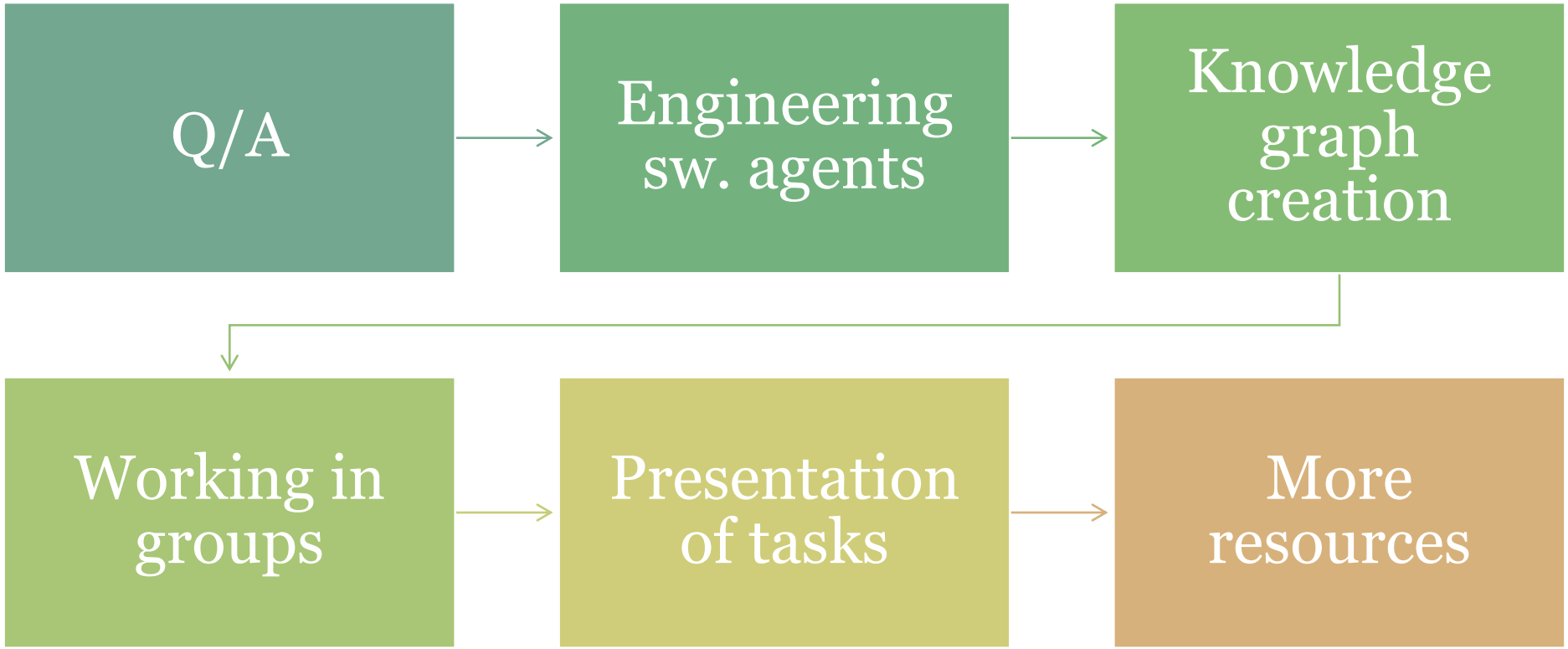
HOW TO BUILD A SOFTWARE AGENT?

Lab. session 2 – November 5 2020.

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

UMEÅ UNIVERSITY

CONTENT


Q/A
Engineering sw. agents
Knowledge graph creation
Working in groups
Presentation of tasks
More resources

QUESTIONS?

UMEÅ UNIVERSITY

BEFORE START…

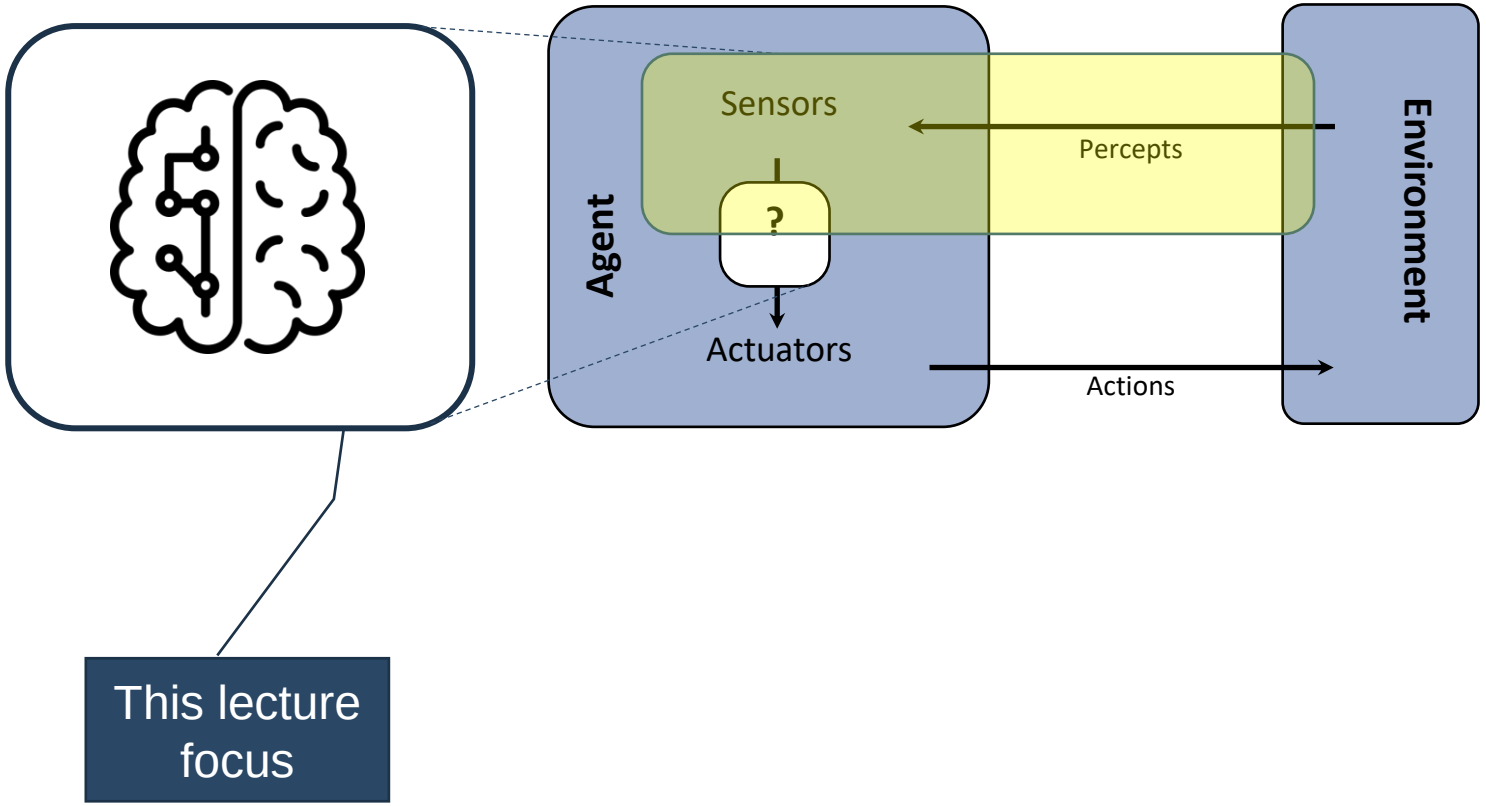
- Any question about the lecture?
- Any suggestion or petition?

