Course 3
Semantic networks and ontologies

Tutore inteligent

- Un sistem care să ajute studentul în actul de învățare
  - capabil să-și organizeze cunoașterea din domeniu ca o ontologie
  - capabil să poarte o conversație asupra domeniului

Funcționare
Situația 1

- Am un text dintr-un domeniu medical => un program e capabil să extragă din el o reprezentare semantică
  - Și dacă același lucru îl exprim altfel?
  - Dar dacă aproximativ aceeași informație o găsesc în două tratate diferite?

Funcționare
Situația 2

- Există o reprezentare semantică a unui domeniu => un program mă ajută să-mi înșusesc acele noțiuni
  - pot pune întrebări => sistemul răspunde
  - sistemul întreabă => eu răspund
  - sistemul generează teste cu opțiuni de răspunsuri
  - sistemul e capabil să gestioneze corect imagini în Î/R și în teste

Funcționare
Situația 3

- Protégé e capabil să lucreze cu ontologii => o interfață îi va adăuga funcționalitatea de a “citi” un text
  - textul descrie o realitate în limbaj natural, care va fi copiată în ontologie

Funcționare
Situația 4

- MOODLE este un sistem capabil să asiste în procesul de învățare
  - o interfață îi va adăuga capacitatea de a dialoga cu studentul și de a genera teste

Knowledge representation
Semantic networks

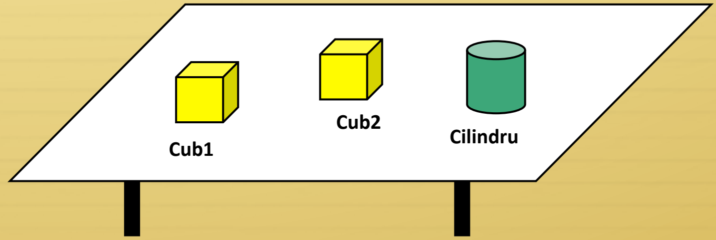
Intelligent agents appropriately react to the environment because they have their inner representation of the reality around them

Reţele semantice descriptive

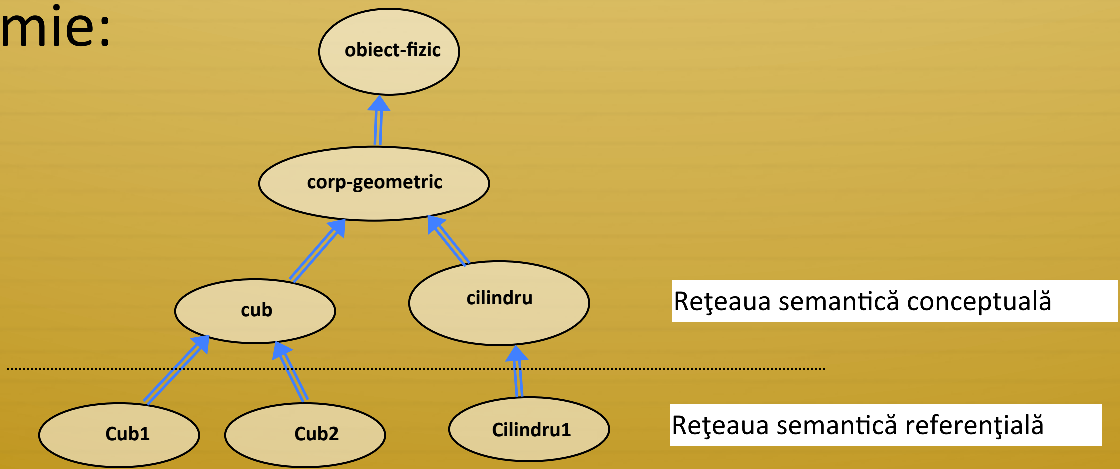
- adecvate reprezentării cunoaşterii statice
- se descriu:
  - entităţi, în ierarhia de la general spre specific
  - relaţii între entităţi
- două niveluri:
  - conceptual (intensiv): concepte (tipuri)
  - referenţial (extensiv): instanţe ale conceptelor

Rețele semantice descriptive

O lume obiectuală:



Taxonomie:



Rețelele semantice descriptive permit reprezentarea economică

- Proprietăţile:
  - explicite – la nivelul conceptual
  - implicite (moștenite) – la nivelul referențial
- Interogări:
  - care este închiderea tranzitivă a relaţiilor taxonomice ISA ale unui nod din reţea?
  - ce valoare este ataşată prin relaţia semantică *R* nodului *n*?
  - care este valoarea regăsită prin navigare în reţea în lungul lanţului de relaţii *R1* … *Rn*, plecând din nodul *n*?
  - care este calea de relaţii semantice ce se poate stabili între două noduri *n1* şi *n2*?

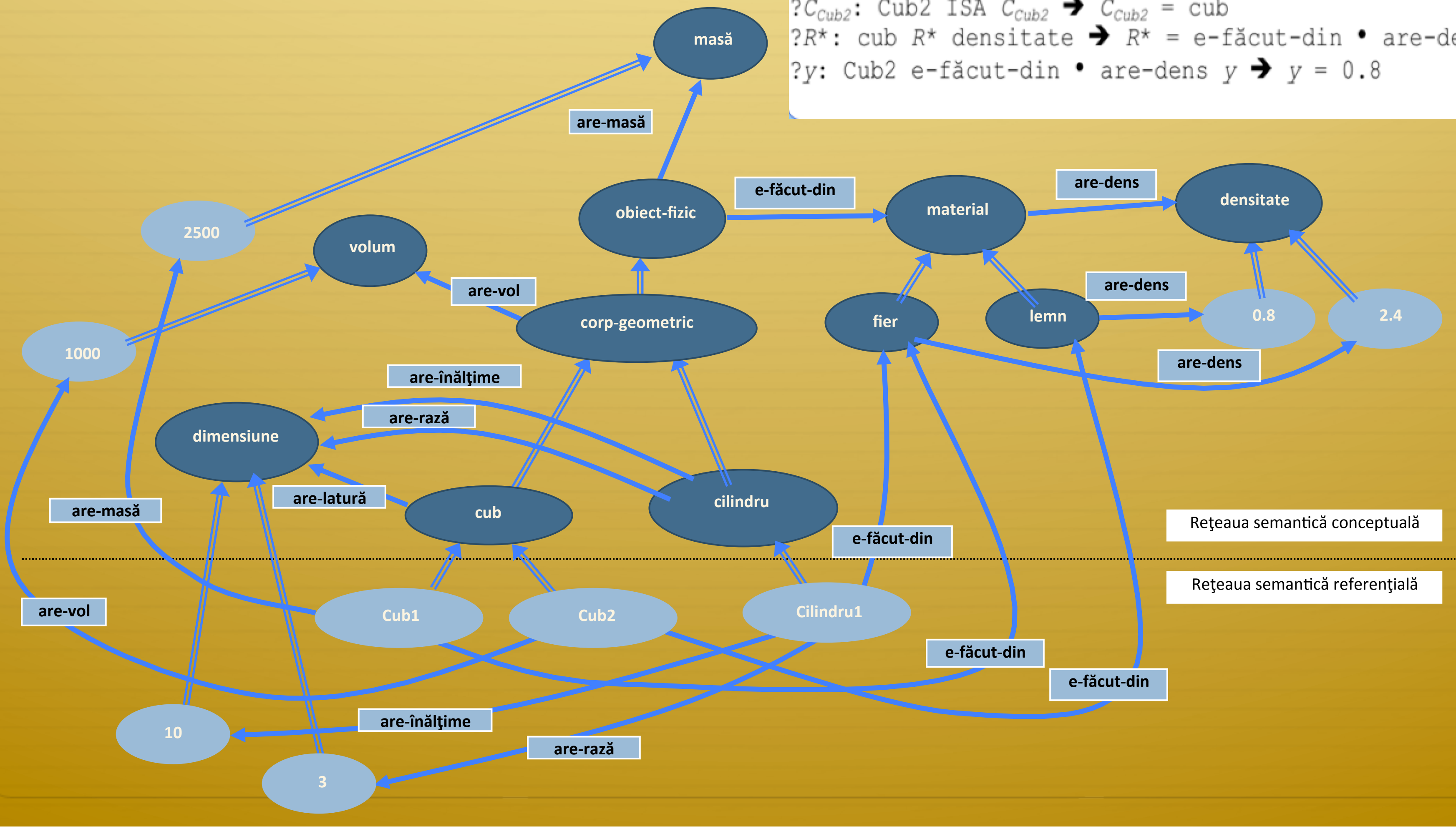
Interogări într-o rețea semantică

- IURES = Ințeleg Ușor Românește Eliminând Sintaxa
- IURES = I Understand and Reply Eliminating Syntax

Interogări într-o rețea semantică

Care este densitatea corpului `Cub2`*?*

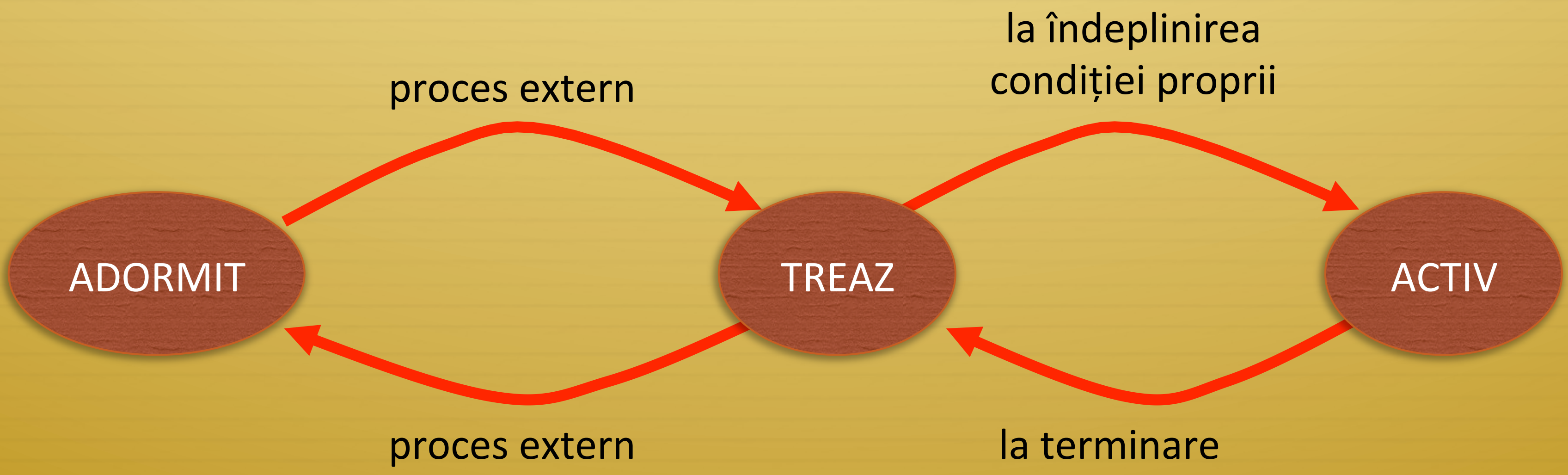
?C_{Cub2}: Cub2 ISA C_{Cub2} ➔ C_{Cub2} = cub
?R^*: cub R^* densitate ➔ R^* = e-făcut-din • are-dens
?y: Cub2 e-făcut-din • are-dens y ➔ y = 0.8



