INFO116 - Gruppeoppgave

## “Making sense of whisky”

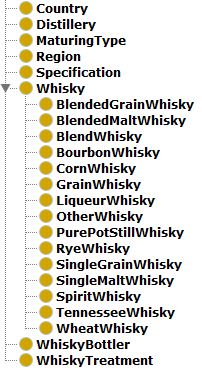
# Introduksjon

For denne oppgaven har gruppen vår arbeidet på en ontologi for konseptet whisky, og da med generelt bruksområde nettstedet whisky.com. Dette er et nettsted som har en stor database med informasjon om whiskyer, der mye av informasjonen er generert av brukerne til nettstedet. Vi vil vise eksempler på hvordan vi har konstruert ontologien vår med bruk av Protegè, forskjellige SPARQL spørringer for uthenting av data, og annotering av nettsider fra whisky.com med RDFa-lite, JSON-LD i kontekst av ontologien vår og schema.org.

Vi vil forklare hva vi har fått til, hvorfor ontologien ble konstruert slik den ble og hvilke spørringer man kan bruke på den. Hvorfor vi valgte å redigere den originale kildekoden til whisky.com og hvilken nytte semantisk annotering gir i denne sammenheng.

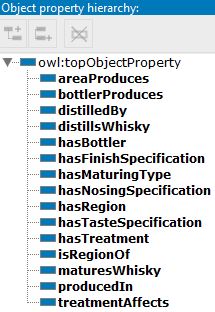
# Ontologi

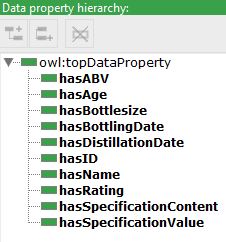
Når vi skulle lage vår ontologi startet vi med å se litt på tidligere oppgaver som vi hadde gjennomført på lab-øving. Vi forsto da at vi ville ha en struktur der vi kunne beskrive individer som en gitt whisky ville ha en nær relasjon til. Dette medførte at vi fikk klasser som Whisky med subklasser som SingleMaltWhisky, BlendWhisky osv. I tillegg kom f.eks; Country, Distillery, Region, WhiskyBottler o.l.

Vi strukturerte ontologien på denne måte fordi vi ønsket å være i stand til å kunne lage spørringer som, “Gi meg en whisky som er produsert i Islay, er av typen SingleMaltWhisky og er eldre enn 5 år”.

Under konstruksjonen av ontologien tilpasset vi den løpende med egenskapene vi ville beskrive whiskyene med. Dette var slik som alkoholprosent, flaskestørrelse, produsert av o.l.

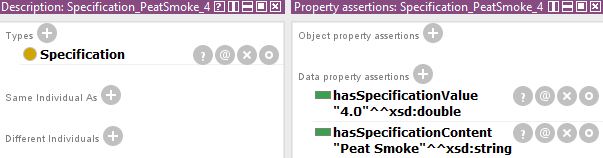
Til venstre kan du se alle klassene/typene som er med i vår ontologi, innenfor disse klassene lager vi enkeltindivider som vi har mulighet til å knytte til andre individer via objekt egenskaper.

Som en ser til venstre på dette bildet kan vi knytte whisky individer til et destilleri ved å bruke *distilledBy*. Vi kan og lage et Region individ knytte dette med *isRegionOf* til et Country individ, for så senere å deklarere *producedIn* en gitt region eller land på en whisky.

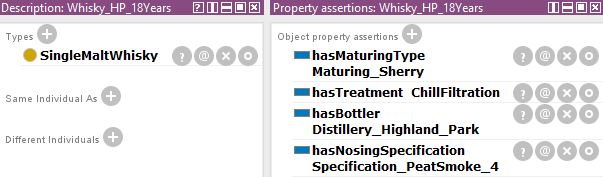
I tillegg la vi opp inverse egenskaper til enkelte der vi følte behovet. Dette vil si at *hasBottler* er inversen av *bottlerProduces*. Den første knytter en whisky til en bottler, og den andre knytter en bottler til en whisky. Dette gjør vi for å ha flere muligheter når man vil lage spørringer, men også fordi det gir et bredere bilde av sammenhengen mellom f.eks: bottler og whisky.

I tillegg hadde vi flere egenskaper som knytter enkelte verdier til individer. Ved hjelp av disse kan vi alkoholprosent, alder, rating, id, navn osv til de gitte individene.

To egenskaper som kanskje skiller seg spesielt ut her er *hasSpecificationContent* og *hasSpecificationValue*. Disse beskriver det vi kaller Specification individ i vår ontologi. Som vi ser under har dette individet to verdier, “4.0” og “Peat Smoke”. Vi brukte dette til å si at en whisky har en egenskap ved seg som er “Peat Smoke” og at dette har en verdi av 4.



Hvilken egenskap dette definerer av en gitt whisky bestemmer en gjennom *hasNosingSpecification*, *hasTasteSpecification* og *hasFinishSpecification*, som knytter en gitt whisky til et Specification individ.



# SPARQL

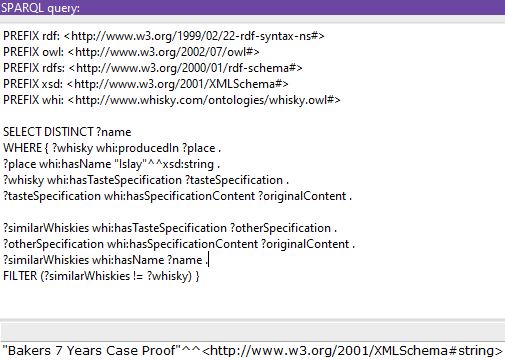
## Setting

Vi har sett for oss at vi har et scenario der vi i prosessen med å utvikle en app for et nettsted som omhandler whisky. Dataene er per dags dato relativt ustrukturerte. Via vår løsning er målet å kunne kjøre nyttige spørringer som kan ta høyde mange ulike aspekt ved de gitte whiskyene på nettstedet. Vi har konstruert over 10 spørringer for å vise fleksibiliteten man har med ontologien og vi har lagt ved scenario eller “use-case” til disse. Alle spørringene ligger ved i en tekst-fil, her viser vi 5 (utdrag fra hver case) av dem:

#### Case 1

Kunden ønsker å ha et enkelt recommender-system for å finne lignende whiskyer som brukeren også kan være interessert i, etter et søk allerede har forekommet.

* Finn whiskyer med lignende smak



Forklaring: Her ser vi et til at kunden har et søk der han ville ha tak i whiskyer som ble produsert i “Islay”. Etter dette spør vi hvilke smaker som tilhører de gitte whiskyene som kunden fann. Så spør vi hvilke av whiskyene i ontologien som har smaker til felles med de som kunden hadde søkt etter, og så viser vi navnet til disse. For resultatet er filter funksjonen brukt til å fjerne whiskyene som kunden allerede fant i første søk.

