TOOLS FOR MIXED REALITY IN SMART ENVIRONMENTS

LAB 4: Decision-making

November 12, 2020

Ph.D. Esteban Guerrero
esteban@cs.umu.se
AGENDA

• Decision-making with SWRL
• Smart environments key aspects for engineering
• Built-in platforms for BDI and simulation agents engineering
BEFORE START...

• Any question about topics of the previous lecture/lab?

Go to www.menti.com and use the code 39 57 81 9
Micro Survey

Question 1:

• Can you write two (or more) key words that were interesting from previous lecture/laboratory?

Go to www.menti.com and use the code 3957819
MICRO SURVEY

Question 2:

- What do you consider is missing from previous lecture/laboratory?

Go to www.menti.com and use the code 3957819
DECISION-MAKING WITH SWRL IN DIFFERENT LANGUAGES
THE GOAL

Given a sensor-based input, how a decision is produced?
SWRL WITH DIFFERENT LANGUAGES
SWRL IN JAVA

SWRLAPI
- https://github.com/protegeproject/swrlapi

Demo: https://github.com/esteban-g/swrl-queries
SWRL IN PYTHON

Owlready2

- [https://pypi.org/project/Owlready2/](https://pypi.org/project/Owlready2/)

Demo: drug.py

```python
from owlready2 import *

# Create the ontology from scratch
onto = get_ontology("http://test.org/onto.owl")

with onto:
    class Drug(Thing):
        def take(self):
            print("I took a drug")

    class ActivePrinciple(Thing):
        pass

    class has_for_active_principle(Drug >> ActivePrinciple):
        python_name = "active_principles"

rule = Imp()
rule.set_as_rule("Drug(\(d\), price(\(p\), \(n\)), number_of_tablets(\(n\), \(n\)), divide(\(p\), \(p\), \(n\)) \rightarrow price_per_tablet(\(d\), \(p\))")

sync_reasoner_pellet(infer_property_values=True,
                      infer_data_property_values=True)

print("\rightarrow drug0 drug price per tablet\):", drug0.price_per_tablet)
```
SMART ENVIRONMENTS ENGINEERING PERSPECTIVE
KEY ASPECTS TO CONSIDER DURING THE DESIGN

1. The agent’s knowledge (agent=smart environment).
   • type of information: context, user profile, etc.
   • status of information: incomplete, inconsistent, etc.
2. The data input.
   • available data: sensor-based, manual/user input, etc.
   • status of data: incomplete, uncertain, etc.
3. The output.
   • feedback concurrency: during or after a user performs an activity.
   • feedback modality: audio, visual, haptic, etc.
4. The decision-making algorithm
   • Computational and time consumption
KEY ASPECTS TO CONSIDER DURING THE DEVELOPMENT

From a software engineering perspective, the links between modules are key.

https://hubs.mozilla.com/
INPUT
- Limited video
- Human activity skeleton vectors
- Supervised ML: AdaBoost + random forest

OUTPUT
- feedback during skiing
- feedback modality: audio and visual
- Projected AR
- C# + Python

EXAMPLE CROSS COUNTRY SKIING

DECISION MAKING
- Answer-set programming
- DLV system for reasoning

KNOWLEDGE
- X-country skiing ontology
- Coaches (humans) made rules
- OWL/RDF

- ACTION
- ACTIVITY
- OBJECT
- PHYSICAL OBJECT
- MENTAL OBJECT
- OBJECTIV E

hasLocation
ConsistsOf
isDefinedBy
is-a
is-a
is-a
is-a
is-a
is-a
is-a
is-a
is-a
OTHER ALTERNATIVES FOR BDI AGENTS
BDI PLATFORMS

1. JaCaMo
   URL: http://jacamo.sourceforge.net
   Language: AgentSpeak + Java

2. GAMA platform
   URL: https://gama-platform.github.io
   Language: GAML language
   Demo: GAMA firefighter agents
BDI-SOCIAL SIMULATION PLATFORMS

1. Netlogo
   URL: https://www.netlogoweb.org/
   Language: Netlogo

2. StarLogo Nova
   URL: https://www.slnova.org/
   Language: Netlogo + graphical
   Demo: Dragon eating elephants
   https://www.slnova.org/esteban_g/projects/772397
OTHER RESOURCES
OWLREADY2 DOCS


```python
>>> from owlready2 import *
>>> onto_path.append("/path/to/your/local/ontology/repository")
>>> onto = get_ontology("http://www.lcofleurdunormal.fr/static/_downloads/pizza_onto.cwl")
>>> onto.load()

```class NonVegetarianPizza(onto.Pizza):

  ```
  ...  equivalent_to = [
  ...    onto.Pizza
  ...    & (onto.has_topping.some(onto.MeatTopping)
  ...    | onto.has_topping.some(onto.FishTopping)
  ...    )]
  ...

  def eat(self): print("Beurk! I'm vegetarian!")
```
RDFLIB JS

Javascript library for working with RDF files, e.g. ontologies

- https://github.com/linkeddata/rdflib.js