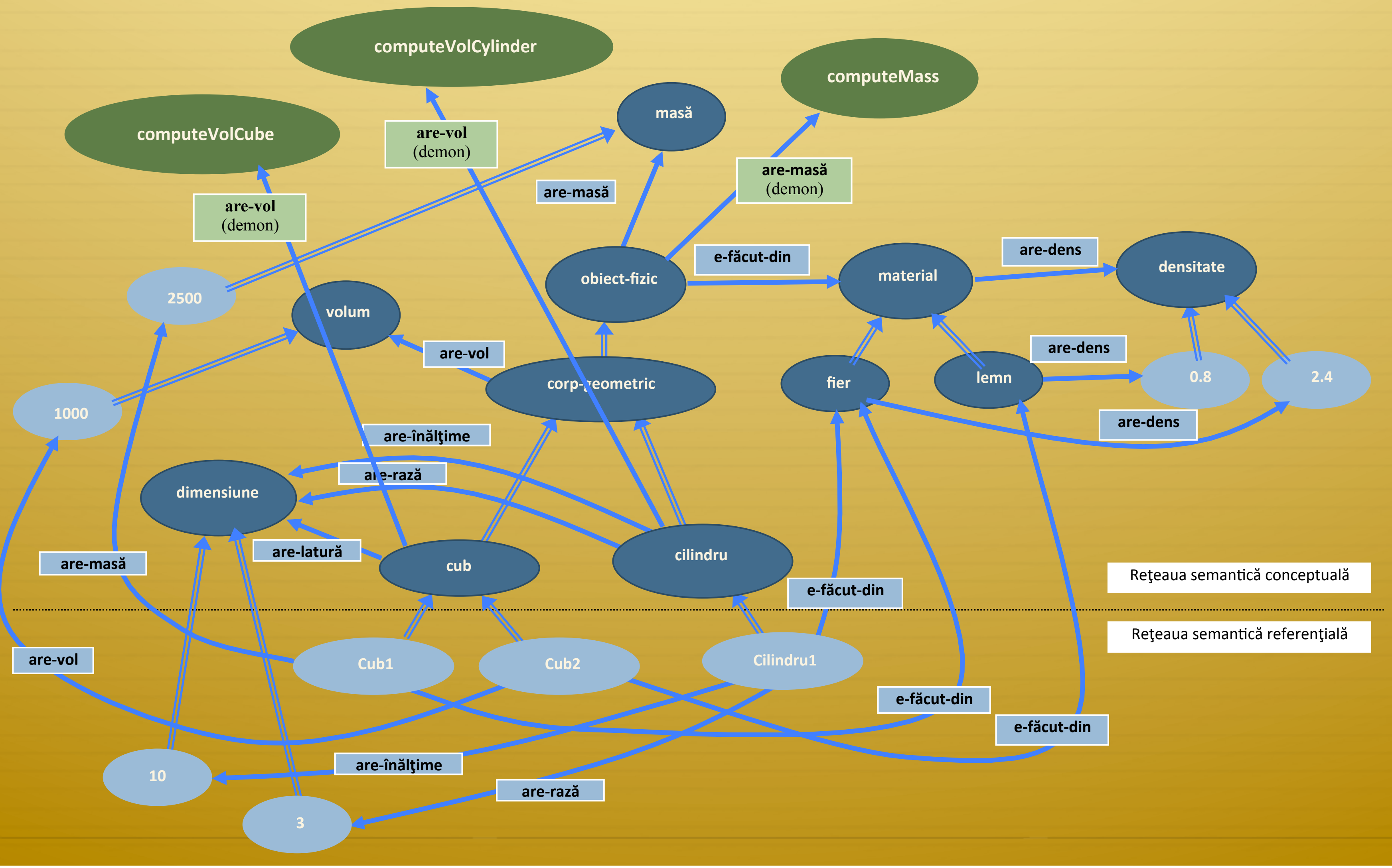
Demoni

- Proceduri care...
  - nu se apelează
  - se activează singure când anumite condiţii pe care ei sunt pregătiţi să le sesizeze sunt îndeplinite
- Stările unui demon:
  - **adormit**
  - **disponibil** (*idle*)
  - **activ**

Tranzițiile demonilor



Demoni într-o rețea semantică



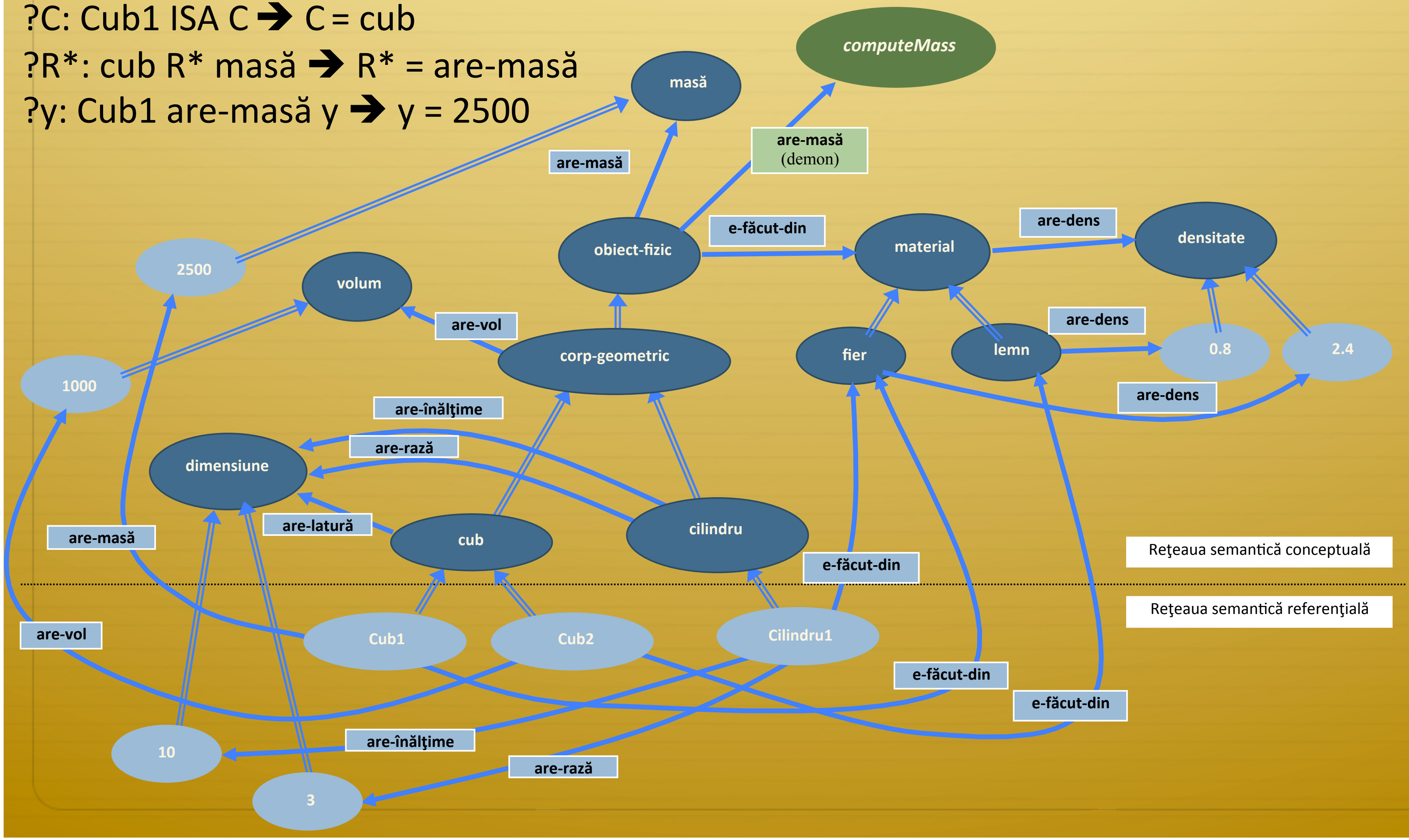
Demonul *ComputeMass*

$m = \rho * V$

```
procedure ComputeMass(x)
begin
; află densitatea lui x:
  ?Cx: x ISA Cx
  ?R1*: Cx R1* densitate
  ?y1: x R1* y1
; află volumul lui x:
  ?R2*: Cx R2* volum
  ?y2: x R2* y2
; calculează masa ca densitate * volum:
  return y1 * y2;
end
```

Activarea demonilor (demonul nu se activează)

Care este masa lui Cub1?
?C: Cub1 ISA C ➔ C = cub
?R*: cub R* masă ➔ R* = are-masă
?y: Cub1 are-masă y ➔ y = 2500