#### Case 2

I appen skal det også være mulig å kunne finne ut hva en gitt produsent eller destilleri sin gjennomsnittsrating er, på grunnlag av hva de har produsert.

* Gjennomsnittsrating på produsent

***SELECT AVG****(?ratings)*

***WHERE*** *{ ?whisky whi:distilledBy ?distillery .*

*?distillery whi:hasName "Blend-Scotch"^^xsd:string .*

*?whisky whi:hasRating ?ratings }*

→ *FIKS BILDE FRA JENA (SERVER PARSE)*

#### 

#### 

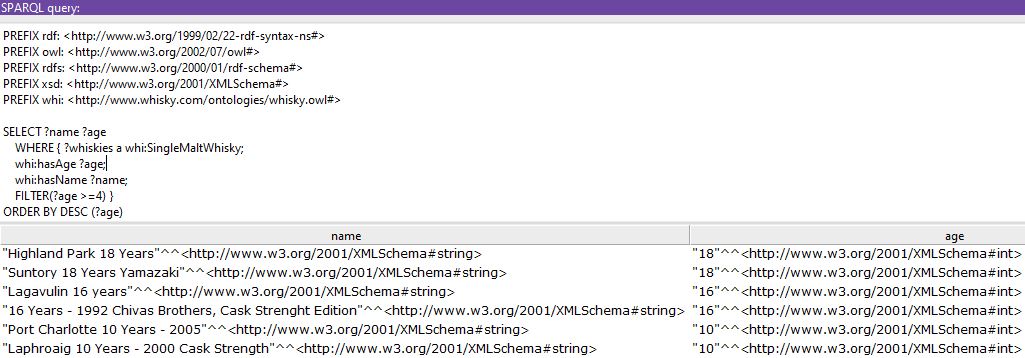
#### 

#### Case 3

Kunden ønsker å kunne finne whiskyer basert spørringer med mange faktorer.

I vår ontologi har vi beskrevet whiskyene med følgende informasjon: *produsent, destilleri, etterbehandling, modning (les: cask), produksjonssted, lukt, smak, ettersmak, rating (vurdering), alkoholprosent, alder, navn, flaskestørrelse, destillerings dato og tappe dato*.

* Her er brukeren ute etter whisky typen SingleMaltWhisky med alder til og med 4 og over. Resultatet er sortert etter alder på whiskeyen i nedgående rekkefølge. Kolonnene navn og alder vises til brukeren.



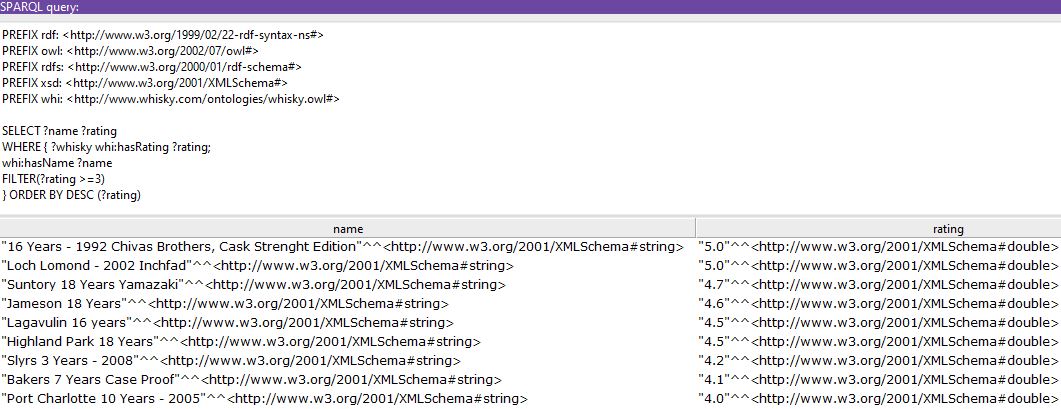
#### 

#### 

#### Case 4

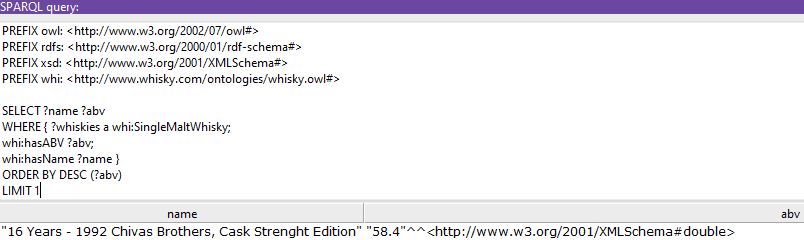
Siden vi har støtte for å beskrive whiskyer med “rating” så har vi lyst å kunne sortere de ulike resultatene (slikt som man finner fra spørringer som tilhører case 3) på denne variabelen.

* Denne queryen henter alle whiskeyene, og gir oss rating som er høyere enn 3, og sorterer resultatet etter høyest verdi først. Vi ser gjerne for oss at dette er en spørring man kan legge til på en mer avansert spørring som en gitt bruker har brukt.



