    },
"ont:isGoodWithFood":[{
        "@type":"ont:Seafood",
        "@id":"ont:Shellfish",
        "ont:hasName":"Skalldyr"
    {
        "@type":"ont:Appertizer",
        "@id":"ont:Aperitiff",
        "ont:hasName":"Aperitiff"
    }],
```

Figure 8:Markup using our own ontology.

In the images above and to the right you can see how we used schema.org to annotate website about "Altareggia Treviso Prosecco Brut", and to the right you can see how we used our or ontology. This shows how we, with both our own ontology and schema.org's, have implemented the semantic data of the webpage.

RDF/a-lite

On the image below you can see a visualization of our RDFa-Lite code, using RDFa-Play.

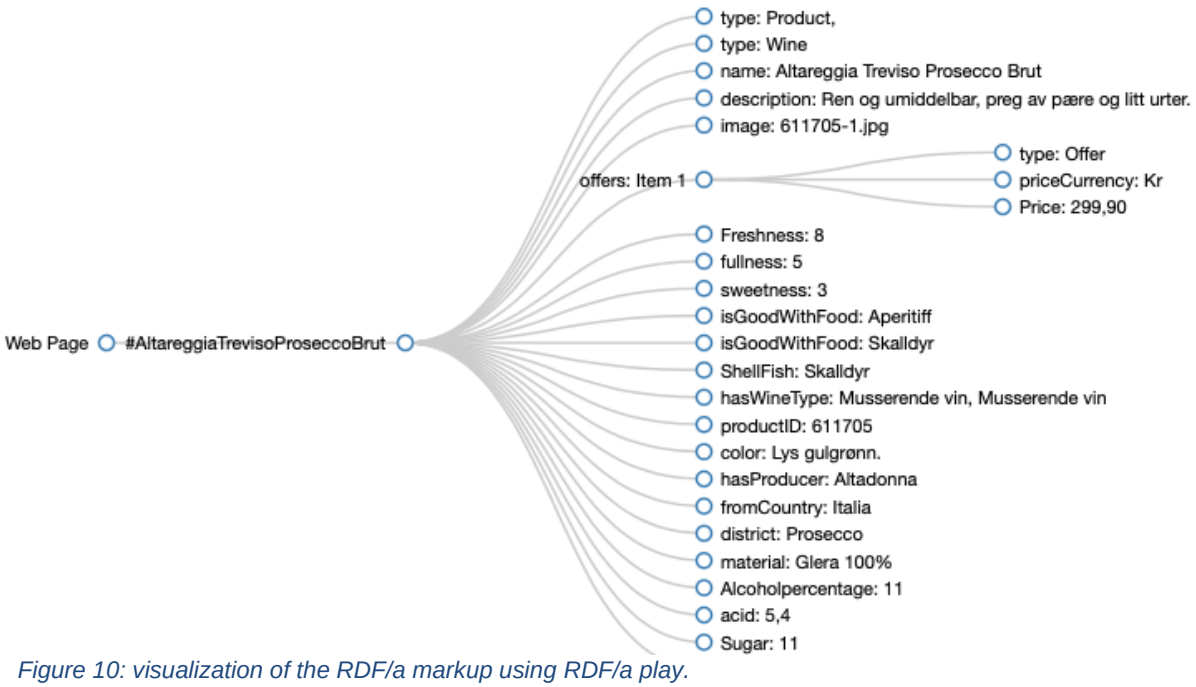

Figure 10: visualization of the RDF/a markup using RDF/a play.

Below you can see the usage of both our own ontology together with schema.org's ontology. The prefix "ont" refers to our ontology, with properties such as "fromCountry", "hasProducer" and "hasWineType". The prefix "schema" refers to schema.org's Product-ontology, with properties like "productID", "description", and "color".