Demonul devine ACTIV

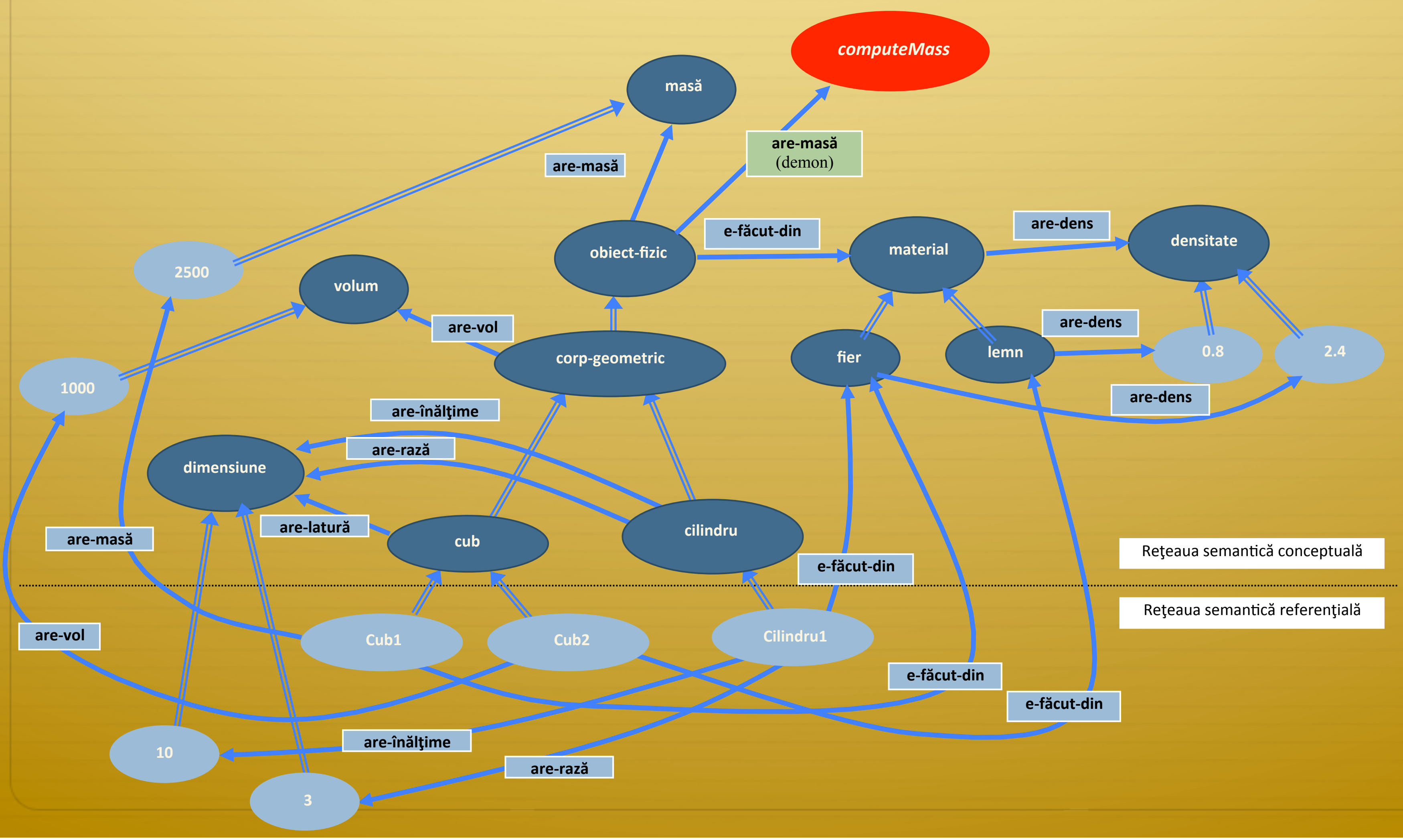
Care este masa lui Cub2?

?C_{Cub2}: Cub2 ISA C_{Cub2} ➔ C_{Cub2} = cub

?R*: cub R* masă ➔ R* = **are-masă**

?y: Cub2 **are-masă** y ➔ nil ➔

ACTIV demonul din vârful relației **are-masă**: *computeMass*



Demonul ComputeMass e activ!

cub2

$m=\rho*V$

```
procedure ComputeMass(x)
begin
; află densitatea lui x:
  ?Cx: x ISA Cx
  ?R1*: Cx R1* densitate
  ?y1: x R1* y1
; află volumul lui x:
  ?R2*: Cx R2* volum
  ?y2: x R2* y2
; calculează masa ca densitate * volum:
  return y1 * y2;
end
```

➔ C_x = cub

➔ R_1* = **e-făcut-din • are-dens**

➔ y_1 = cub2 **e-făcut-din • are-dens** = 0.8

➔ R_2* = **are-vol**

➔ y_2: cub2 **are-vol** y_2 ➔ y_2 = 1000

return 0.8 * 1000 = 800

Demoni într-o rețea semantică

Care este **masa cilindrului 1**?

?$C_{Cilindru1}$: Cilindru1 ISA $C_{Cilindru1}$ ➔

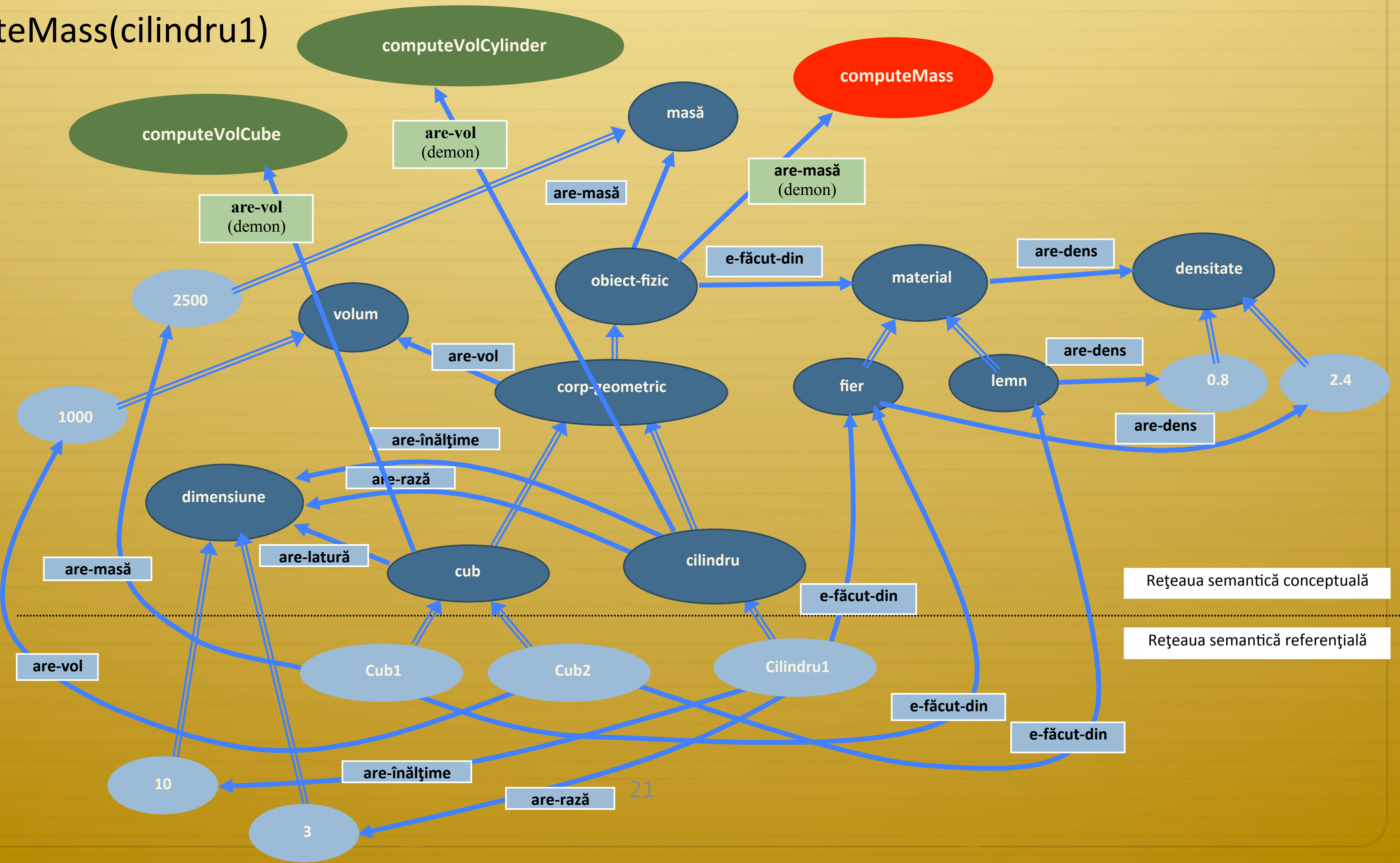
$C_{Cilindru1}$ = cilindru

?R*: cilindru R* masă ➔ R* = **are-masă**

?y: Cilindru1 **are-masă** y ➔ nil

➔ACTIV demonul

computeMass(cilindru1)



Demonul *ComputeMass* e activ!