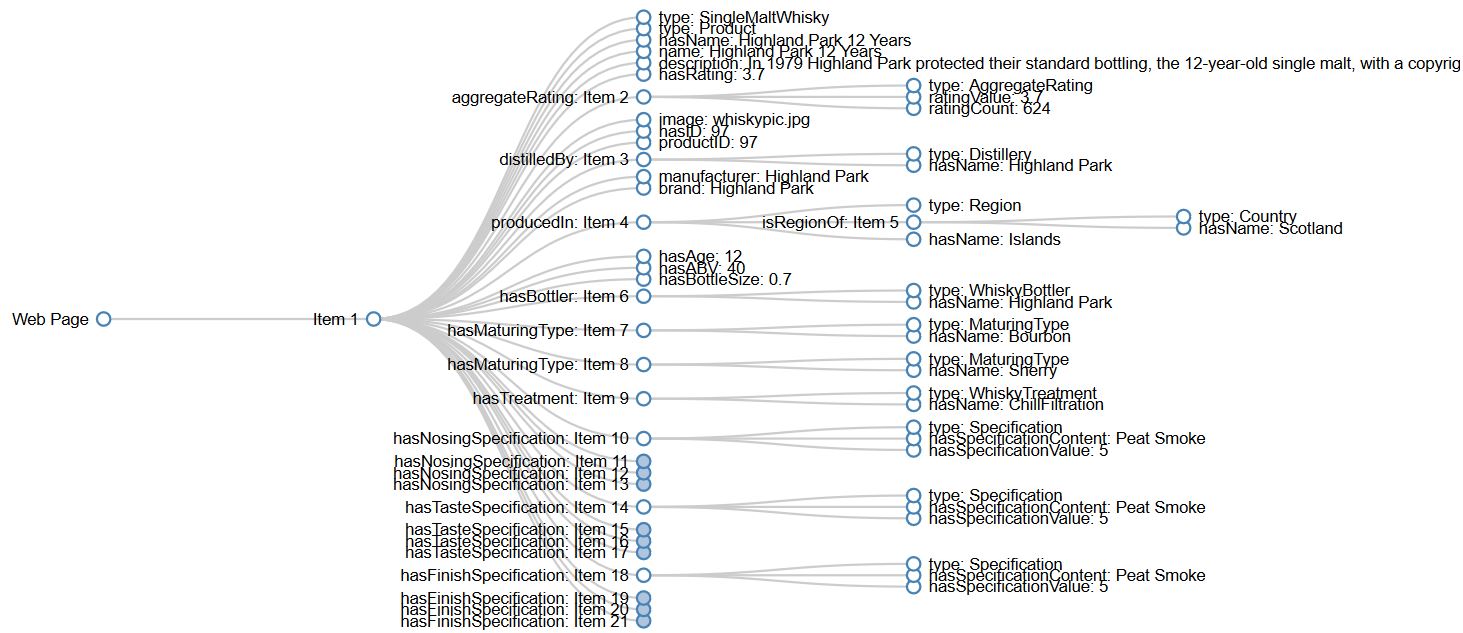
#### Case 5

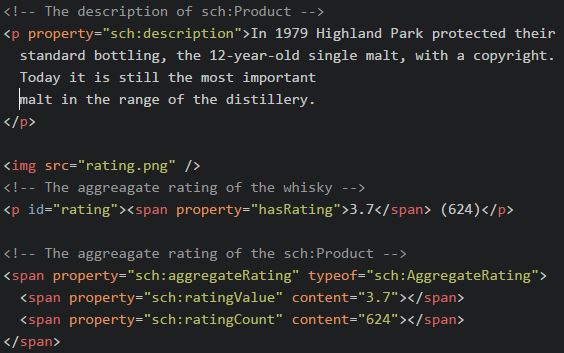
I appen skal der være funksjonalitet som kan tillate en gitt bruker å finne en whisky (basert på faktorer gitt i case 3) men også kunne velge å få vist den som har høyest/minst alkoholprosent og høyest/minst lukt, smak eller ettersmak, eventuelt en kombinasjon av disse.

* Her har brukeren valgt å finne en whisky med høyest alkoholprosent innenfor alle Single Malt Whisky.

# Annotering av nettsider

## RDFa-lite og JSON-LD



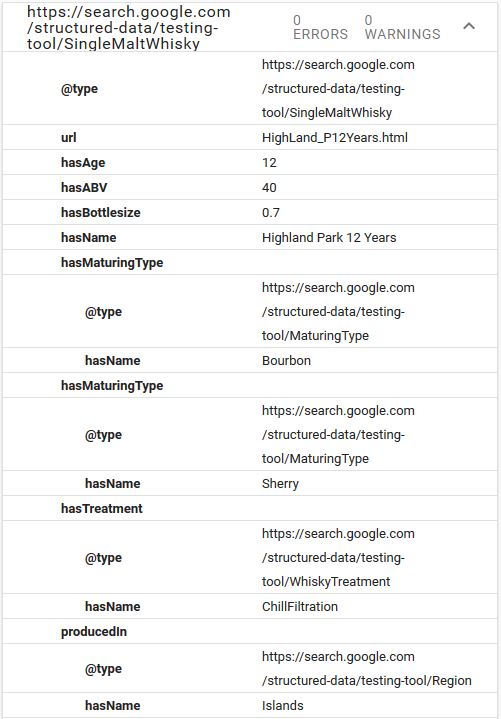
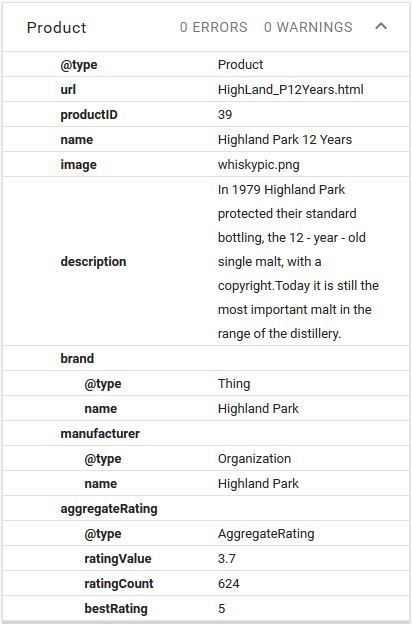
Vi tok et utvalg av 5 whiskyer fra whisky.com og annoterte informasjonen som var tilgjengelig på disse. På bildet over ser vi en graf av RDFa dataene som ligger inne i en av de annoterte sidene (noen av dataene er kollapset for å gi et bedre overblikk). Vi brukte både vår egen ontologi og schema.org til å beskrive informasjonen tilgjengelig. For å validere annoteringsprosessen ble “<http://rdfa.info/play/>” brukt for å lage grafen over, “<https://search.google.com/structured-data/testing-tool>” for å skjekke både JSON-LD og RDFa, samt “<https://json-ld.org/playground/>” for ekstra skjekk av JSON-LD.

Til venstre her ser vi har “sch:” er brukt som en prefix for å henvise til schema beskrivelser, og ellers har vi *hasRating* som er fra vår egen ontologi. Flere steder er lik informasjon beskrevet med både schema og ontologien.

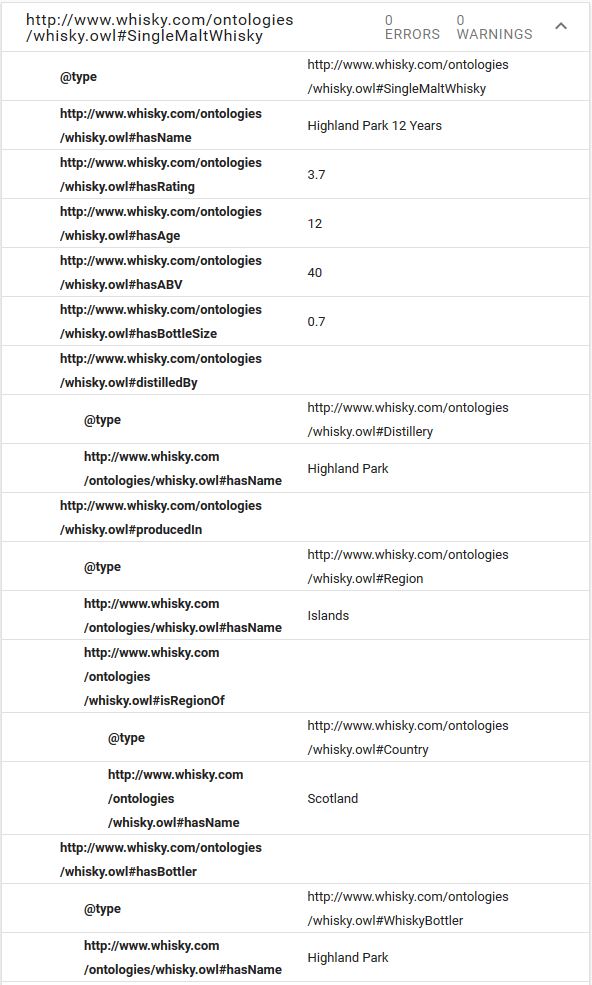


Her ser vi bruk av JSON-LD for å implementere data både fra schema og ontologien vår. Dette er da en av whiskyene vi valgte å annotere som heter “Highland Park 12 Years”.