Go to **www.menti.com** and use the code **63 46 25 7**

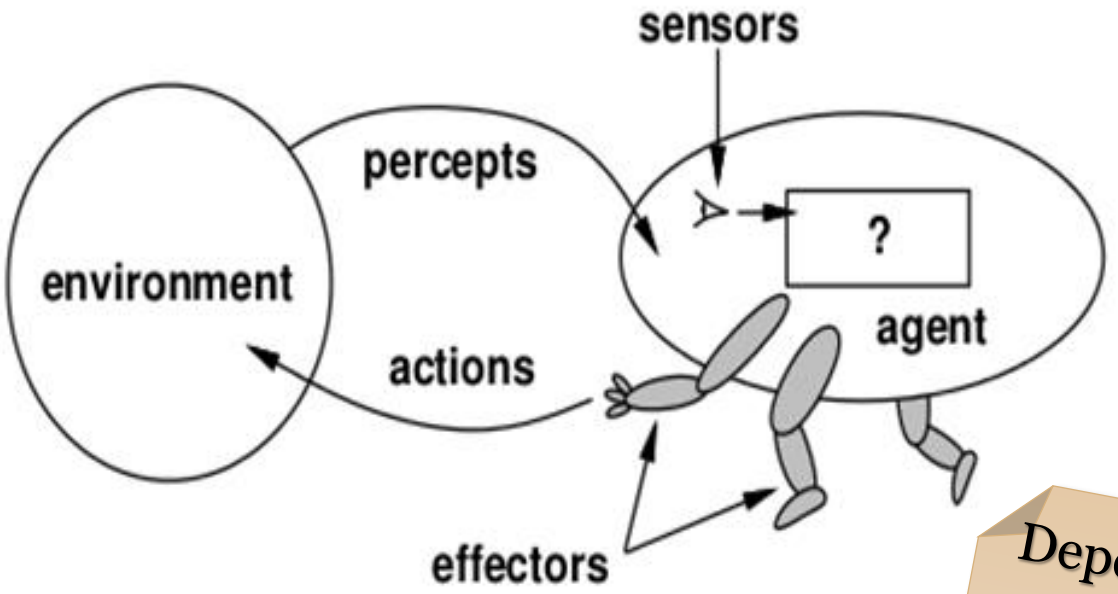
ENGINEERING AGENTS

UMEÅ UNIVERSITY

TOOLS FOR BUILDING AGENT'S MODULES



TOOLS FOR BUILDING AGENT'S MODULES



Depends on the problem.
Tools selection depends on ways to solve that problem

TOOLS

BDI Frameworks

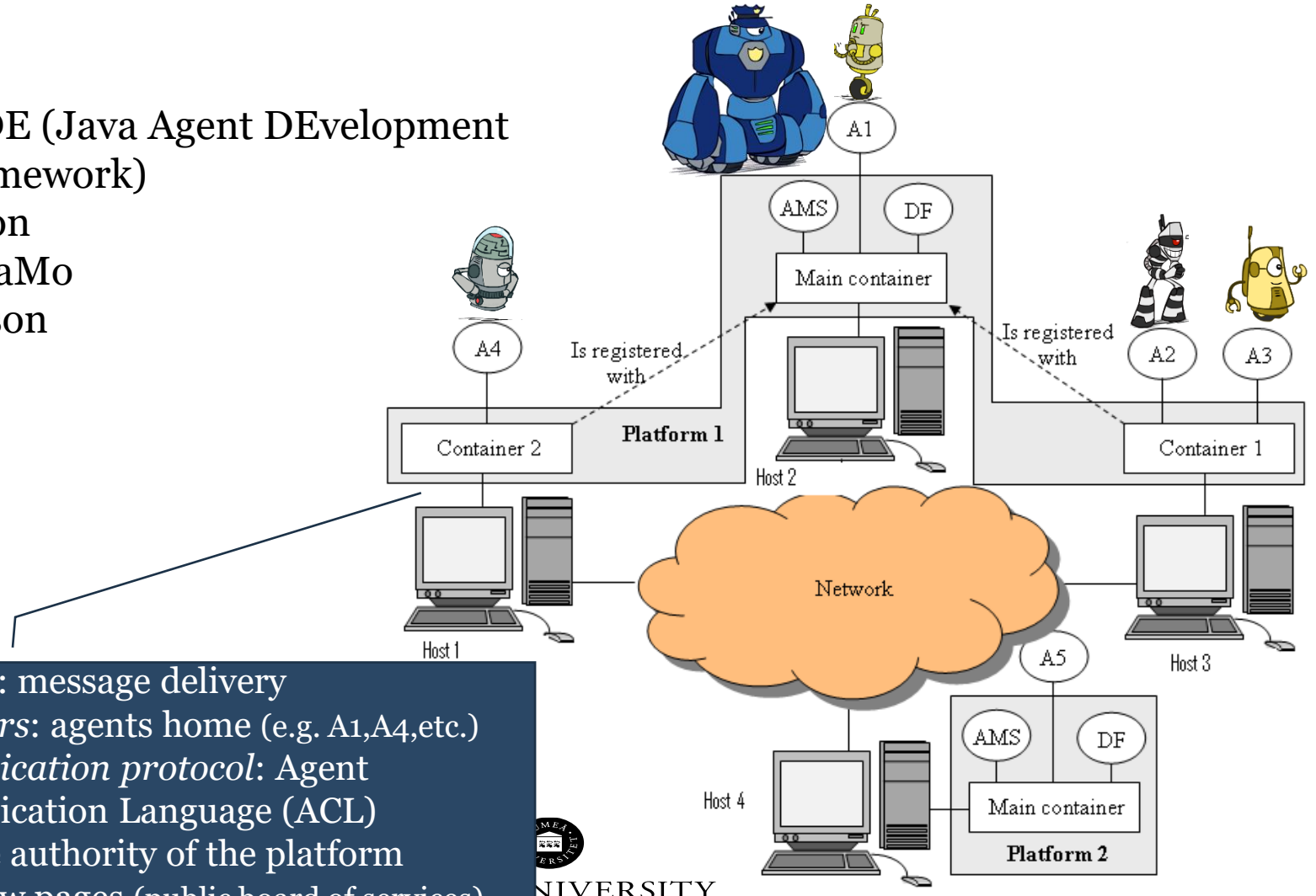
- JADE (Java Agent DEvelopment
- JaCaMo (Jason,Cartago,Moise)
- Jason
- Jack
- JS-son
- ROS (Robot Operating System)

Specialized tools/sw

- Comercial AI-based tools: Amazon aws, Google AI, Facebook AI, IBM Watson, Microsoft Azure AI, etc.
- Open source
  - **Python**: nltk, OpenCV, pandas, OpenAI Gym, etc.
  - **Java** libraries: Caffe, Deeplearning4J, etc.
  - **R** packages: scikit-learn, DataExplorer, MLR, parsnip, Ranger, purrr, etc.