```
<dt>Varetype:</dt>
    <dd><span property="ont:hasWineType">Musserende vin, Musserende vin</span></dd>

<dt>Varenummer:</dt>
    <dd><span property="schema:productID">611705</span></dd>
<dt>Smak:</dt>
    <dd><span property="schema:description">Ren og umiddelbar, preg av pære og litt urter.</span></dd>
<dt>Lukt:</dt>
    <dd>Litt dropsaktig aroma med innslag av pære, sitrus og urter.</dd>
<dt>Farge:</dt>
    <dd><span property = "schema:color">Lys gulgrønn.</span></dd>
<dt>ProdusentProdusent:</dt>
    <dd><span property="ont:hasProducer" property="ont:Altadonna">Altadonna</span></dd>
<dt>Land, distrikt, underdistrikt:</dt>
<dd><span property="ont:fromCountry">Italia</span>,<span property="ont:district">Prosecco</span></dd>
<dt>Råstoff:</dt>
    <dd><span property="schema:material">Glera 100%</span></dd>
</dl>
```

Figure 11: Markup of the webpage using RDF/a with our own ontology and schema.org

The images below show the results from Google's structured data testing tool of the schema markup. We did only screenshot a part of the snippet for our own ontology because the snippet list is long.

@type	Product
url	https://www.vinmonopolet.no/vmp/Producenter/Altadonna/Altareggia-Treviso-Prosecco-Brut/p/611705
name	Altareggia Treviso Prosecco Brut
description	Ren og umiddelbar, preg av pære og litt urter.
productID	611705
image	https://bilder.vinmonopolet.no/cache/300x300-0/611705-1.jpg
color	Lys gulgrønn
material	Glera 100%
brand	
@type	Brand
name	Altareggia
offers	
@type	Offer
priceCurrency	NOK
price	299.9

Figure 12:Rich snipper of the json-ld markup, using schema.org.

ADVA

@type	http://www.semanticweb.org/116assignment#Wine
http://www.semanticweb.org/116assignment#Packaging	Glass, Miljøsmart
http://www.semanticweb.org/116assignment#hasRawmaterial	Glera 100%
http://www.semanticweb.org/116assignment#hasWineType	
@type	http://www.semanticweb.org/116assignment#WineType
@id	http://www.semanticweb.org/116assignment#Sparkling
http://www.semanticweb.org/116assignment#hasName	Musserende Vin
http://www.semanticweb.org/116assignment#hasAlcoholpercentage	
@type	http://www.w3.org/2001/XMLSchema#decimal
@value	11
http://www.semanticweb.org/116assignment#hasSweetness	
@type	http://www.w3.org/2001/XMLSchema#integer
@value	3
http://www.semanticweb.org/116assignment#hasFullness	
@type	http://www.w3.org/2001/XMLSchema#integer
@value	5
http://www.semanticweb.org/116assignment#hasFreshness	
@type	http://www.w3.org/2001/XMLSchema#integer
@value	8
http://www.semanticweb.org/116assignment#hasAcid	
@type	http://www.w3.org/2001/XMLSchema#decimal
@value	5.4
http://www.semanticweb.org/116assignment#isGoodWithFood	
@type	http://www.semanticweb.org/116assignment#Seafood
@id	http://www.semanticweb.org/116assignment#Shellfish
http://www.semanticweb.org/116assignment#hasName	Skalldyr
http://www.semanticweb.org/116assignment#isGoodWithFood	
@type	http://www.semanticweb.org/116assignment#Appertizer
@id	http://www.semanticweb.org/116assignment#Aperitiff

Figure 13:Rich snippet of the json-ld markup, using our own ontology.

Benefits of the annotated web pages

With the semantics that have been added, the web pages are now rich with structured data. This is good for various reasons. Firstly, this means that search engines will have an easier time with finding relevant information on the pages. Secondly, this will make it easier for users to find the information they are looking for because they will be shown a more detailed representation of what the web pages has to offer. Thirdly, all the annotated information about the product will be shown in the search results.

SPARQL

In total we have five SPARQL queries which demonstrates how an application could make good use of our ontology.

Prefixes:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ont: <http://www.semanticweb.org/116assignment#>

Our first query finds all the individual beers that are from the country Denmark that is good with fish and returns the brand of the beers and their respective names.

Query:

```
SELECT ?brand ?instance ?name
        WHERE {
        ?brand rdfs:subClassOf ont:Beer.
        ?instance rdf:type ?brand.
        ?instance ont:fromCountry ont:Denmark.
        ?instance ont:hasName ?name.
        ?instance ont:isGoodWithFood ont:Fish}
```

Result:

brand	instance	name
Carlsberg	CarlsbergPilsner	"Carlsberg Pilsner"^^<http://www.w3.org/2001/XMLSchema#string>
Carlsberg	CarlsbergPremiumBeer	"Carlsberg Premium Beer"^^<http://www.w3.org/2001/XMLSchema#string>
Carlsberg	CarlsbergNonAlcoholic	"Carlsberg Non Alcoholic"^^<http://www.w3.org/2001/XMLSchema#string>

This query pulls all red wines that has a taste and aroma that is fresh and fruity, while only showing the wines that cost less than 300 kr.

```
SELECT ?brand ?wine ?name ?price
WHERE {
?brand rdfs:subClassOf ont:Wine.
        ?wine rdf:type ?brand.
        ?wine ont:hasWineType ont:Red.
        ?wine ont:hasTasteAndAroma ont:FreshAndFruity.
        ?wine ont:hasName ?name.
        ?wine ont:hasPrice ?price
        FILTER(?price <300)
}
```

Result:

brand	wine	name	price
RivaLeone	RivaLeonePiemonteRosso2016	"Riva Leone Piemonte Rosso 2016"^^<http://www.w3.or	"129.90"^^<http://www.w3.org/2001/XMLSchema#decim
LaCantinaDelle7	LaCantinaDelle7DonnePiemonteBarbera	"La Cantina Delle 7 Donne Piemonte Barbera"^^<http://w	"124.90"^^<http://www.w3.org/2001/XMLSchema#decim

This query finds all liquors that have glass packaging and is ordering the result after the price of the products.

```
SELECT ?brand ?spirit ?packaging ?Price
WHERE { ?brand rdfs:subClassOf ont:Liquor.
        ?spirit rdf:type ?brand.
```

```
        ?spirit ont:hasLiquorType ont:Whisky.
        ?spirit ont:hasPackaging ?packaging.
        ?spirit ont:hasPrice ?Price.
        FILTER regex( ?packaging, "Glass", "i")
}
ORDER BY DESC(?Price)
```

Result:

brand	spirit	packaging	Price
Jameson	Jameson_Crested	"Glass"^^<http://www.w3.org/2001/XMLSchema#string>	"439.90"^^<http://www.w3.org/2001/XMLSchema#decim
Jameson	Jameson_Caskmates_IPA_Edition	"Glass"^^<http://www.w3.org/2001/XMLSchema#string>	"419.90"^^<http://www.w3.org/2001/XMLSchema#decim
JackDaniels	JackDanielsTenessee	"Glass"^^<http://www.w3.org/2001/XMLSchema#string>	"399.90"^^<http://www.w3.org/2001/XMLSchema#decim

This query pulls all beer that is a Light Lager, it also shows what brand the beer is, the name of the beer, the price and the alcohol percentage of the beverage.

```
SELECT ?brand ?name ?price ?alchoholic
WHERE {
        ?brand rdfs:subClassOf ont:Beer.
        ?beer rdf:type ?brand.
        ?beer ont:hasName ?name.
        ?beer ont:hasBeerType ont:LightLager.
        ?beer ont:hasPrice ?price.
        ?beer ont:hasAlcoholpercentage ?alchoholic