All den annoterte informasjon over gjør at vi kan teste dette med “rich snippet” verktøyet fra google; for å vise at informasjonen kan bli tolket og evt videre prosessert. JSON-LD under.

*(JSON-LD for schema.org) (JSON-LD for vår ontologi)*

Bildet som viser JSON-LD med vår egen ontologi er kun et utdrag av informasjonen ligger der, dette fordi listen er svært lang med lukt, smak o.l. Bildet som viser JSON-LD for schema.org er så godt som indentisk til hvordan verktøyet tolket RDFa for den annoterte siden, dette er fordi vi har prøvd å bruke samme annoteringer både i JSON-LD og RDFa, denne er derfor ikke avbildet.

På bildet til venstre ser vi hvordan verktøyet til google har tolket RDFa informasjonen som tilhører ontologien vår som vi har annotert en nettside med.

Dette er også kun et utdrag av informasjonen som blir vist, siden listen er svært lang.

*(RDFa-Lite data med bruk av ontologien)*

## Kommentarer angående kildekoden

Vi begynte først med å laste ned en webside fra whisky.com og forsøkte å begynne å tagge, men vi innså at det kom til å bli svært utfordrende å gjøre dette med høy kvalitet. Websiden har over ni tusen linjer kode, og annoterings-prosessen i seg selv var utfordrende nok i starten før en ble vandt til det. *Vi kontaktet derfor foreleser for å høre om det gikk ann å lage en mockup side som tok utdrag fra den originale kildekoden som vi trengte. Dette vart akseptert så lenge vi tok hensyn til den originale strukturen som tilhørte sidene.*

Dette gav oss mulighet til å presentere de samme dataene som var nødvendig for vår bruk av annotering med RDFa og JSON-LD, men på et par hundre linjer i stedet for rundt ni tusen.

Først ble det laget en enkel utgave av siden. Vi matet deretter inn informasjon om en whisky og begynte arbeidet med å tilføye RDFa og JSON-LD til denne. Etter vi var tilfreds med denne ene whiskyen, brukte vi dette som en mal for resten vi ville annotere. Eksempler på dette ble vist tidligere i teksten.

## Fordeler med den tilføyde semantikken

Med semantikken som er blitt lagt til tilbyr nettsidene nå rik og strukturert informasjon. Dette gjør for eksempel at søkemotorer kan enklere finne relevant informasjon på websidene, og presentere detaljerte data til brukerne. For eksempel kan produktinformasjon som alkoholprosent, smak osv. hjelpe brukeren raskere med å finne informasjonen de er ute etter.

Vi la både til RDFa-Lite og JSON-LD til nettsidene, dette gjør dataene mer søkbare, generelt sett mulige kategorisere og kan knyttes til både schema.org og ontologien vår (om vi hadde “hostet” denne ontologien på nettet)

Kanskje man googler whisky som har en spesiell type smak. Ved hjelp av den semantiske informasjonen kan websiden nå tilby denne dataen.

# Gruppearbeid og konklusjon

Generelt sett at “<https://todoist.com/>” og rombooking via UiB blitt mye brukte til å kunne møtest å diskutere hvordan vi skulle gå løs på selve oppgaven og hvilke målsetninger vi burde sette innenfor gitte tidsbegrensninger. Dette fungerte svært bra etter vi ble mer komfortable med Protegè som vi bygde ontologien vår i og hvordan vi skulle annotere sider skikkelig. Vi sørget for å sette opp en “Github-repo” til å holde all data som vi arbeidet med, slik at vi hadde versjonskontroll tilgjengelig og noe skulle gå skeis.

Samtidig som arbeidet med det rent tekniske pågikk ble det laget utkast av informasjon som vi ønsket å få med i rapporten. Vi førte og en logg av oppmøte under samlingene vi hadde under hele prosessen. Vi legger ved en logg av dette lenger nede som gir en viss indikasjon på deltagelse, det ble dog arbeidet relativt mye mellom disse møtene.

Hele prosessen har gitt oss et godt innblikk i hvordan man kan skape ontologier i Protegè, og kjøre SPARQL spørringer mot disse. I tillegg har vi lært å legge til semantisk informasjon mot websider med både RDFa-Lite og JSON-LD.

## Logg

24.10.17 - Gruppemøte

Deltagere: kanditat-157, kanditat-105, kandidat-122, kandidat-251

Vi satte opp en prosjektplan i Todoist, opprettet repository i github, og påbegynt rapport i Google Docs. Vi begynte å se på hva informasjon som ligger på enhetene på whisky.com

27.10.17 - Gruppemøte

Deltagere: kanditat-157, kandidat-251, kandidat-122

Satt opp alle klasser ut i fra informasjonen websiden har tilgjengelig.

31.10.17 - Gruppemøte

Deltagere: kanditat-157, kanditat-105, kandidat-251

Arbeid med ontologi i protege. Forsøkt å legge opp en test-flaske for å se hva vi mangler og hva som fungerer eller ikke fungerer.

01.11.17 - Gruppemøte

Deltagere: kandidat-251, kanditat-157

Arbeid med å legge opp properties i ontologien. Store utfordringer med å finne ut hva vi skal bruke fra whisky.com fordi det er svært varierende mellom typene om hva som ligger på de av informasjon. Enkelte viser knapt ABV, mens andre har distillation date, bottling date, bottle size, bottler, series, maturing osv. Enkelte har ikke en gang distillery eller brand.

06.11.17 - Lab-gruppe

Deltagere: kandidat-251, kanditat-157, kandidat-122, kanditat-105

Påbegynt arbeid med å sette opp sparql-queries. Tenkt opp et use-case for bruken av queryene og ontologien vi har laget.

08.11.17 - Gruppemøte

Deltagere: kanditat-157, kandidat-251, kanditat-105

Arbeid med å legge opp tasting, nosing og finish på en sample whisky for å demonstrere mulighetene ved querying.

09.11.17 - Gruppemøte

Deltagere:kanditat-157, kandidat-251, kanditat-105, kandidat-122

Arbeid med å lage use cases\scenario som vil være relevante og nyttige utgangspunkt for queryene som skal utformes.