TOOLS (BDI FRAMEWORKS)

- JADE (Java Agent DEvelopment Framework)
- Jason
- JaCaMo
- JS-son
- …



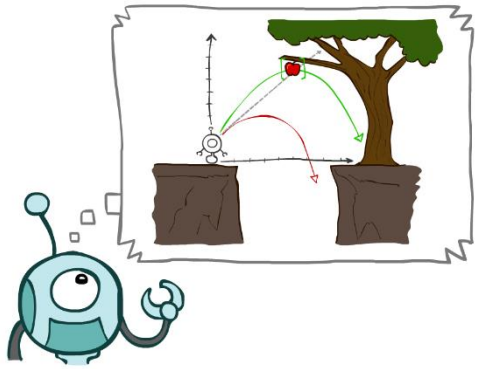
Platform: message delivery
Containers: agents home (e.g. A1,A4,etc.)
Communication protocol: Agent Communication Language (ACL)
AMS: the authority of the platform
DF: yellow pages (public board of services)

JADE Architecture Overview. https://jade.tilab.com

TOOLS (BDI FRAMEWORKS)

JaCaMo=**Ja**son for programming autonomous agents +
Cartago for programming environment artifacts +
Moise for programming multi-agent organisations

http://jacamo.sourceforge.net/



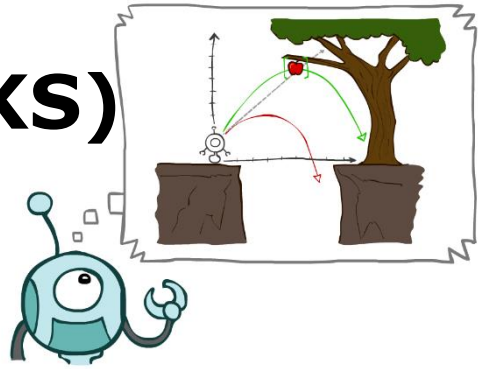
Jason

Cartago

Moise

CArtAgO: Common ARTifact infrastructure for AGents Open environments
Moise: Model of Organization for multI-agent SystEms
Jason: Java-based Agentspeak interpreter used with Saci for multi-agent distribution Over the Net

TOOLS (BDI FRAMEWORKS)



Jason

by perception

beliefs annotated with source(percept) are automatically updated accordingly to the perception of the agent

by intention

the operators + and - can be used to add and remove beliefs annotated with source(self)

```
+lier(alice); // adds lier(alice)[source(self)]
-lier(john); // removes lier(john)[source(self)]
-+lier(john); // updates lier(john)[source(self)]
```

Types of goals

- Achievement goal: goal *to do*
- Test goal: goal *to know*

Syntax

Goals have the same syntax as beliefs, but are prefixed by
! (achievement goal)
? (test goal)

Example (initial goal of agent Tom)

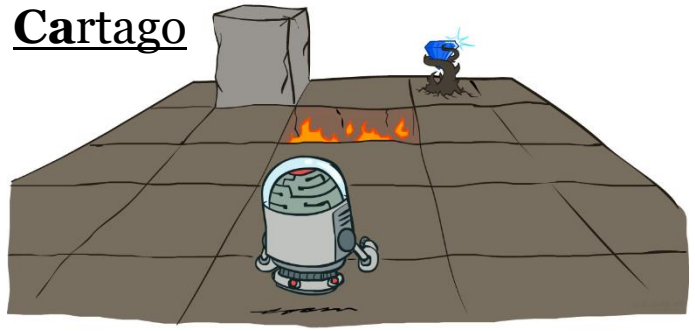
```
!write(book).
```

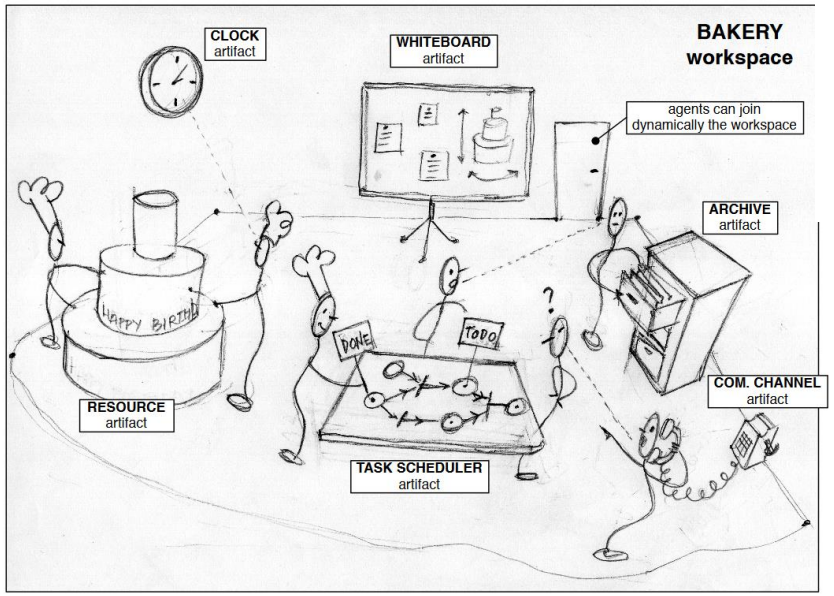
An AgentSpeak plan has the following general structure:

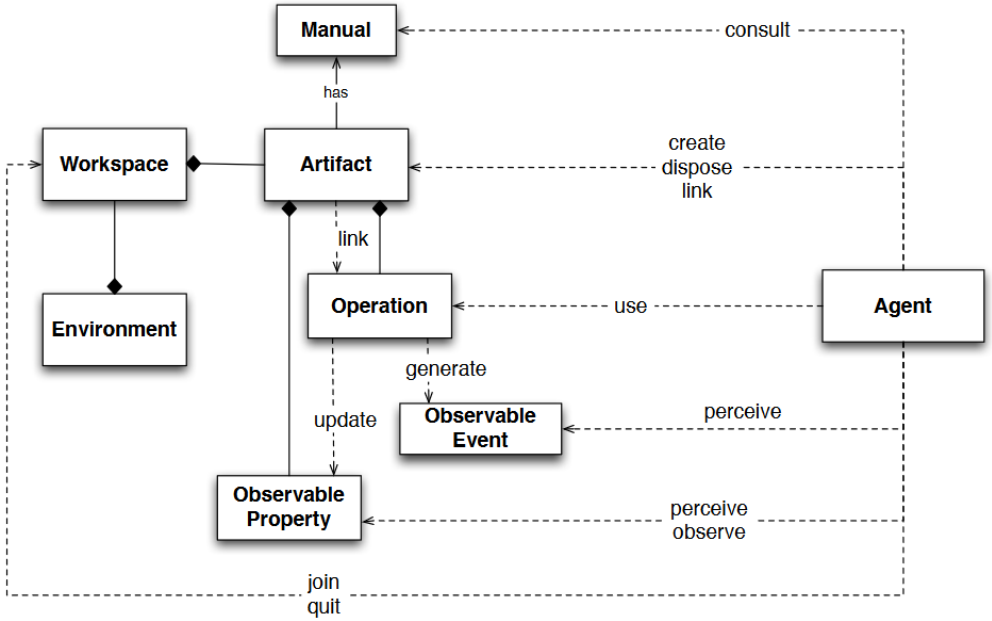
triggering_event : context <- body.

