Easter holidays
next lecture Friday 25. April
- Presentation on API
Status 4. April 2014

- Generic Scenario and Use Case ✓
- Verbal expression of query ✓
- Decision on OR
  - Prob.3.5 with SWRL
  - Prob.4.3 with SPARQL
  - Working on: apply SWRL / SPARQL on ontology
- Upcoming: Interface to a program to
  - OWL API Work with the knowledge
  - Java API...? 28.04.2014
- Final: Presentation of your integrated Mid May solution
Real time analysis

(1) (Marcel) Continuos C-SPARQL

(2) knowledge space

API

Example:

- train network
- schedule train set
- "real time info"
- program

Program with real-time information

2 min

7/5 min

typical update

/Day /Hour
http://cwi.unik.no/wiki/Applying_SWRL_to_your_ontology

SWRL examples

- Is the person running?

  \[
  
  \text{Person(?person)} \land \text{hasSpeed(?person, ?speed)} \land \text{swrlb:greaterThanOrEqual(?speed, 10)} \land \text{hasStatus(?person, running)}
  \]

- Is Susana walking?

  \[
  
  \text{Person(Susana)} \land \text{hasSpeed(Susana, ?speed)} \land \text{swrlb:greaterThanOrEqual(?speed, 1)} \land \text{hasStatus(Susana, walking)}
  \]

- which songs the person likes

  \[
  
  \text{Person(?person)} \land \text{hasPreference(?person, ?prefer)} \land \text{A Music(?music)} \land \text{hasStyle(?music, ?prefer)} \land \text{like(?person, ?music)}
  \]

- LowRisk state range for the persons who have a passive action and a heart rate between 81 – 120

  \[
  
  \text{Person(?p)} \land \text{swrlb:LowRiskState(?n)} \land \text{A Passive(?y)} \land \text{hasAction(?x, ?y)} \land \text{hasHeartRateOf(?x, ?z)} \land \text{swrlb:greaterThanOrEqual(?z, 81)} \land \text{swrlb:lessThanOrEqual(?z, 120)} \land \text{hasHealthCondition(?x, ?n)}
  \]

Ontology examples

- OWL ontologies Code:
  - http://www.co-ode.org/ontologies/
  - http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary

RDF examples

- RDF Code: http://justinian.leibnizcenter.org/MetaLex/metalex-cen.owl
Query in Protégé: SPARQL

- Syntax based

(2) - DL Query tab

- More graphics based

DL Query tab

The DL Query tab provides a powerful and easy-to-use feature for searching a classified ontology. It is a standard Protégé 4 plugin available both as a tab and also as a view widget that can be positioned into any other tab. The query language (class expression) supported by the plugin is based on the Manchester OWL syntax, a user-friendly syntax for OWL DL that is fundamentally based on collecting all information about a particular class, property, or individual into a single construct, called a frame.

http://answers.semanticweb.com/questions/26501/owl2query-plugin-for-protege-43

But this, does: PREFIX pre0:http://www.onto-au.edu/ontologias/estudiante-generico# SELECT ?x ?y WHERE { ?x pre0:tieneHabilidad ?y . }

This is valid even if you have checked your ontology IRI - prefix line in the upper grid.
class in the query like this:

Of course, these are extremely simple queries; the Manchester syntax is much more capable.

Literal constants can be expressed with type by using `^^` and then the type:

- `hasAge value "21"^^long`

Or, a more general expression that uses type:

- `hasAge some int`

Following are just a few more examples to get you going:

- `hasChild some Man`
- `hasSibling only Woman`
- `hasCountryOfOrigin value England`
- `hasChild min 3`
- `hasChild exactly 3`
- `hasChild max 3`

1) translate verbal query → DL Query

2.1. what is the syntax in SPARQL

2.2. API to use the DL Query
First

- Protégé 4.3
- Reasoner
  - The «quality test» of your ontology.
  - Crtl+R
  - Inconsistency
  - Broken Rules
    - Ann and Per have the same mother.
    - They are not siblings.
      - Illegal rule

Hernát
SPARQL Syntax

- PREFIX
  - What Schema to use?
  - Classes with attributes.
- SELECT
  - What do you want?
- WHERE
  - Matches your criteria
- Ex.
  - SELECT{cars}
  - WHERE{Colour is «Blue»}

RDF: can create own ":" 

select all people
select all people with age > 18
My Ontology

Add: Pål

has location

= Participant

has event

selected

Oslo
Find all Instances/Individuals

SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX osw: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://www.semanticweb.org,andreas/ontologies/2014/2/untitled-ontology-7#>

SELECT ?x
WHERE { ?x rdfs:subClassOf* :ContextAwareCalendar .
         ?y rdfs:subClassOf* ?x }
```

"X" all anchor
Many cursor
Lists all subclasses

<table>
<thead>
<tr>
<th>x</th>
<th>Per</th>
<th>Lise</th>
<th>Ole</th>
<th>Oslo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event/Activity</td>
<td></td>
<td>Fest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event/Activity</td>
<td></td>
<td>Meeting</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Same as previous, asks only for people
Who is in Oslo?

```
WHAT IS WHAT?

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

SELECT distinct ?instance ?property ?object 
WHERE { ?instance a ?Person.
  ?object a ?HasLocation.
  ?property a ?HasPerson.
  ?object a ?Oslo . }
ORDER BY DESC(?property)
```
Not: archive in belows
all people in Oslo or Bergen
Next step: understand query syntax