10.11.17 - Gruppemøte

Deltagere: kanditat-105, kanditat-157, kandidat-251, kandidat-122

Påbegynt utforming av SPARQL-queries. Rydding i rapporten. Avtalte at vi jobber videre hjemme hver for oss med å få ferdig de ulike queryene.

13.11.17 - Lab-gruppe

Deltakere: kanditat-105, kandidat-122, kanditat-157, kandidat-251, kanditat-148

Påbegynt å utforme whisky sider med RDFA. Planlagt å lage en mal side som alle kan jobbe ut fra. Testing og øving på RDFa-annotering

16.11.17 - Gruppemøte

Deltagere: kandidat-251, kanditat-157, kanditat-105, kanditat-148

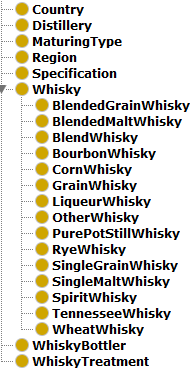
Kollektiv gjennomgang av vår første semantiske webside, før vi bruker denne som mal og legger opp flere typer whisky

# -OPPRYDDET OVER-

# GAMMEL VERSJON:

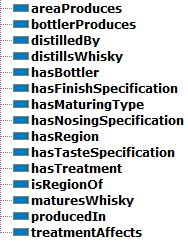
## Klasser

Etter at vi opprettet et nytt prosjekt i protege, begynte med å opprette klassene. De ulike Whisky-klassene ble lagt opp som disjoint, fordi en whisky ikke kan være flere typer av disse.



## Object properties

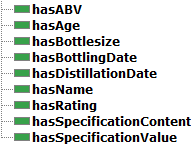
Deretter begynte vi med å opprette object properties, og å legge opp karakteristikk og beskrivelse av disse. Alle properties har en inverse-property for å gi mer muligheter når vi gjør spørringer uten å legge opp duplikatinformasjon Eksempelvis areaProduces er lagt opp som en inverse of producedIn.



## Data properties

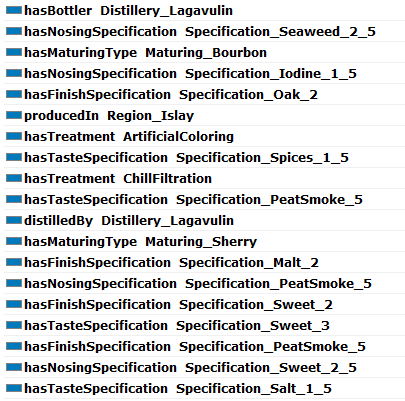
Vi begynte så med data properties. Dette er data som i svært varierende grad ligger tilgjengelig på de ulike whisky-typene, men vi legger så godt opp som mulig alt som er tilgjengelig.

De ulike feltene er knyttet opp til relevante datatyper. Eksempelvis er hasABV double, siden prosentene ofte har desimaler. Samme gjelder bottlesize, rating osv. For bottlingdate og distillation-date valgte vi å benytte date-time. Den krever dato i formatet <år>-<måned>-<dato>T<time>:<minutt>:<sekund>. Dette er selvsagt mer konkret enn det som er tilgjengelig på de ulike whiskyene, så vi går her til første dagen i måneden med 00:00:00.



I utgangspunktet hadde vi bestemt oss for å ikke ta med smakssystemet på whisky.com, fordi det var store hull i hva som var tilgjengelig på de ulike typene av whisky. Etter litt diskusjon internt i gruppen bestemte vi oss for å ta det med likevel, siden det kunne gi oss mer meningsfylte spørringer, og grundigere annotering i oppgave fem. Vi forenklet det ned til tasting, nosing og finish. Hver av disse kan ha en rekke “specification content” med respektive specification values. Specification av ulike smakstyper må legges opp som individer, og disse individene knyttes opp til whisky-flaskene.

Her er et eksempel på hvordan object property assertions blir seende ut for whiskyindivider.



## Individuals

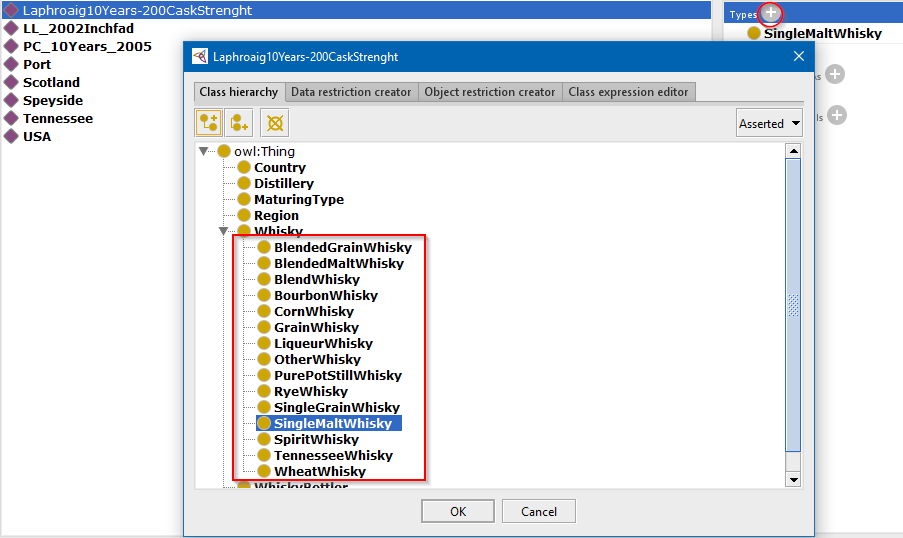
I starten la vi opp alle idividuals med navn som resource name. Dette ble etterhvert fryktelig rotete, og vi ble nødt til å ta i bruk prefix. Prefixene vi bruker i resouce names er Country, Region, Distillery, Maturing og Whisky. Dette gjorde det mer oversiktelig å jobbe i protege siden vi klarer å finne frem til riktige individuals etterhvert som det blir ganske mange. Under følger et eksempel på hvordan vi legger opp individuals i Protege.



### Legge opp whisky som individ i protege

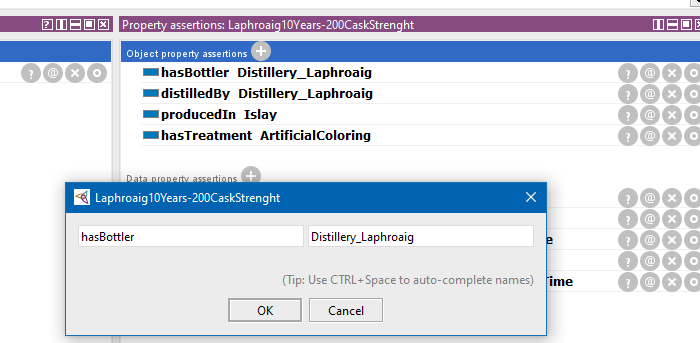
Legg først opp en individual med et gjenkjennelig navn. Ikke bruk spesialtegn. Bør innholde navnet på whiskeyen, men forkort gjerne merket. Eksempelvis Jack Daniels Gentleman’s Jack til JD\_Gentleman\_Jack.