TOOLS (BDI FRAMEWORKS)

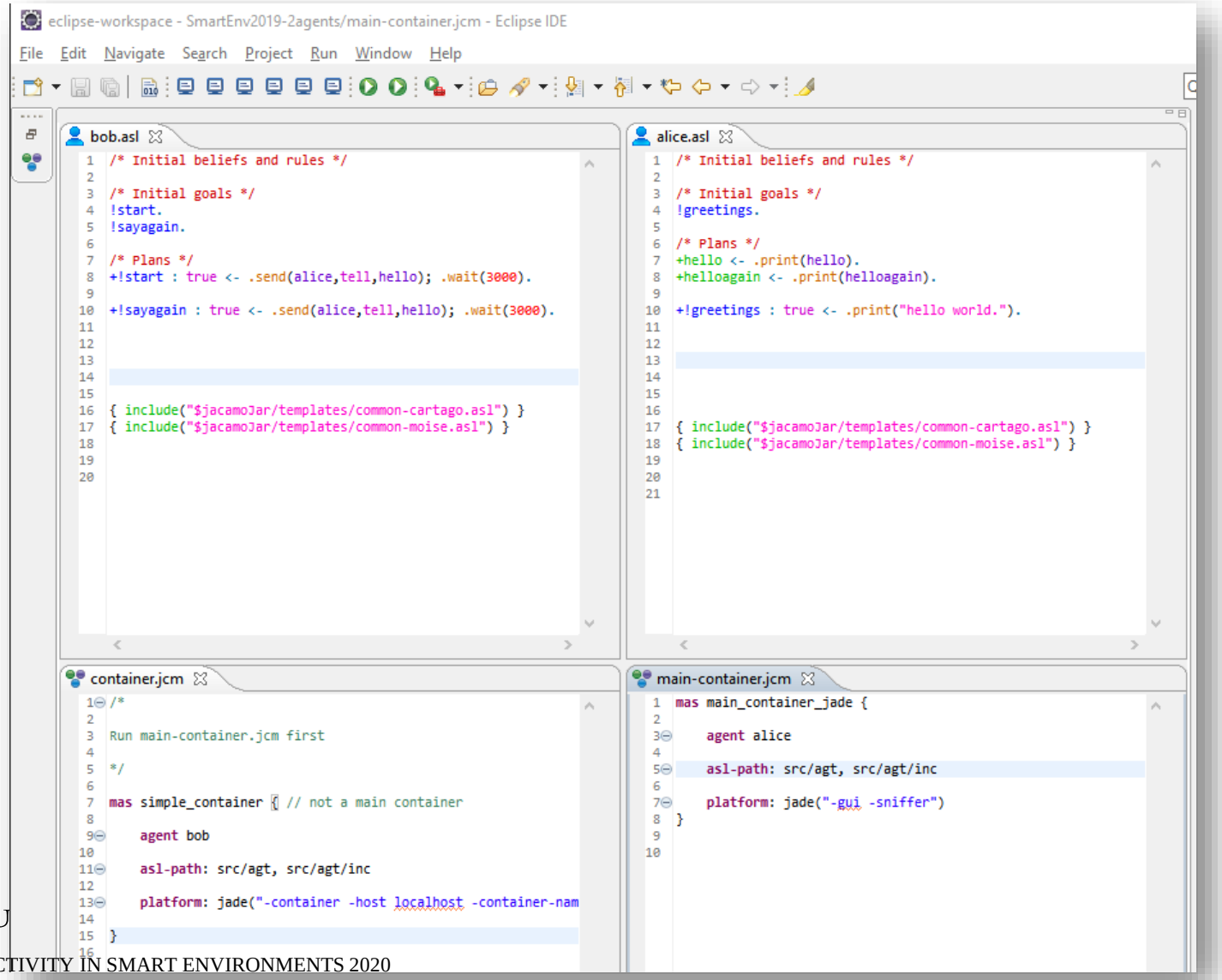
Cartago







JACAMO DEMO



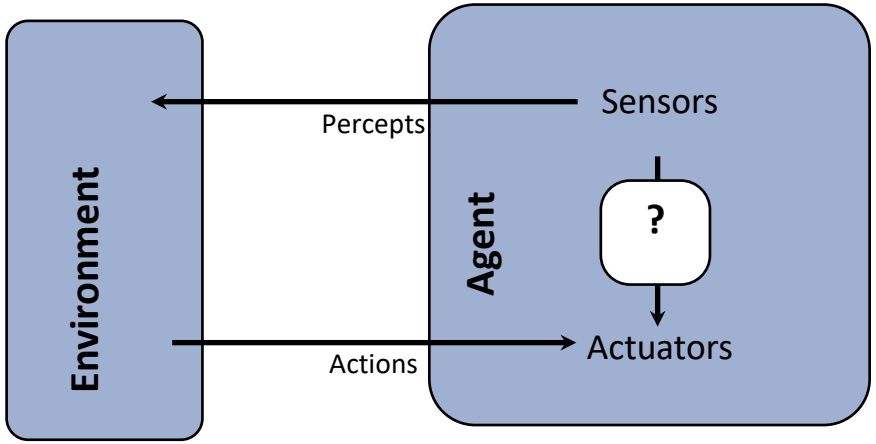
Demo: JaCaMo multi-agents

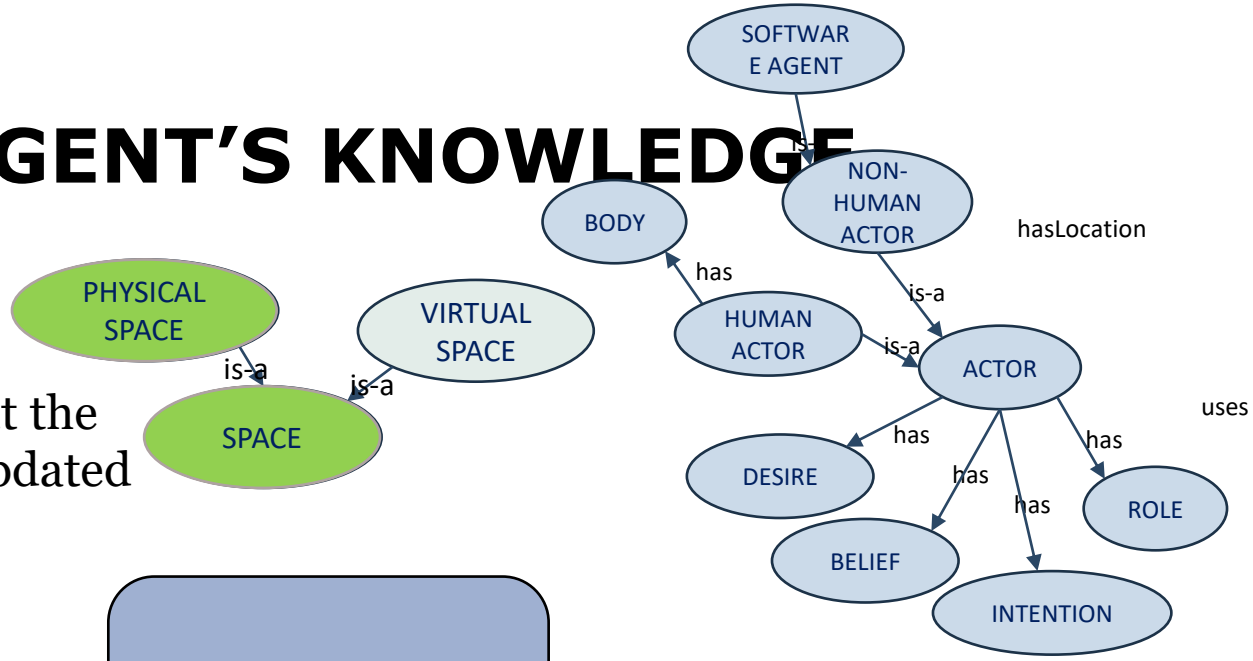
KNOWLEDGE GRAPH CREATION

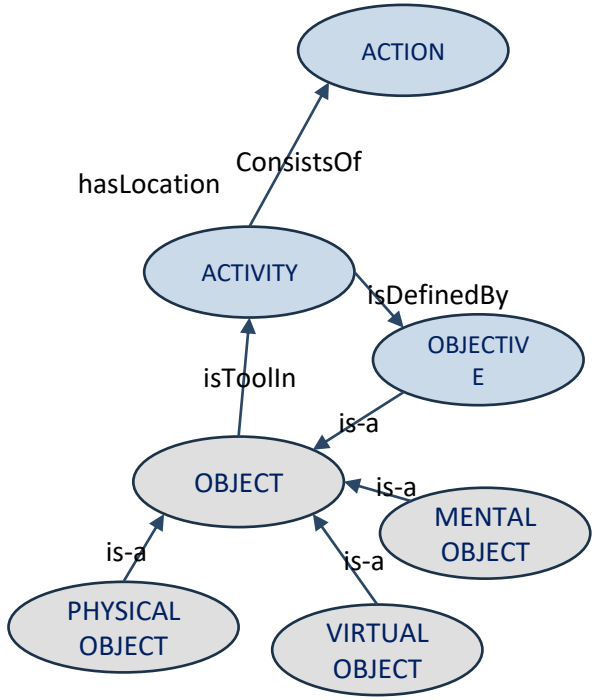