Cilindru1

```
procedure ComputeMass(x)
begin
; află densitatea lui x:
  ?Cx: x ISA Cx
  ?R1*: Cx R1* densitate
  ?y1: x R1* y1
; află volumul lui x:
  ?R2*: Cx R2* volum
  ?y2: x R2* y2
; calculează masa ca densitate * volum:
  return y1 * y2;
end
```

m=ρ*V

➔ C_x = cilindru

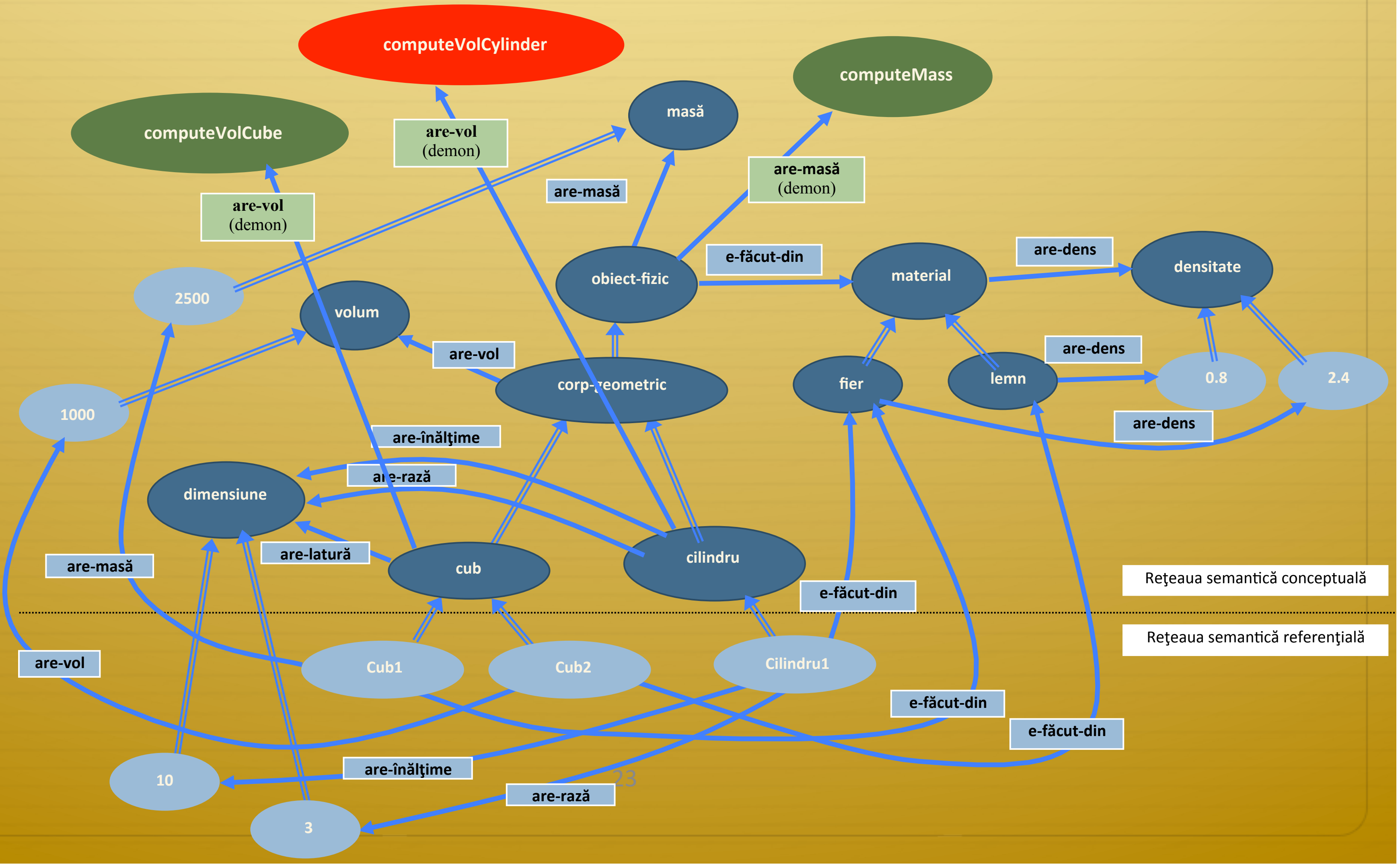
➔ R_1* = **e-făcut-din • are-dens**

➔ y_1 = Cilindru1 **e-făcut-din • are-dens** = 2.4

➔ R_2*: Cilindru R_2* **volum** ➔ R_2* = **are-vol**

➔ y_2: Cilindru1 **are-vol** y_2 ➔ nil ➔...

Demonul *computeVolCylinder* devine ACTIV



Demonul *ComputeVolCylinder* e activ!

Cilindru1

$V = \pi * r^2 * H$

```
procedure ComputeVolCylinder(x)
begin
; află raza bazei lui x:
  ?r: x are-rază r                  → 3
; află înălţimea lui x:
  ?h: x are-înălţime h              → 10
; calculează volumul:
  return 3.14 * r * r * h; return 3.14 * 3 * 3 * 10 = 282.6
end
```

Acum *ComputeMass* poate termina calculul!

Cilindru1

$m=\rho*V$

```
procedure ComputeMass(x)
begin
; află densitatea lui x:
  ?C_x: x ISA C_x
  ?R_1*: C_x R_1* densitate
  ?y_1: x R_1* y_1
; află volumul lui x:
  ?R_2*: C_x R_2* volum
  ?y_2: x R_2* y_2
; calculează masa ca densitate * volum:
  return y_1 * y_2;
end
```

→ C_x = cilindru

→ R_1* = **e-făcut-din • are-dens**

→ y_1 = Cilindru1 **e-făcut-din • are-dens** = 2.4

→ R_2*: Cilindru R_2* **volum** → R_2* = **are-vol**

→ y_2: Cilindru1 **are-vol** y_2 → nil → … . 282.6

return 2.4 * 282.6 = 678.24

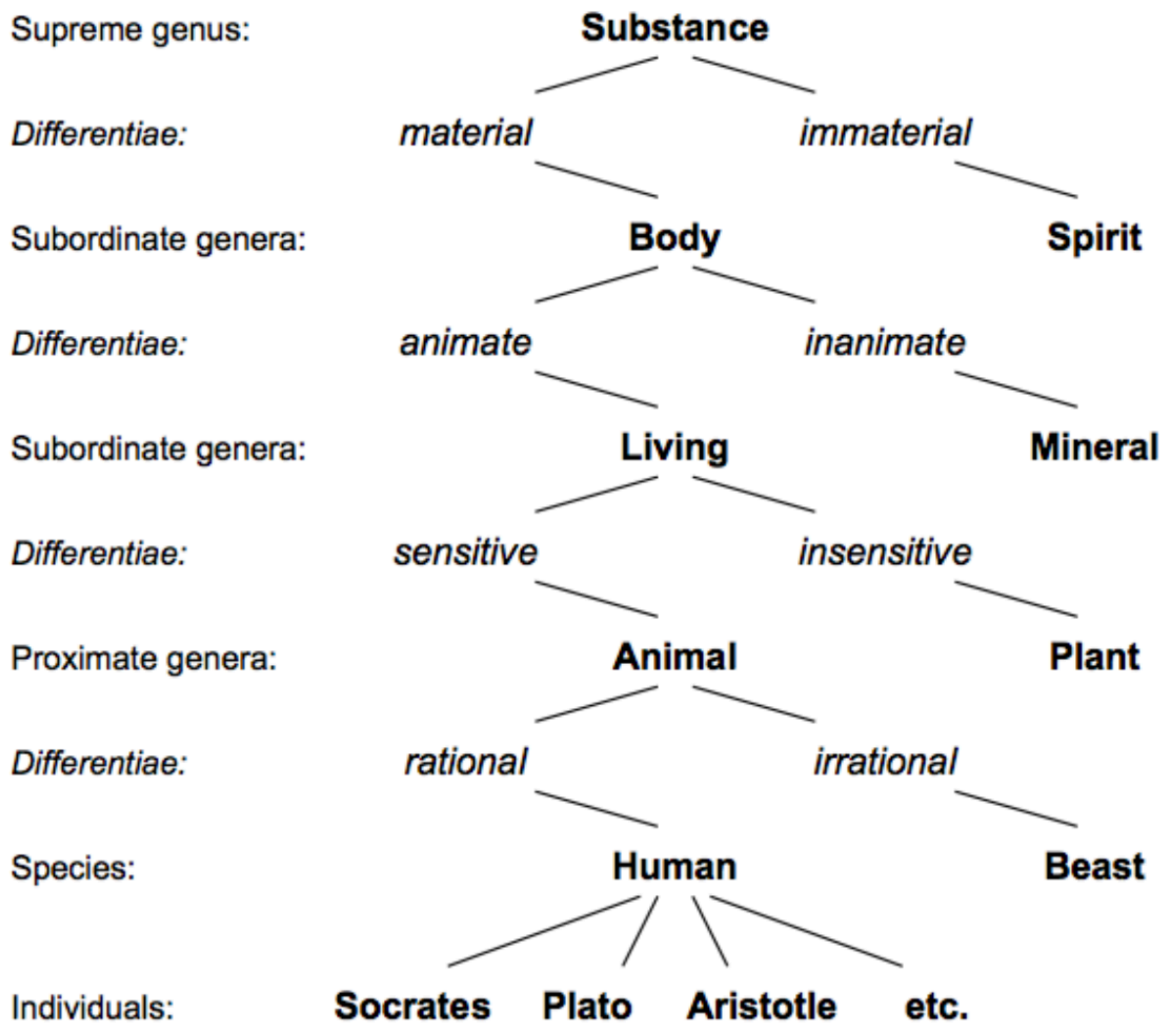
Ontologies

- Rich conceptual schemas:
  - give formally defined meanings to the terms used in annotations, transforming them into semantic annotations
  - "Ontologies serve as metadata schemas, providing a controlled vocabulary of concepts, each with explicitly defined and machine-processable semantics. By defining shared and common domain theories, ontologies help people and machines to communicate concisely—supporting semantics exchange, not just syntax. Hence, the Semantic Web's success and proliferation depends on quickly and cheaply constructing domain-specific ontologies."

From: A. Maedche and S. Staab. Ontology learning for the semantic web. IEEE Intelligent Systems, 16(2):72–79, 2001.