}
```

Result:

brand	name	price	alchoholic
Carlsberg	"Carlsberg Pilsner"^^<http://www.w3.org/2001/XMLSche	"399.60"^^<http://www.w3.org/2001/XMLSchema#decim	"5"^^<http://www.w3.org/2001/XMLSchema#integer>
Mack	"GullMack"^^<http://www.w3.org/2001/XMLSchema#strin	"43.9"^^<http://www.w3.org/2001/XMLSchema#decimal>	"6.5"^^<http://www.w3.org/2001/XMLSchema#decimal>
Carlsberg	"Carlsberg Premium Beer"^^<http://www.w3.org/2001/XI	"29.90"^^<http://www.w3.org/2001/XMLSchema#decimal	"5"^^<http://www.w3.org/2001/XMLSchema#integer>
Ringnes	"Ringnes Extra Gold"^^<http://www.w3.org/2001/XMLSch	"35.90"^^<http://www.w3.org/2001/XMLSchema#decimal	"6.3"^^<http://www.w3.org/2001/XMLSchema#decimal>

This query finds all the beers that goes well with beef and all the wines that goes well with shellfish that is made with the raw material chardonnay. We can also find the wines that is made with only chardonnay by writing “Chardonnay 100%” in the filter.

```
SELECT ?brand ?instance ?name
WHERE {
 {
        ?brand rdfs:subClassOf ont:Beer.
        ?instance rdf:type ?brand.
        ?instance ont:hasName ?name.
        ?instance ont:isGoodWithFood ont:Fish
}
UNION
{
        ?brand rdfs:subClassOf ont:Wine.
        ?instance rdf:type ?brand.
        ?instance ont:hasName ?name.
        ?instance ont:isGoodWithFood ont:ShellFish.
        ?instance ont:hasRawmaterial ?Rawmaterial.
        FILTER regex(?Rawmaterial, "Chardonnay", "i")
        }
}
```

Result:

brand	instance	name
Poppels	PoppelsAmericanPaleAle	"Poppels American Pale Ale"^^<http://www.w3.org/2001/XMLSchema#strin
Mack	GullMack	"GullMack"^^<http://www.w3.org/2001/XMLSchema#string>
Seguin-Manuel	SeguinManuelBourgogneChardonnay2015	"Seguin-Manuel Bourgogne Chardonnay 2015"^^<http://www.w3.org/2001/
DomMarcMoreyeetFlis	DomMarcMoreyetFilsBourgogne2016	"Dom. Marc Morey et Fils Bourgogne 2016"^^<http://www.w3.org/2001/XML
Lefebvre	LefebvreChampagneCuvéeReserveBrut	"Lefebvre Champagne Cuvée Reserve Brut"^^<http://www.w3.org/2001/XML

Contribution to the assignment and the work progress

To make the work progress easier we used different tools such as Google Docs for the report, Visual Studio Live Share for the markup, Discord and Facebook messenger for communication, Dropbox for file transfer, WebProtegé for the ontology as well as weekly meetings to better discuss our overall progress over the weeks.

The use of tools like Visual Studio Live Share, WebProtegé and Google Docs made us work much more effective without having to physically meet up at a location each time we needed to work together, creating room for flexibility. WebProtegé was really good for collaborating on the ontology, because it let everyone change and update the ontology at any given time, instead of one person having to send the files to all the group members each time a change had been made.

We all contributed in answering the different tasks of the assignment, coming with suggestions to improve the different aspects of the tasks and filling in where we saw needed. We all agreed to use a couple of hours outside the labs every week to work on the project and to discuss what to do next.

Conclusion

All in all, this project has given us a good understanding of how ontologies work and how to build them. With the ontology we have extracted some of the key properties from some of the products from the website. We made some SPARQL queries to which demonstrates how the ontology can be used. Lastly, we annotated two web pages with schema.org and our ontology, with RDFa and JSON-LD.

Used Tools

RDFa/Play

http://rdfa.info/play/

RDFa validator

https://www.w3.org/2012/pyRdfa/Validator.html/

JSON-LD Playground

https://json-ld.org/playground/

Visual Studio Code

https://code.visualstudio.com/

Google Drive

https://drive.google.com/drive/

Google Structured Data Testing Tool

https://search.google.com/structured-data/testing-tool/u/0/?hl=no/

WebProtegé and Protegé

https://webprotege.stanford.edu/