Velg types og legg til typen. For eksempel SingleMaltWhisky. Dette kan du enkelt velge i klassehierakiet.



Velg deretter objet property assertions. Her må du legge til følgende:

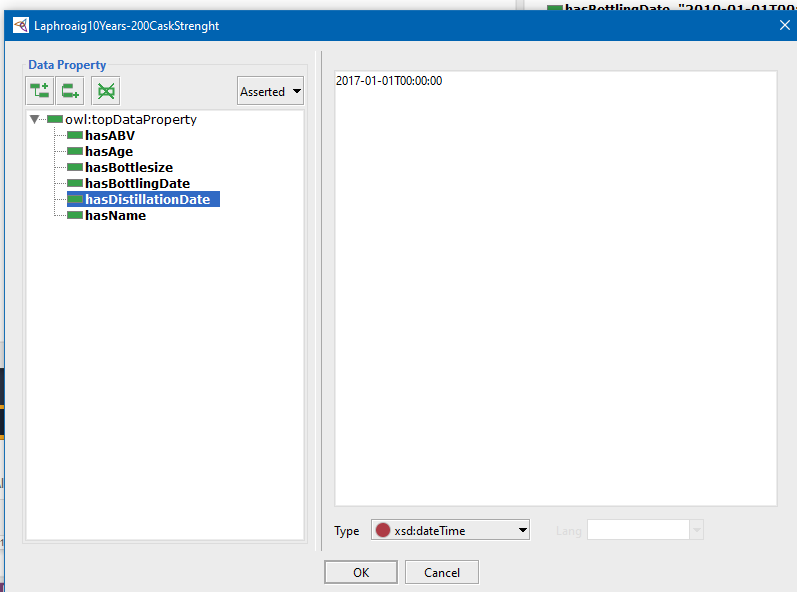
* producedIn
* hasBottler
* distilledBy
* hasMaturingType
* hasTreatment



Denne informasjonen må legges til som egne individer på samme måte som en whiskeytype, dersom den ikke eksisterer fra før. Det varierer fra type til type på whisky.com på hva som er tilgjengelig av informasjon, så her må vi bare hente ut det vi kan. Hvis det mangler informasjon må vi bare se bort i fra respektive object properties. producedIn skal knyttes mot regionen den er produsert i dersom dette er tilgjengelig. Da vil protege aukanditat-148atisk forstå at den også da er produsert i landet regionen tilhører, siden et region-individ må opprettes og knyttes til et land. Vi tar utgangspunkt i at hasBottler er det samme som brand. Derfor har disse ikke separate felter.

Videre er det flere data properties. Tilgjengelige er følgende:

* hasBottlingDate <datetime>
* hasDistillationDate <datetime>
* hasABV <double>
* hasAge <int>
* hasBottlesize <double>
* hasName <string>

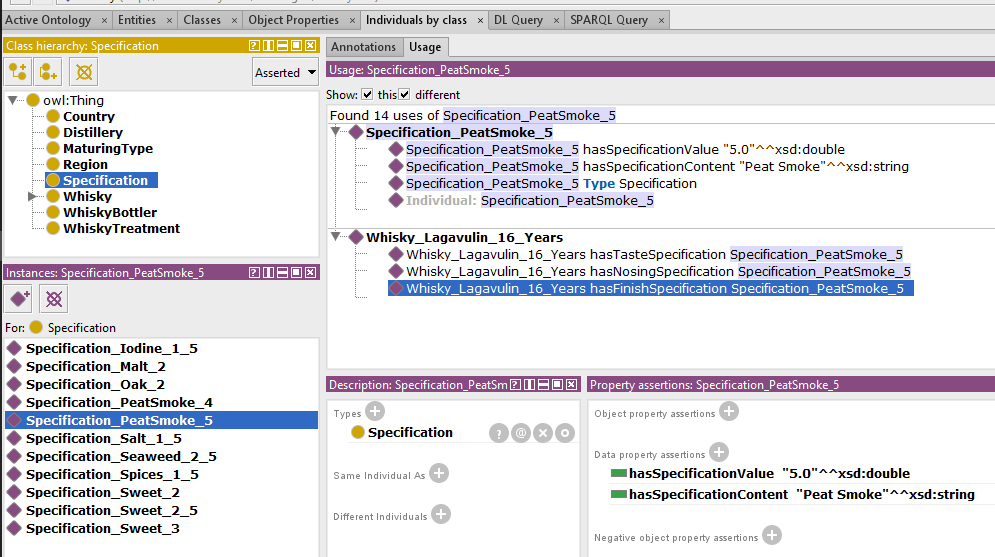


Dette legges til på whiskeyen, men riktig datatype må benyttes for de ulike feltene. Eksempelvis er hasABV double, på grunn av eventuelle desimaler. Bottling date\destillation date: Ved manglende informasjon på whiskey defaulter vi til 01 ved midnatt. dateTime 2016-01-01T00:00:00.

### Legge opp smak på whiskeyene

Som vi diskuterte tidligere i rapporten laget vi et rammeverk for å beskrive smak, lukt og finish på whiskyene. Derfor har vi opprettet en klasse som heter Specification. Videre har vi opprettet object properties som heter hasFinishSpecification, hastasteSpecification og hasNosingSpecification. Disse er knyttet opp mot domain Whisky med range Specification. Videre har vi opprettet individer som beskriver de ulike smakene, luktene og finish som et whisky individ kan ha.

Syntaksen her er Specification\_<taste\smell\finsih>\_<Type>\_<Verdi>. Vi har lagt opp dette for et par typer whisky for å kunne demonstere mer komplette søk med bred funksjonalitet. Siden whisky.com tilbyr ekstremt mange typer smakskategoriseringer vil det være en tidkrevende jobb å legge opp for alle whiskyene. Dette har ikke vi gjort, men vi har demonstrert at ontologien vår tar høyde for at det er mulig. Vi legger opp noen variasjoner på enkelte flasker for å demonstrere hvordan vi har løst det.



## 

## 

# 

# SPARQL

## Setting

Vi har sett for oss at vi har et scenario der vi i prosessen med å utvikle en app for et nettsted som omhandler whisky. Dataene er per dags dato relativt ustrukturerte. Via vår løsning er målet å kunne kjøre nyttige spørringer som kan ta høyde mange ulike aspekt ved de gitte whiskyene på nettstedet.