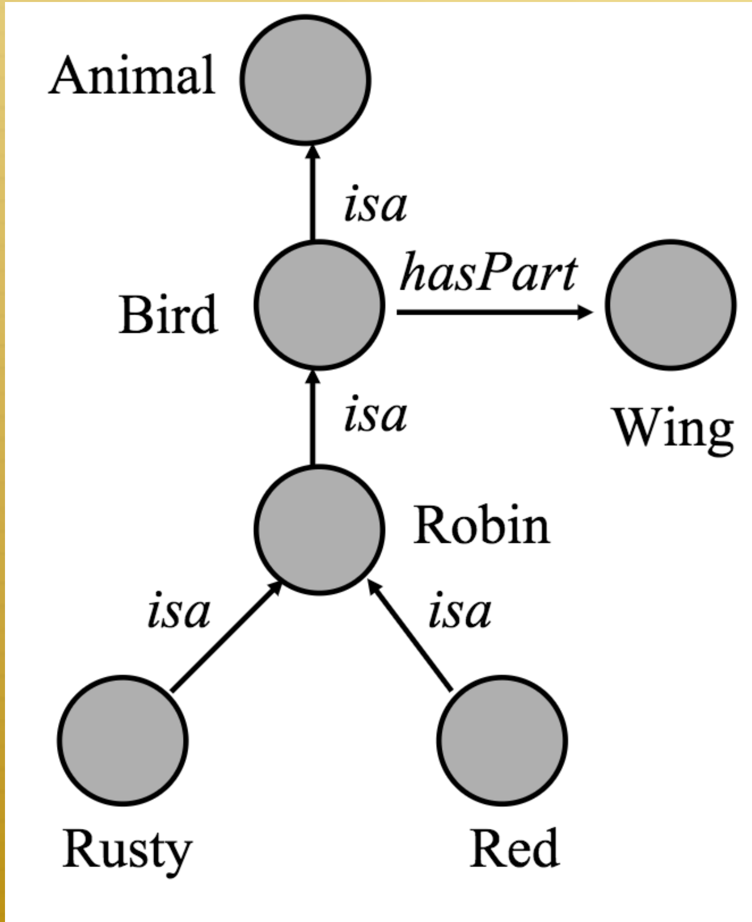
Ontology

- The study of ontology
  - traced back to the work of Plato and Aristotle
  - development of hierarchical categorisations of different kinds of entities and their distinguishing features:
    - the well known “tree of Porphyry” identifies animals and plants as sub-categories of living things distinguished by animals being sensitive, and plants being insensitive



Tree of Porphyry.

Rețele semantice: ce e greșit aici?...



prihor

Plasarea conceptului *bone* într-o ontologie

Class Hierarchy

Thing
- + anatomical entity
  - + material anatomical entity
    - + anatomical structure
      - + multi-tissue structure
        - + bone
          - + endochondral bone
          - - membrane bone

Superclasses & Asserted Axioms

- has_part some osteoblast
- part_of some skeletal system
- bone

Class:skeletal system Definition: The rigid support system for the body.

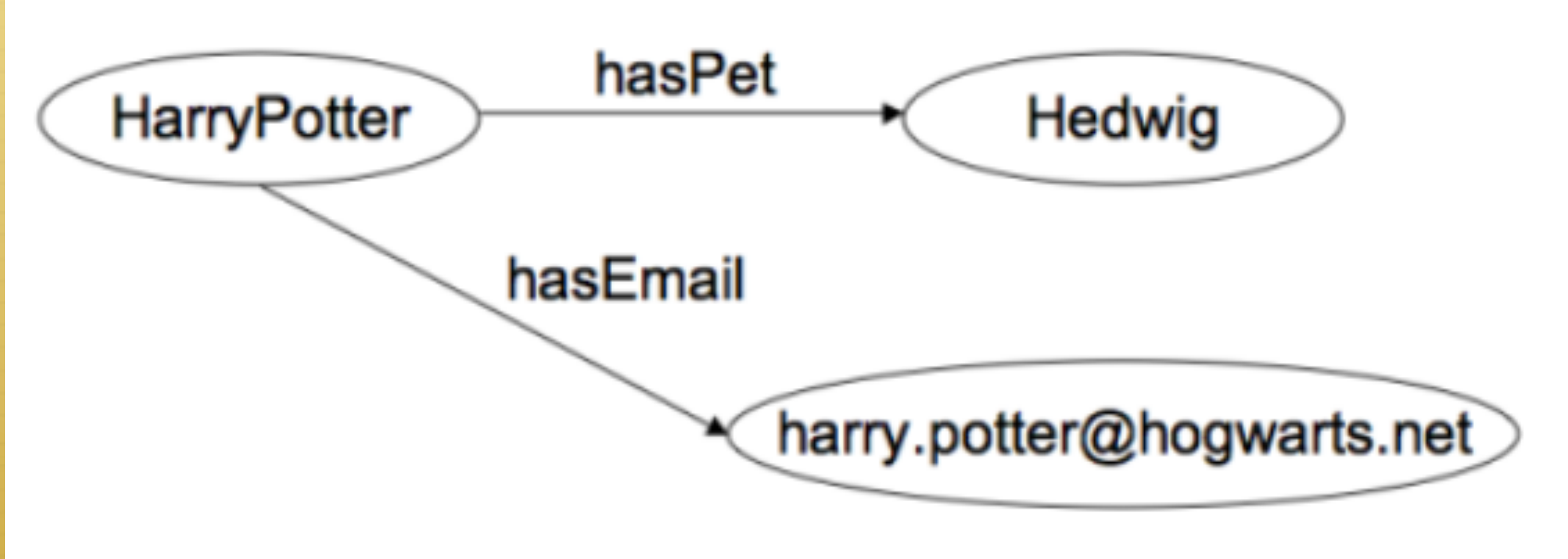
http://www.ontobee.org/ontology/AEO?iri=http://purl.obolibrary.org/obo/AEO_0000085

OWL – un limbaj de descriere a ontologiilor

- The World Wide Web Consortium (W3C) set up a standardisation working group to develop a standard for a web ontology language => OWL ontology language standard

- OWL is based on Description Logics (DLs): a family of logic-based knowledge representation formalisms that are descendants of Semantic Networks and KL-ONE, but that have a formal semantics based on first order logic

Basic things in OWL



Concepts (classes)
Instances (individuals)
Properties (roles)

Semantică vs sintaxă

- Sintaxa => descrie regularități de formă ale unui limbaj
  - o exprimare poate fi ambiguă => mai multe interpretări
  - mai multe exprimări => aceeași reprezentare semantică
- Semantica => descrie înțelesul exprimărilor, semnificația lor
  - reprezentările semantice nu au voie să dea loc la interpretări multiple
  - trebuie să fie independente de limbajul sintactic

Now-a-days web

- Web content consists mainly of distributed hypertext and hypermedia accessed via a combination of keyword based search and link navigation
- Its simplicity contributed to its large dissemination and use
- Web pages use images, often including active links, to present information, and even when content is annotated, the annotations typically take the form of natural language strings and tags
- But, the search engines are incapable to answer complex queries

Semantic Web

- The goal of semantic web research: to allow the vast range of web-accessible information and services to be more effectively exploited by both humans and automated tools.
- Exploitation of the vast web
  - through RDF and OWL: standard formats for the sharing and integration of data and knowledge—the latter in the form of rich conceptual schemas called **ontologies**

Examples of queries that cannot be answered by the actual web search engines

- The list of presidents of EU countries
  - List of EU countries:
    - https://europa.eu/european-union/about-eu/countries/member-countries_en
  - List of presidents of countries:
    - https://en.wikipedia.org/wiki/List_of_current_heads_of_state_and_government)

Examples of queries…

- German writers contemporary with Beethoven
  - a book: **Music and Literature in German Romanticism**, by Siobhán Donovan and Robin Elliott
    - abstract, presented at http://www.jstor.org/stable/10.7722/j.ctt81t1f
  - an Wikipedia article: Beethoven and his contemporaries
    - but mainly composers, at https://en.wikipedia.org/wiki/Beethoven_and_his_contemporaries

German writers contemporary with Beethoven

- One possible solution:
  - Beethoven has lived between T1 and T2
  - German writters borned between T1-20 and T2-20

Examples of queries…

- Classical example of a semantic web application:
  - an automated travel agent that, given various constraints and preferences, would offer the user suitable travel or vacation suggestions
  - key feature of such a “software agent”: it would not simply exploit a predetermined set of information sources, but would search the web for relevant information in much the same way that a human user might do when planning a vacation

Key idea behind the semantic web

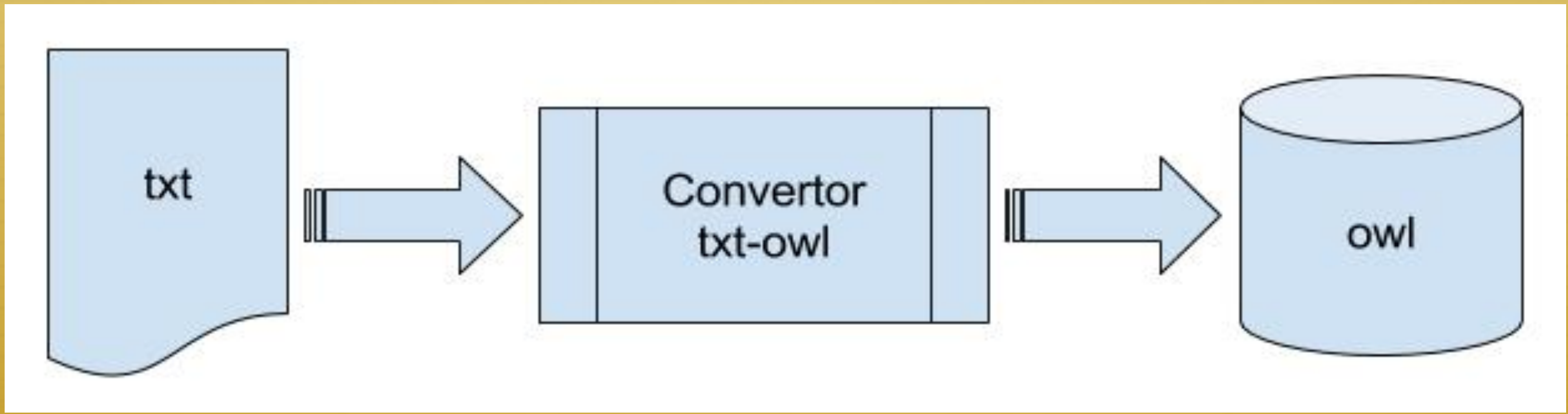
- Address the problem of offering automated reasoning by giving the machine accessible semantics to annotations.
- Achieved through **ontologies**
- Areas:
  - knowledge representation and reasoning, databases, computational linguistics, computer vision, agent systems