UMEÅ UNIVERSITY

THE AGENT'S KNOWLEDGE

Information about the environment is updated

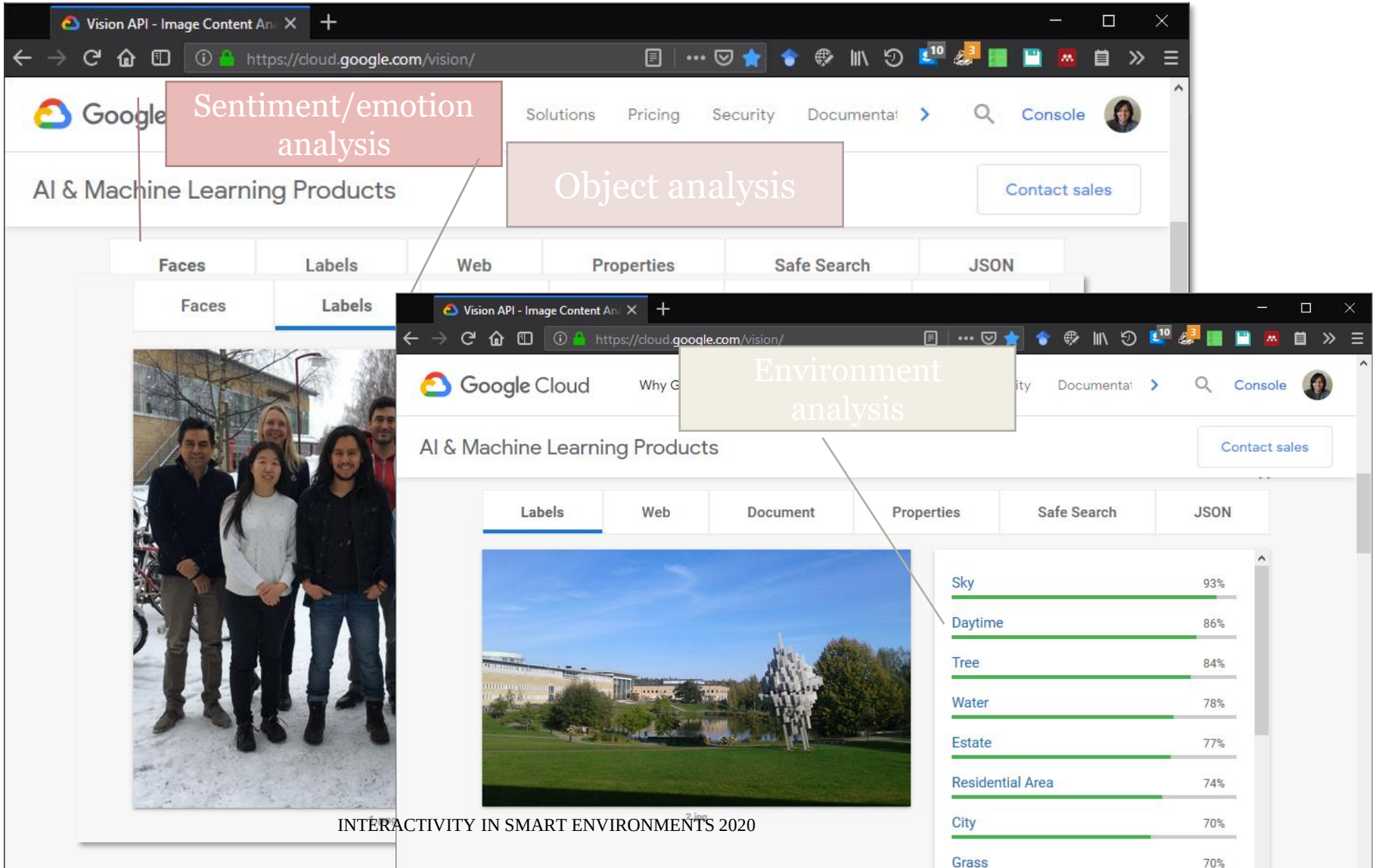




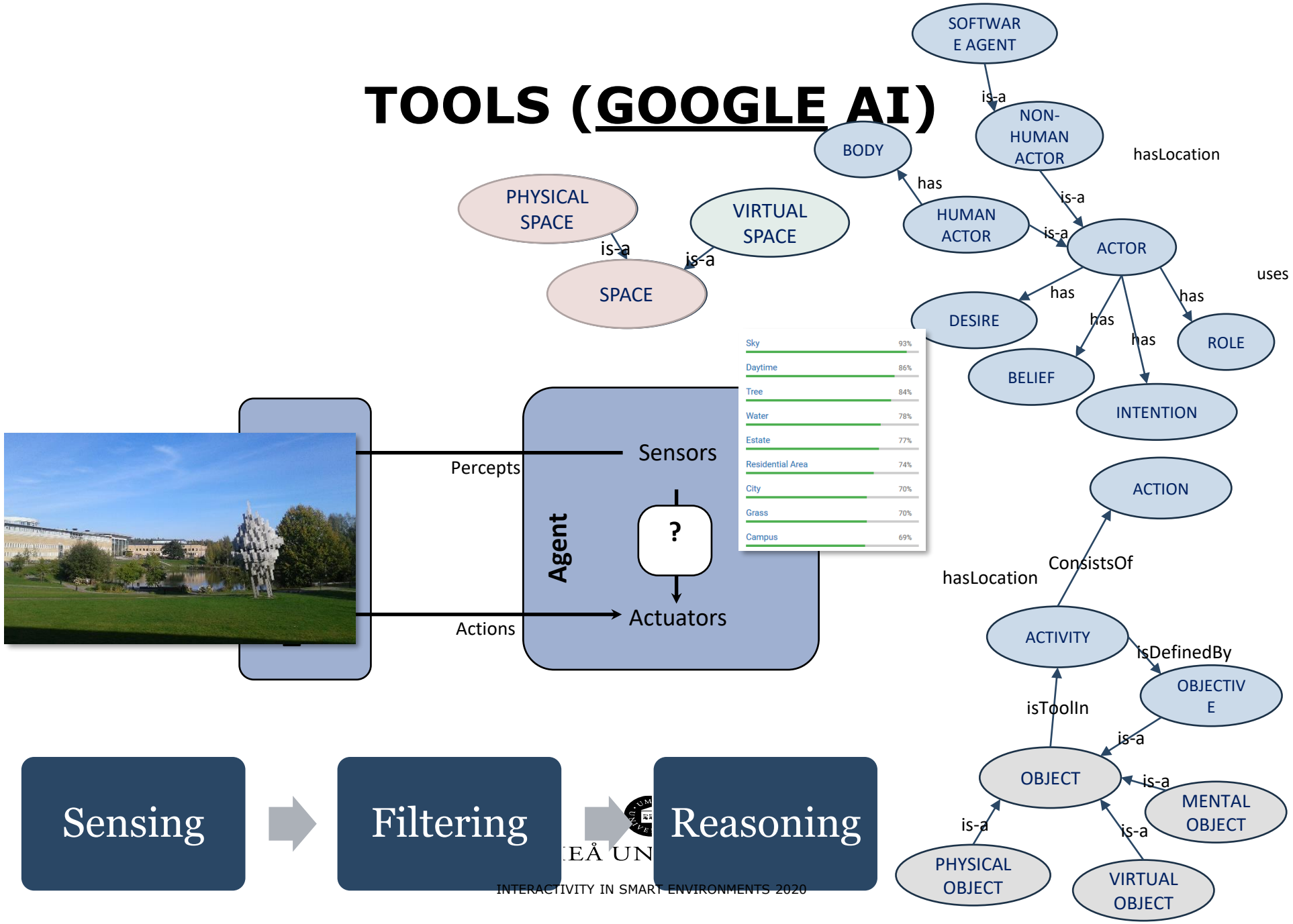


TOOLS (GOOGLE AI)

Demo: Google API vision



TOOLS (GOOGLE AI)