#### Case 1

Kunden ønsker å ha et enkelt recommender-system for å finne lignende whiskyer som brukeren også kan være interessert i, etter et søk allerede har forekommet.

* Finn whiskyer med lignende smak

***SELECT DISTINCT*** *?name*

***WHERE*** *{ ?whisky whi:producedIn ?place .*

*?place whi:hasName "Islay"^^xsd:string .*

*?whisky whi:hasTasteSpecification ?tasteSpecification .*

*?tasteSpecification whi:hasSpecificationContent ?originalContent .*

*?similarWhiskies whi:hasTasteSpecification ?otherSpecification .*

*?otherSpecification whi:hasSpecificationContent ?originalContent .*

*?similarWhiskies whi:hasName ?name .*

***FILTER*** *(?similarWhiskies != ?whisky) }*

Forklaring: Her ser vi et til at kunden har et søk der han ville ha tak i whiskyer som ble produsert i “Islay”. Etter dette spør vi hvilke smaker som tilhører de gitte whiskyene som kunden fann. Så spør vi hvilke av whiskyene i ontologien som har smaker til felles med de som kunden hadde søkt etter, og så viser vi navnet til disse.

* Finn whiskyer innenfor samme whisky-typer som det opprinnelige søket

***SELECT*** *?allNames*

***WHERE*** *{ {* ***SELECT DISTINCT*** *?whiskyTypes*

***WHERE*** *{?search whi:hasMaturingType ?maturingTypes .*

*?maturingTypes whi:hasName "Oak"^^xsd:string .*

*?search a ?whiskyTypes .*

*?whiskyTypes rdfs:subClassOf whi:Whisky . } }*

*?allEqual a ?whiskyTypes;*

*whi:hasName ?allNames }*

Forklaring: Her har vi et tilfelle der brukeren har hatt et søk etter whiskyer som har blitt lagret på tønne av slaget “oak”. Her henter vi ut hvilke typer whisky som ble lagret på en slik tønne, for så å finne alle whiskyer som har samme type som disse.

* Finn whiskyer fra samme produsent som det opprinnelige søket

***SELECT*** *?all*

***WHERE*** *{ ?whisky whi:hasTreatment ?treatment .*

*?treatment whi:hasName "ChillFiltration"^^xsd:string .*

*?whisky whi:hasABV ?abv .*

***FILTER****(?abv > 48)*

*?whisky whi:hasBottler ?bottler .*

*?all whi:hasBottler ?bottler .*

***FILTER*** *(?all != ?whisky) }*

Forklaring: Her ser vi et søk etter alle whiskyer som har behandlingen “Chill Filtration” og at whiskyene skal ha en alkoholprosent over 48%. Etter søket finner vi hvem som produserte whiskyene i søkeresultatet, og så finner vi alle andre whiskyer som de produserer som ikke er i resultatet som kunden fikk.

#### Case 2

I appen skal det også være mulig å kunne finne ut hva en gitt produsent eller destilleri sin gjennomsnittsrating er, på grunnlag av hva de har produsert.

* Gjennomsnittsrating på produsent

***SELECT AVG****(?ratings)*

***WHERE*** *{ ?whisky whi:distilledBy ?distillery .*

*?distillery whi:hasName "Blend-Scotch"^^xsd:string .*

*?whisky whi:hasRating ?ratings }*

* Gjennomsnittsrating på bottler

***SELECT AVG****(?ratings)*

***WHERE*** *{ ?whisky whi:hasBottler ?bottler .*

*?bottler whi:hasName "Douglas Laing"^^xsd:string .*

*?whisky whi:hasRating ?ratings }*

**Disse spørringene fungerer fint i Jena Fuseki, men i protege fungerer det ikke. Det ser ikke ut til at protege støtter AVG-funksjonen.**

#### Case 3

Kunden ønsker å kunne finne whiskyer basert spørringer med mange faktorer.

I vår ontologi har vi beskrevet whiskyene med følgende informasjon: *produsent, destilleri, etterbehandling, modning (les: cask), produksjonssted, lukt, smak, ettersmak, rating (vurdering), alkoholprosent, alder, navn, flaskestørrelse, destillerings dato og tappe dato*.

* Her er brukeren ute etter whisky typen SingleMaltWhisky med alder til og med 4 og over. Resultatet er sortert etter alder på whiskeyen i nedgående rekkefølge. Kolonnene navn og alder vises til brukeren.

***SELECT*** *?name ?age*

***WHERE*** *{ ?whiskies a whi:SingleMaltWhisky;*

*whi:hasAge ?age;*

*whi:hasName ?name;*

***FILTER****(?age >=4) }*

***ORDER BY DESC*** *(?age)*

* Her er brukeren også ute etter whiskytypen SingleMaltWhisky med alkoholprosent over 20. Resultatet er sortert etter alkoholprosent i stigende rekkefølge. I kolonnene vises whiskies, navn og alkoholprosent.

***SELECT*** *?whiskies ?name ?abv*

***WHERE*** *{ ?whiskies a whi:SingleMaltWhisky;*

*whi:hasABV ?abv;*

*whi:hasName ?name;*

***FILTER****(?abv >=20) }*

***ORDER BY ASC****(?abv)*

* Her har brukeren veldig lyst på en whisky som inneholder Peat Smoke. Men brukeren er veldig glad i å velge whiskier etter type sted der de tapper. Så resultatet er i rekkefølge synkende etter tapper.

***SELECT DISTINCT****?name ?bottler ?specContent*

***WHERE*** *{?Specification whi:hasName ?name;*

*whi:hasBottler ?bottler;*

***OPTIONAL****{?whiskies whi:hasSpecificationContent ?specContent}*

***FILTER****(?specContent="Peat Smoke"^^xsd:string) }*

***ORDER BY DESC*** *(?bottler)*

* Her har brukeren virkelig ikke lyst på Peat Smoke whisky, men er fortsatt veldig glad i å velge whiskier etter type sted der de tapper. Men denne gangen er resultatet i økende rekkefølge etter tapper

***SELECT DISTINCT****?name ?bottler ?specContent*

***WHERE*** *{?Specification whi:hasName ?name;*

*whi:hasBottler ?bottler;*

***OPTIONAL****{?whiskies whi:hasSpecificationContent ?specContent}*

***FILTER****(?specContent!="Peat Smoke"^^xsd:string) }*

***ORDER BY ASC****(?bottler)*

#### Case 4

Siden vi har støtte for å beskrive whiskyer med “rating” så har vi lyst å kunne sortere de ulike resultatene (se case 3) på denne variabelen.

* Denne queryen henter alle whiskeyene, og gir oss rating som er høyere enn 3, og sorterer resultatet etter høyest verdi først.