Module

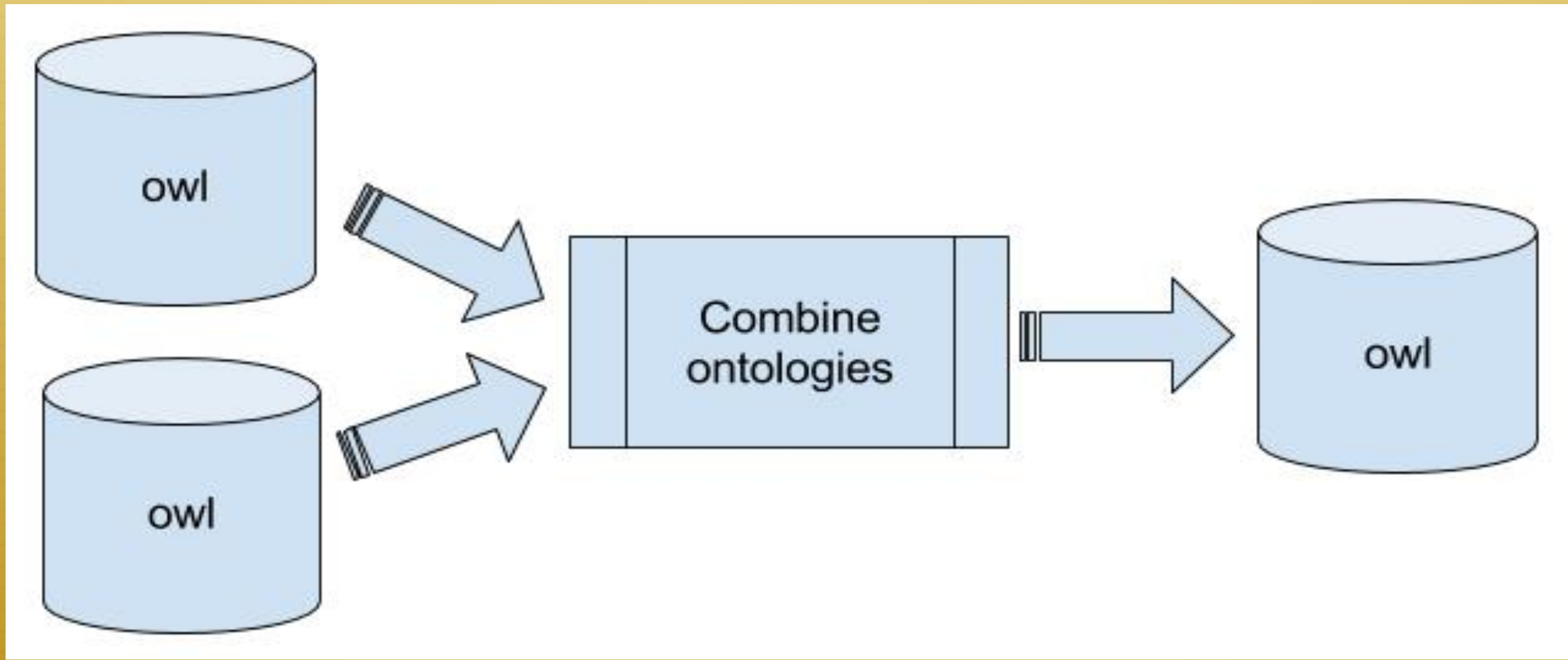
1. Achiziția unei ontologii din text
2. Combinare de ontologii (cu validare etc.)
3. Interfață text-Protégé pentru vizualizare/editare
4. Generare de teste plecând de la o ontologie
5. Chatbot capabil să poarte un dialog cu studentul (sesiune de Î/R cu inițiativă mixtă)
6. Interfața cu MOODLE pentru introducerea de note individualizate, exemple din manual, vizualizări de părți din ontologie
7. Valorificarea imaginilor în teste

=> Testare, date de test, convenții/standarde de interfațare între module

Achiziția ontologiei din text



Combinarea de ontologii

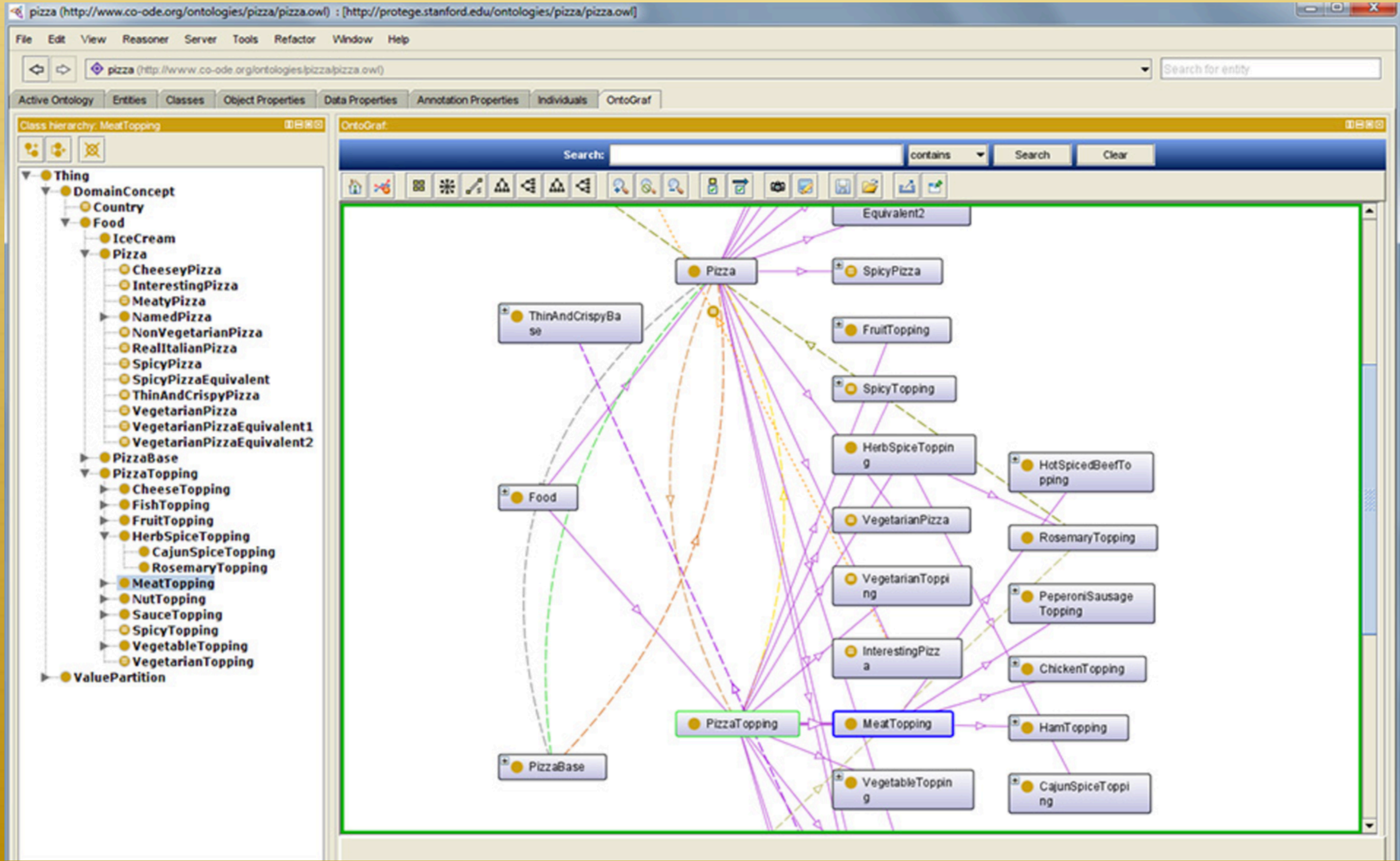


Protégé

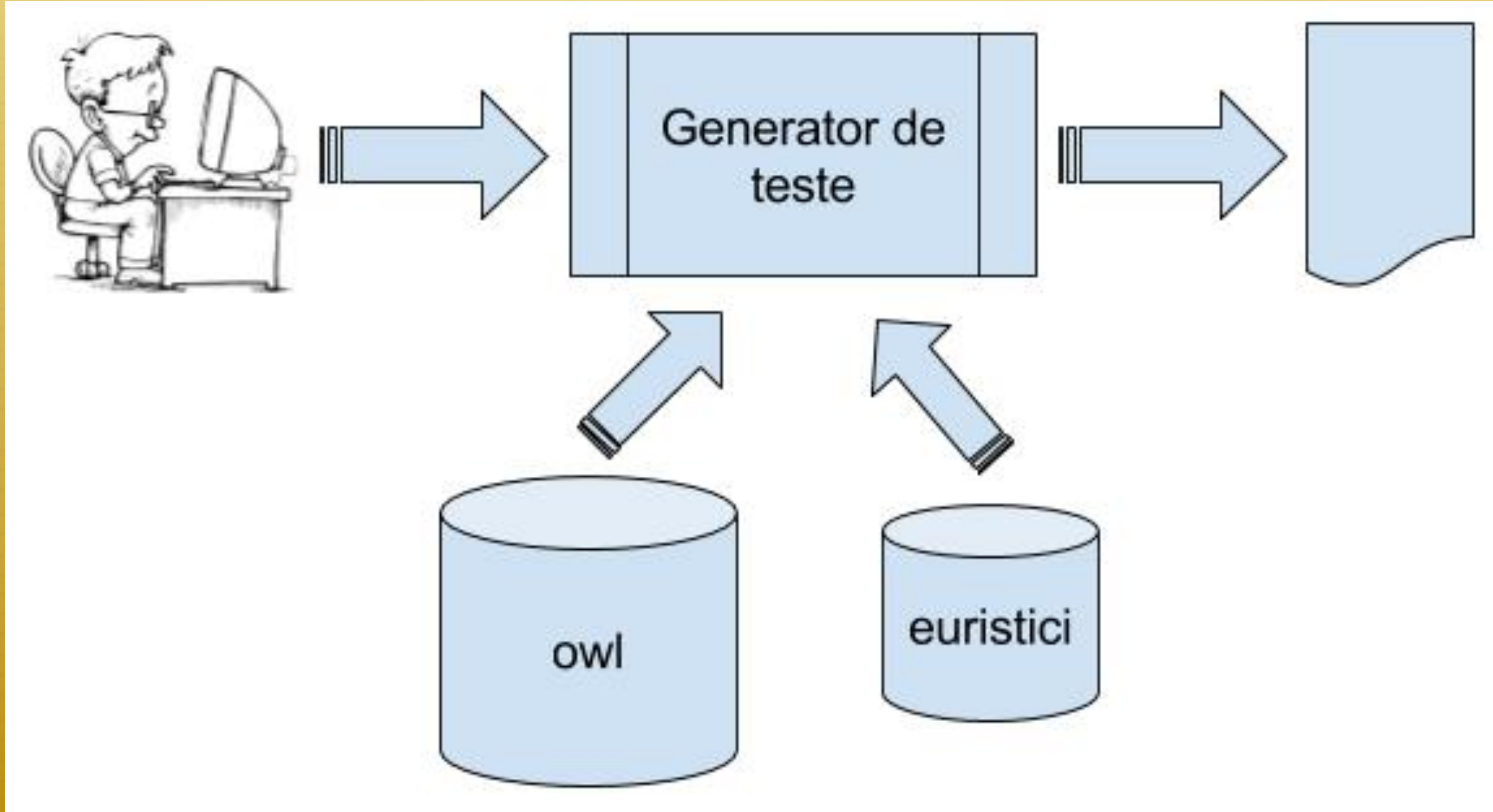
- Mediu online de construire și editare de ontologii și creare de sisteme inteligente
  - *Protégé Desktop supports creation and editing of one or more ontologies in a single workspace via a completely customizable user interface. Visualization tools allow for interactive navigation of ontology relationships. Advanced explanation support aids in tracking down inconsistencies. Refactor operations available including ontology merging, moving axioms between ontologies, rename of multiple entities, and more.*