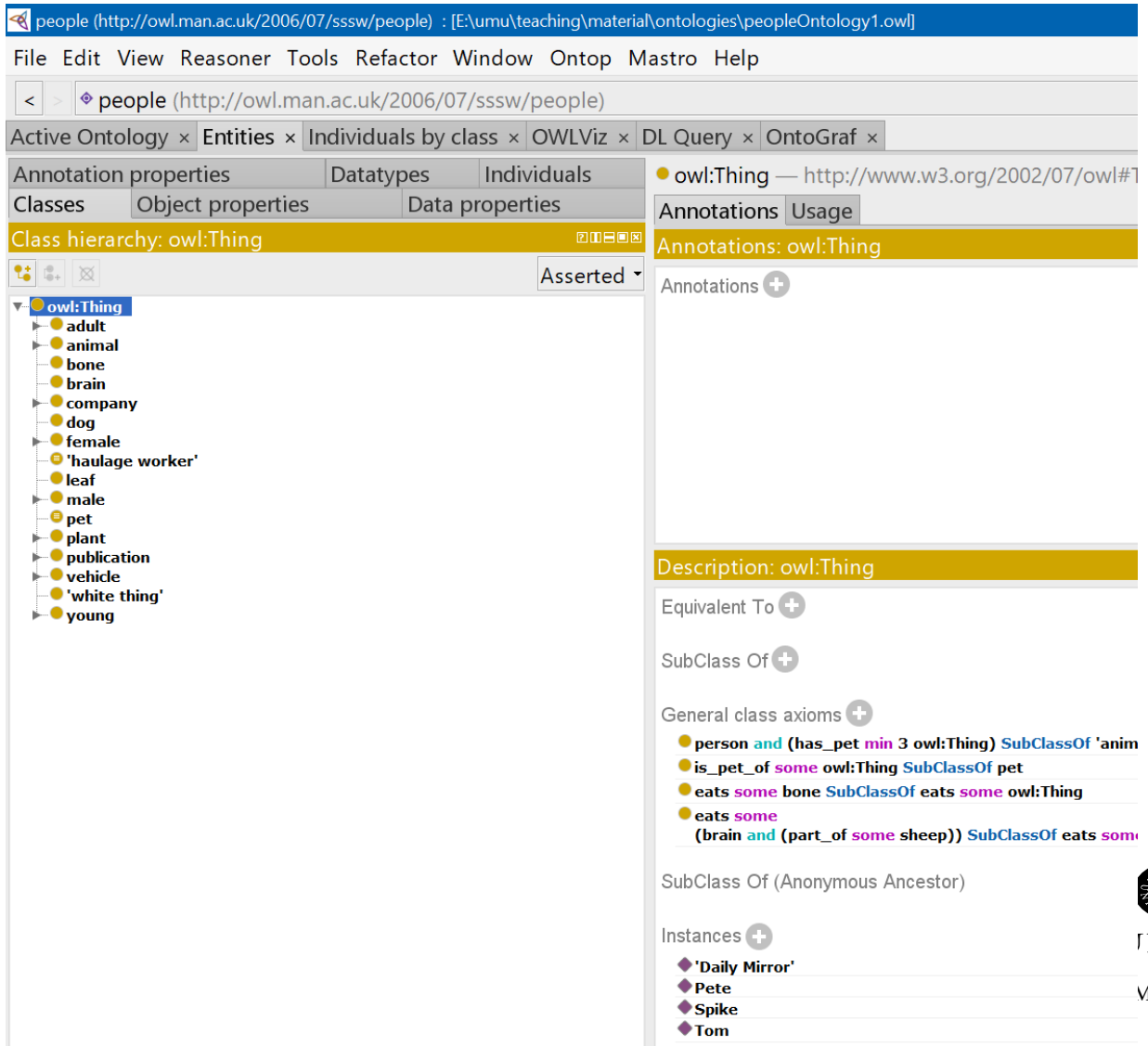


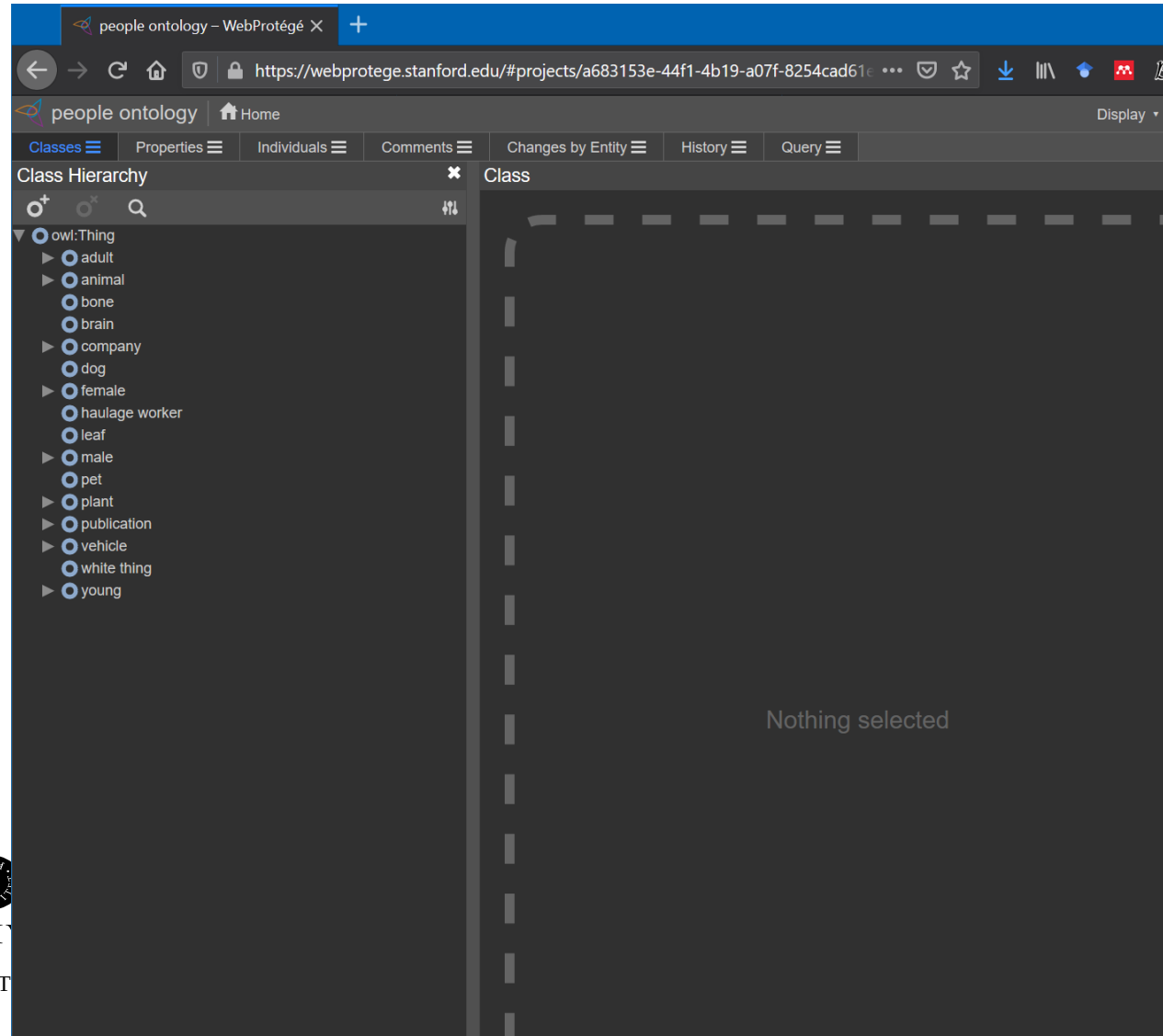
Sensing → Filtering → Reasoning

WEB PROTÉGÉ

UMEÅ UNIVERSITY

PROTÉGÉ VERSIONS





TASK 2