**SELECT** ?whisky ?rating

**WHERE** { ?whisky whi:hasRating ?rating

***FILTER****(?rating >=3)* }

**ORDER BY DESC** (?rating)

#### Case 5

I appen skal der være funksjonalitet som kan tillate en gitt bruker å finne en whisky (basert på faktorer gitt i case 3) men også kunne velge å få vist den som har høyest/minst alkoholprosent og høyest/minst lukt, smak eller ettersmak, eventuelt en kombinasjon av disse.

* Her har brukeren valgt å finne en whisky med høyest alkoholprosent innenfor alle Single Malt Whisky.

**SELECT ?name ?abv**

**WHERE { ?whiskies a whi:SingleMaltWhisky;**

**whi:hasABV ?abv;**

**whi:hasName ?name }**

**ORDER BY DESC (?abv)**

**LIMIT 1**

* Eks 2

# 

# 

# RDFa & JSON-LD

Vi begynte først med å laste ned en webside fra whisky.com og forsøkte å begynne å tagge, men vi innså at det kom til å bli svært utfordrende å gjøre dette med høy kvalitet. Websiden har over ni tusen linjer kode, og dette i seg selv hadde vært utfordrende nok. I tillegg bruker de tyske ord i koden for eksempel i klassebeskrivelser. Indenteringen er helt kaotisk, og generelt sett er koden full av feil.

Vi kom frem til at den beste måten å løse det på var å lage en mockup-side med noenlunde tilsvarende design og struktur som whisky.com. Her har vi mulighet til å presentere de samme RDFa-featurene, men på et par hundre linjer i stedet for ni tusen.

Først laget vi en enkel utgave av siden. Vi matet deretter inn informasjon om en whisky og begynte arbeidet med å tagge denne. Dette inkluderer all informasjon fra whisky.com som lå under den aktuelle flasken. Både det som vi hadde med i ontologien, og andre ting som whiskyID.

Websidene tilbyr nå rik, strukturert informasjon. Dette gjør for eksempel at søkemotorer som Google.com kan enklere finne relevant informasjon på websidene, og presentere riktige data direkte i søkeresultatene. For eksempel kan produktinformasjon som alkoholprosent, smak osv. hjelpe brukeren raskere med å finne informasjonen de er ute etter.

Med semantiske data er det også lettere for søkemotorer å finne relevant informasjon. Kanskje man googler whisky som har en spesiell type smak. Ved hjelp av semantisk informasjon kan websiden vår hjelpe med dette, fordi en ontologi kan bidra til å kategorisere smaken på en strukturert måte.

# 

# 

# Konklusjon

Når vi utformet ontologien lot vi oss inspirere av PizzaOntology-øvingsoppgaven vi fikk utlevert på lab noen uker før vi begynte på Whisky-prosjektet. Vi følte at dette var en gjennomtenkt ontologi, som ville gi oss et godt rammeverk for å få inn all nødvendig informasjon om produktet, og legger opp fornuftige relasjoner slik at vi ville ha et godt utgangspunkt for å gjøre spørringer. Før vi ville si oss fornøyd med ontologien, ville vi være sikker på at vi hadde et godt grunnlag for å kunne lage gode og meningsfulle spørringer.

Etter at ontologien vår ble godkjent internt i gruppa begynte vi arbeidet med å tenke ut et fornuftig scenario, og en rekke use cases for løsningen vår. Dette gjorde at vi kom opp med en rekke spørringer som både ville vise at vi hadde forstått hvordan det fungerer, og presentere gode eksempler på hva man kan oppnå.

Til slutt brukte vi en del tid på å researche RDFa og JSON-LD for å virkelig forstå teknologien før vi begynte. Vi knyttet opp ontologi fra schema.org og vår whisky ontologi på produktsiden. Dette testet vi mot rdfa.info play og rich snippet tool, for å se at informasjonen var knyttet opp på riktig måte.

Dette har gitt oss et godt innblikk i hvordan man kan skape ontologier i Protege, og kjøre SPARQL spørringer mot disse. I tillegg har vi lært å implementere både egne og eksterne ontologier mot websider med RDFa og JSON-LD.

Loggen som følger under gir en viss indikasjon på hvem som har bidratt i prosjektet.

# Logg

24.10.17

Gruppemøte

Deltagere: kanditat-157, kanditat-105, kandidat-122, kandidat-251

Vi satte opp en prosjektplan i Todoist, opprettet repository i github, og påbegynt rapport i Google Docs.

Vi begynte å se på hva informasjon som ligger på enhetene på whisky.com

27.10.17

Gruppemøte

Deltagere: kanditat-157, kandidat-251, kandidat-122

Satt opp alle klasser ut i fra informasjonen websiden har tilgjengelig.

31.10.17

Gruppemøte

Deltagere: kanditat-157, kanditat-105, kandidat-251

Arbeid med ontologi i protege. Forsøkt å legge opp en test-flaske for å se hva vi mangler og hva som fungerer eller ikke fungerer.

01.11.17

Gruppemøte

Deltagere: kandidat-251, kanditat-157

Arbeid med å legge opp properties i ontologien. Store utfordringer med å finne ut hva vi skal bruke fra whisky.com fordi det er svært varierende mellom typene om hva som ligger på de av informasjon. Enkelte viser knapt ABV, mens andre har distillation date, bottling date, bottle size, bottler, series, maturing osv. Enkelte har ikke en gang distillery eller brand.

06.11.17

Lab-gruppe

Deltagere: kandidat-251, kanditat-157, kandidat-122, kanditat-105

Påbegynt arbeid med å sette opp sparql-queries. Tenkt opp et use-case for bruken av queryene og ontologien vi har laget.

08.11.17

Gruppemøte

Deltagere: kanditat-157, kandidat-251, kanditat-105

Arbeid med å legge opp tasting, nosing og finish på en sample whisky for å demonstrere mulighetene ved querying.

09.11.17

Gruppemøte

Deltagere:

kanditat-157, kandidat-251, kanditat-105, kandidat-122

Arbeid med å lage use cases\scenario som vil være relevante og nyttige utgangspunkt for queryene som skal utformes.

10.11.17

Gruppemøte

Deltagere: kanditat-105, kanditat-157, kandidat-251, kandidat-122

Påbegynt utforming av SPARQL-queries. Rydding i rapporten. Avtalte at vi jobber videre hjemme hver for oss med å få ferdig de ulike queryene.

13.11.17

Lab-gruppe

Deltakere: kanditat-105, kandidat-122, kanditat-157, kandidat-251, kanditat-148

Påbegynt å utforme whisky sider med RDFA. Planlagt å lage en mal side som alle kan jobbe ut fra. Testing og øving på RDFa-annotering

16.11.17

Gruppemøte

Deltagere: kandidat-251, kanditat-157, kanditat-105, kanditat-148

Kollektiv gjennomgang av vår første semantiske webside, før vi bruker denne som mal og legger opp flere typer whisky.