https://protege.stanford.edu/

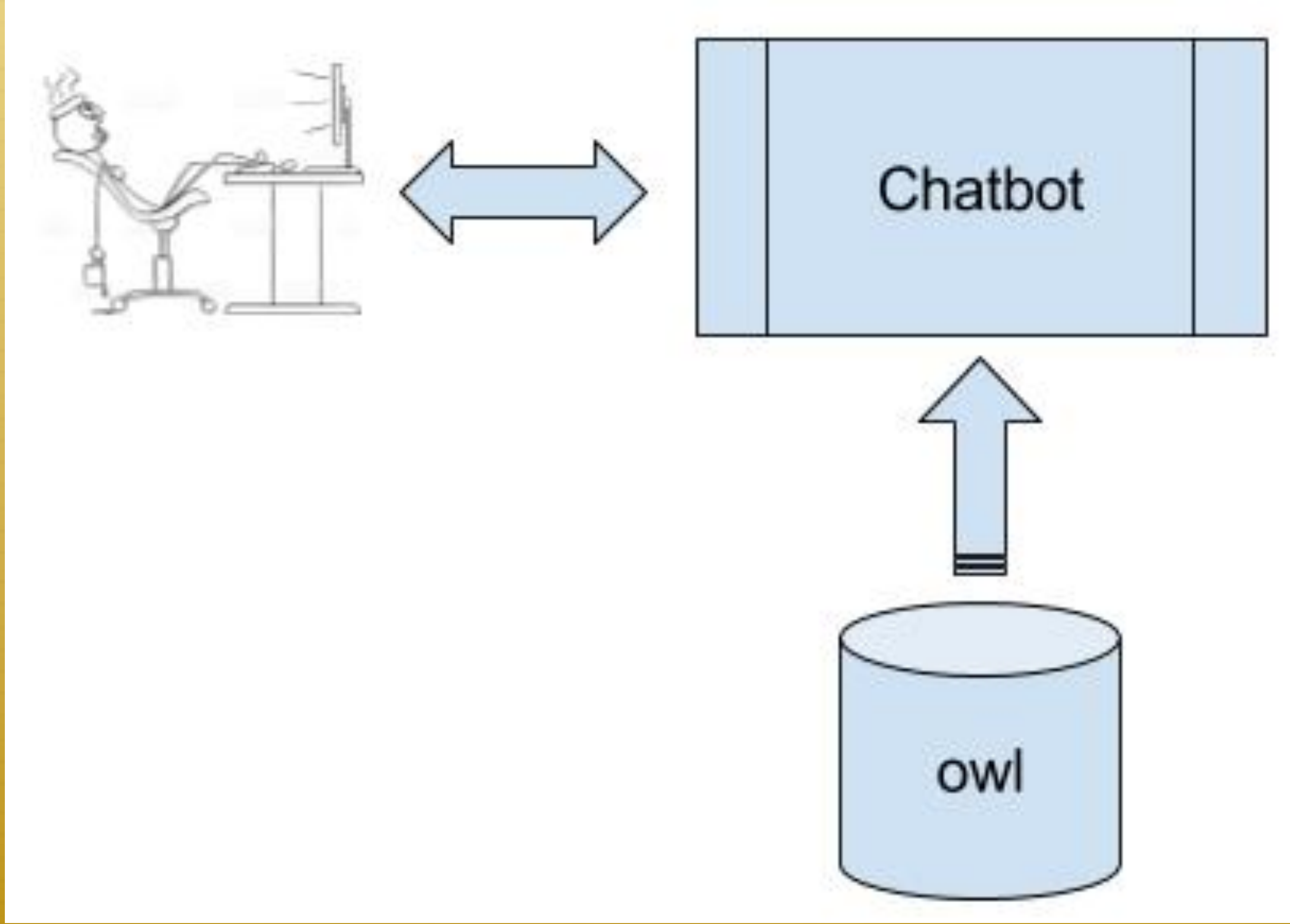
Protégé


pizza (http://www.co-ode.org/ontologies/pizza/pizza.owl) : [http://protege.stanford.edu/ontologies/pizza/pizza.owl]
File Edit View Reasoner Server Tools Refactor Window Help
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OntoGraf
Class hierarchy: MeatTopping
Thing
DomainConcept
Country
Food
IceCream
Pizza
CheeseyPizza
InterestingPizza
MeatyPizza
NamedPizza
NonVegetarianPizza
RealItalianPizza
SpicyPizza
SpicyPizzaEquivalent
ThinAndCrispyPizza
VegetarianPizza
VegetarianPizzaEquivalent1
VegetarianPizzaEquivalent2
PizzaBase
PizzaTopping
CheeseTopping
FishTopping
FruitTopping
HerbSpiceTopping
CajunSpiceTopping
RosemaryTopping
MeatTopping
NutTopping
SauceTopping
SpicyTopping
VegetableTopping
VegetarianTopping
ValuePartition
OntoGraf
Search:
contains
Search
Clear
Pizza
SpicyPizza
ThinAndCrispyBase
FruitTopping
SpicyTopping
HerbSpiceTopping
HotSpicedBeefTopping
Food
VegetarianPizza
RosemaryTopping
VegetarianTopping
PeperoniSausageTopping
InterestingPizza
ChickenTopping
PizzaTopping
MeatTopping
HamTopping
PizzaBase
VegetableTopping
CajunSpiceTopping

Generare de teste plecând de la o ontologie



Chatbot

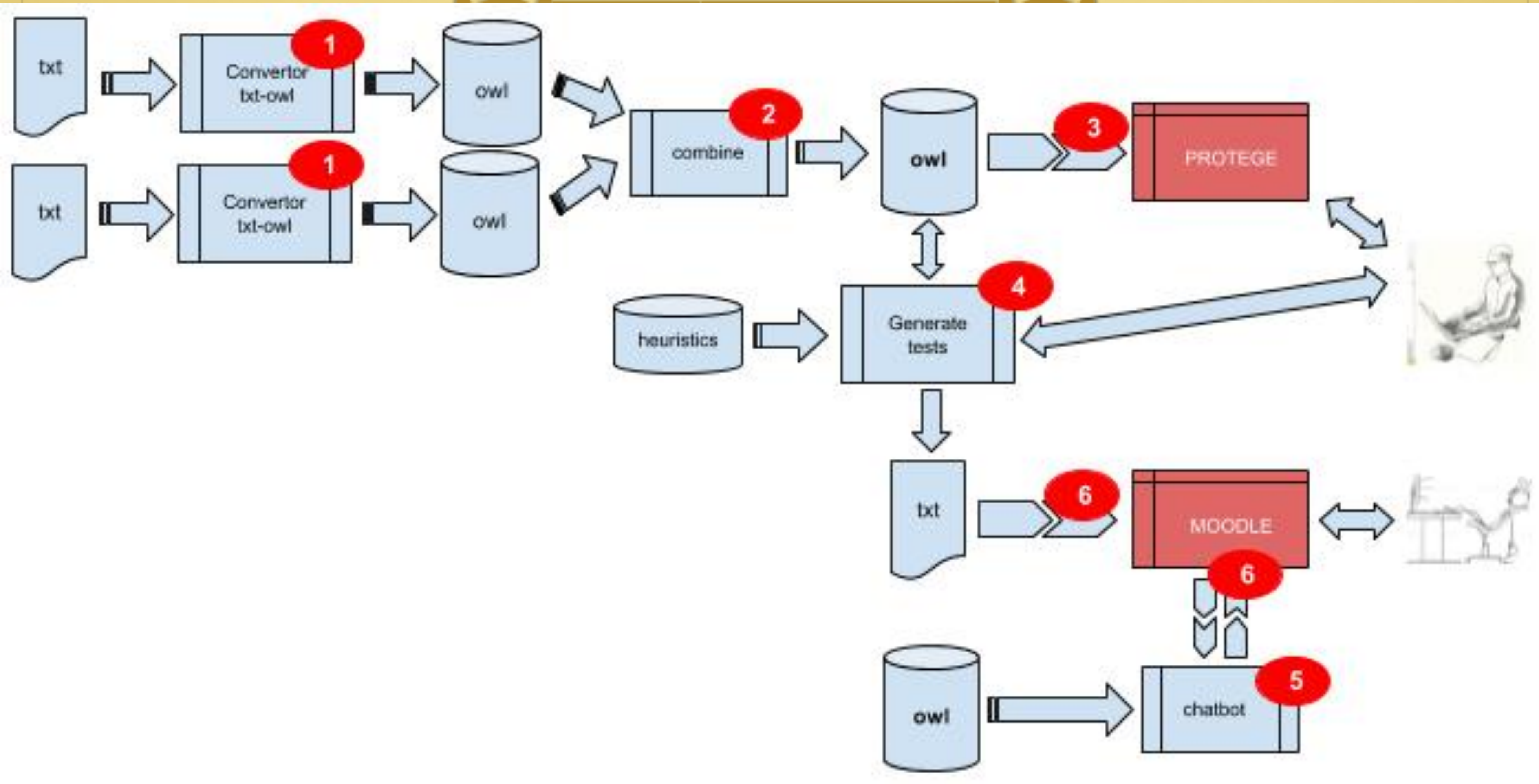


MOODLE

- Moodle (*modular object-oriented dynamic learning environment*) allows for extending and tailoring learning environments using community sourced plugins (Wikipedia)
- Mediu educațional open-source pentru crearea de cursuri online și educație la distanță

https://moodle.org/

O arhitectură



Documentații

- WebProtégé User Guide:
  - https://protegewiki.stanford.edu/wiki/
- Medicină:
  - Boala Alzheimer: http://www.medtorrents.com/load/neurologie/curs_boala_alzheimer/25-1-0-991