BUILDING A KNOWLEDGE BASE COLLABORATIVELY

Adding extra information about sensing

Make the first query

UMEÅ UNIVERSITY

COLLABORATIVE TASK

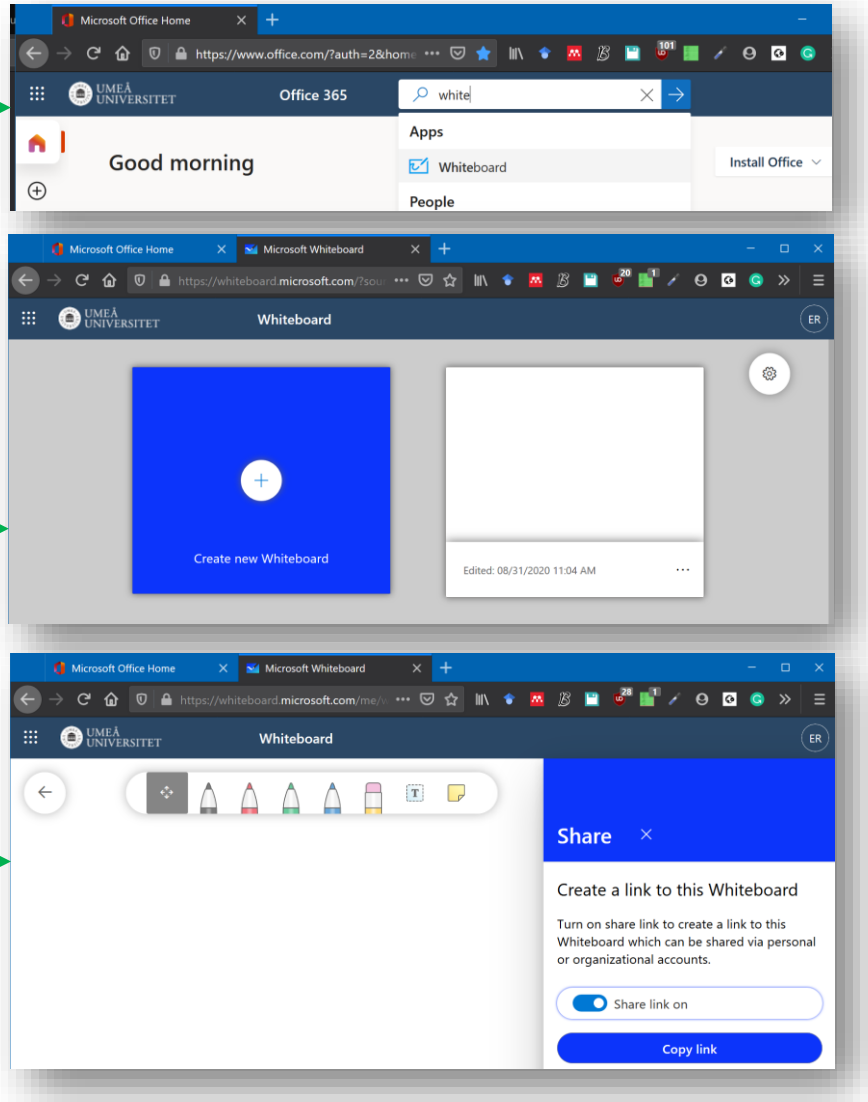
Procedure:

- Split the group in Zoom rooms to form teams **–randomly-**
- Select a specific running example of a smart environment
    1. Examples: an older adult living at home with some specific needs (smart home environment), a tourist looking for specific city places with particular needs (smart city), a mixed reality pet that work as a companion of a person with specific needs (smart virtual environment), etc. **–please imagine other scenarios**
    1. **Re-use the previous example - maybe**
- Knowledge graph collaboratively building based on the example
    2. In the Zoom room one person opens the Zoom Whiteboard (Share Screen->Whiteboard-> Share)
    3. Create the taxonomy of entities (nodes) with relationships ***(is-a)*** using a color
    4. Create semantic relationships (e.g. **has_pet, is_pet_of**) with other color
    5. Add some *individuals* in other words, examples or instances of those entities with other color
    6. Add names of the group member in the top-left of the drawing
    7. Add a title to the graph, example: older adult smart environment graph, tourist smart city graph, etc.
- Save the drawing graph locally
    8. In Zoom click in Save on top of the Whiteboard
- Answer questions (next slide), take notes about those answers. Then present those answers.
- **Time 10 minutes**

COLLABORATIVE TASK

Alternatively, for sharing if Annotations and Whiteboard in Zoom is limited:

1. All the team members go to 0365.umu.se/
2. Login with UmU credentials
3. One person creates and opens a new Whiteboard
4. Share the link of the new whiteboard with the other members via Zoom
5. Make the knowledge graph collaboratively
6. Save the graph



COLLABORATIVE TASK

Procedure:

- Open Web protégé: https://webprotege.stanford.edu/
- Create an account
- Create the taxonomy (the families of knowledge entities)

•Time 10 minutes

COLLABORATIVE TASK

Questions

- What information of the graph is *necessary*?
- What parts of the graph *change on time* (dynamic smart environment variables)?
- What information may be *uncertain* or *incomplete*?
- What sensors can be used to capture that information?

PRESENTATION OF THE RUNNING EXAMPLE AND KNOWLEDGE GRAPH

UMEÅ UNIVERSITY

PRESENTATION OF EXAMPLE AND KNOWLEDGE GRAPH

Procedure:

- Join the main Zoom session
- Different members of the group present the running example.
- Others present the graph.
- Other the reflections

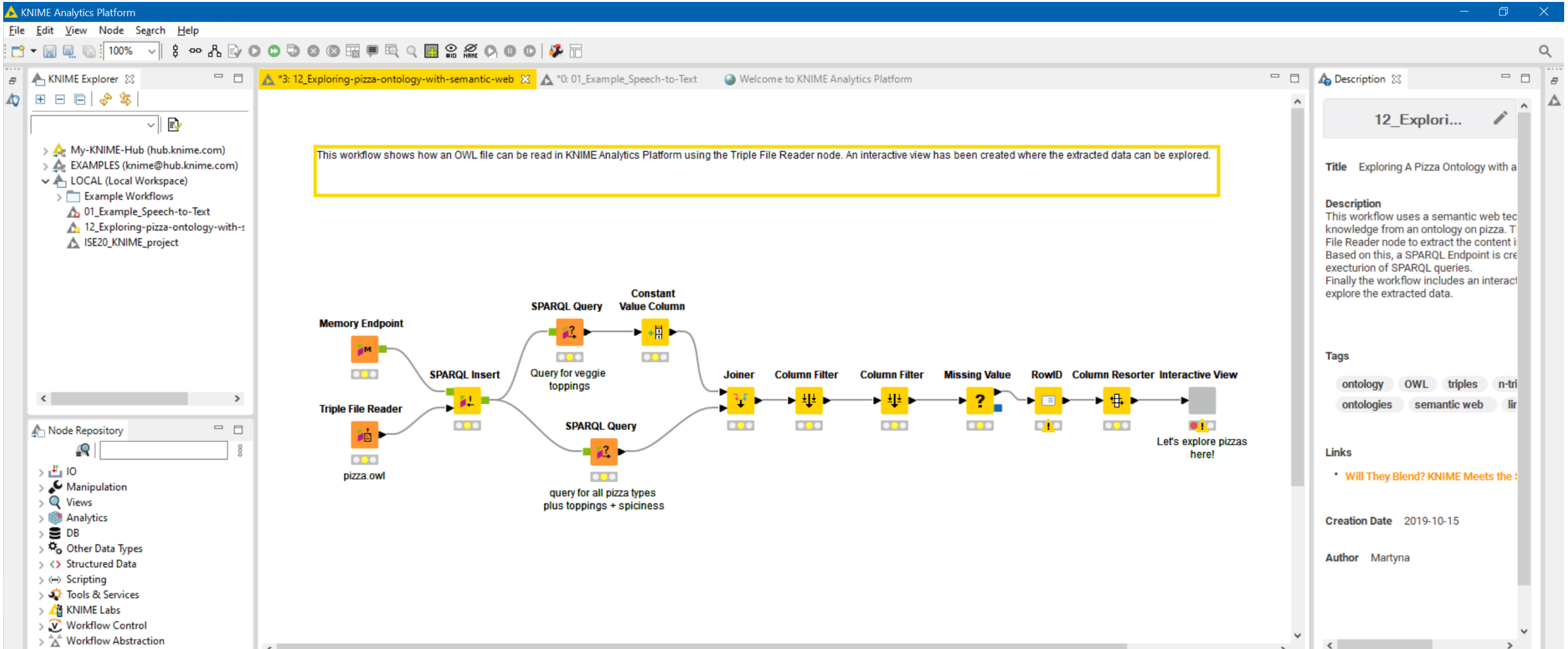
Time for every presentation 5-10 minutes

MORE EXAMPLES AND RESOURCES

UMEÅ UNIVERSITY

EXAMPLES

Demo: KNIME example of an ontology, querying using SPARQL




THANK YOU