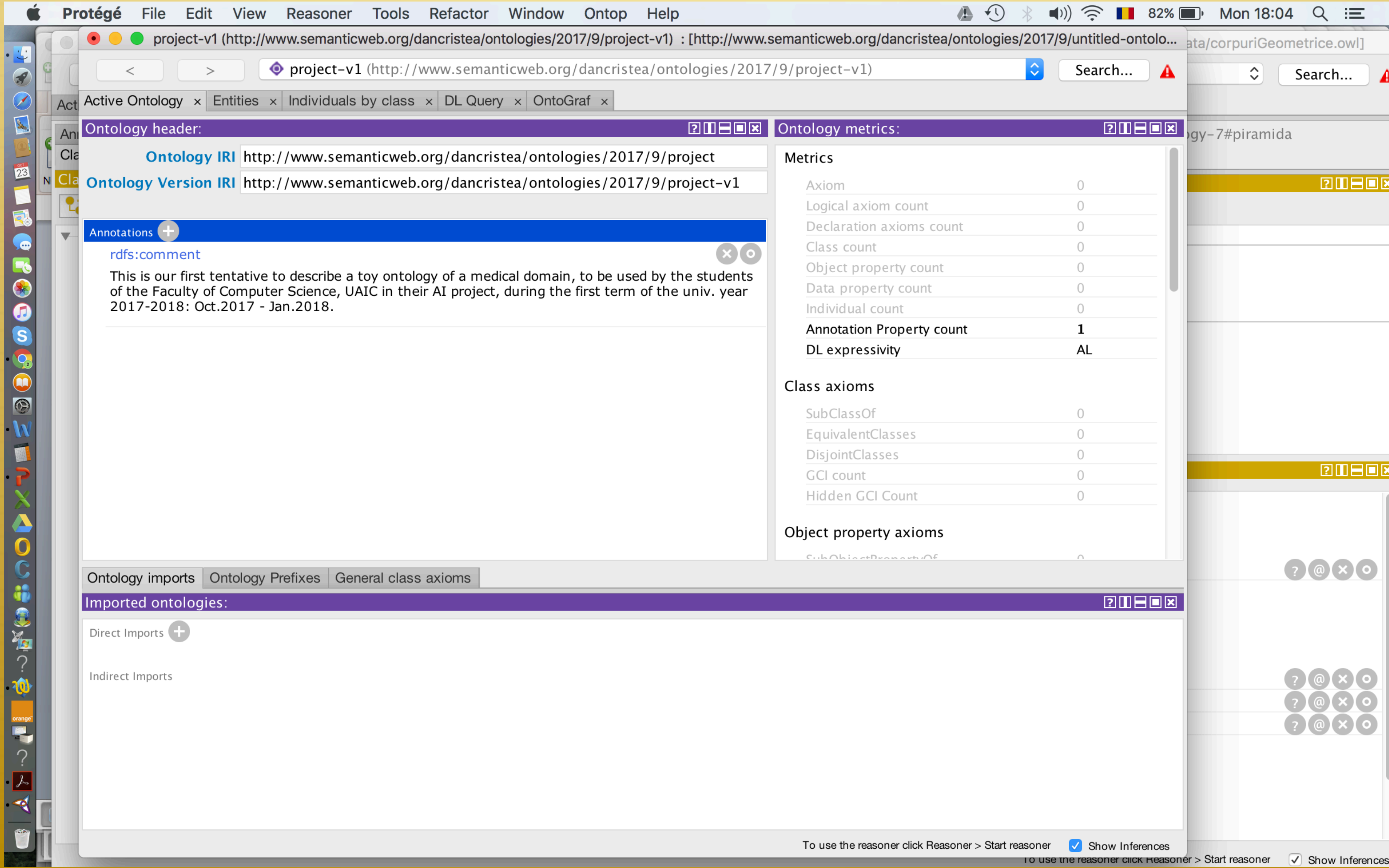
Ontologies

- An ontology describes the concepts of a domain and the relations that hold among them.
- An ontology includes:
  - concepts (classes – in Protégé's language)
  - instances of concepts (individuals)
  - relations (proprieties) of concepts and instances
- In an ontology navigates a program: the *reasoner*

Protégé and OWL

- https://protege.stanford.edu/short-courses.php

The Protégé interface



Interfața Protégé

- IRI – *Internationalized Resource Identifier*, definit de către *Internet Engineering Task Force* (IETF) în 2005 ca un nou stardard de internet, extinzând schema lui *Uniform Resource Identifier* (URI)
    - IRI extinde URI pentru că folosește *Universal Character Set,* în timp ce URI erau limitate la coduri ASCII care pot codifica mult mai puține caractere

Ontologia Pizza

- http://protege.stanford.edu/ontologies/pizza/pizza.owl

The classes

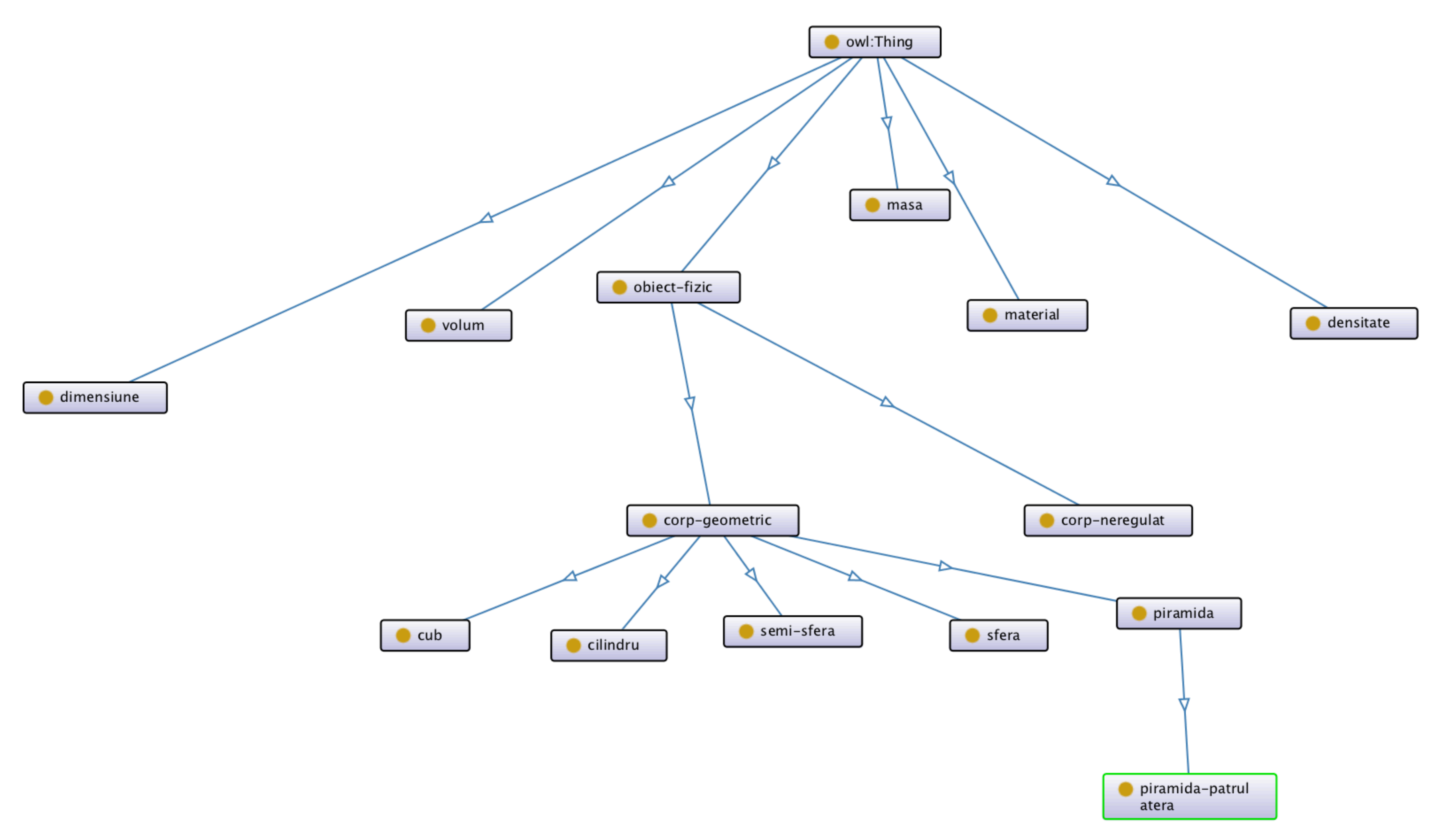
- A class (concept): a set that includes individuals
- An individual may belong to more classes
- A superclass-subclass hierarchy is called a taxonomy

```
<Declaration>
   <Class IRI="#corp-geometric"/>
</Declaration>
```

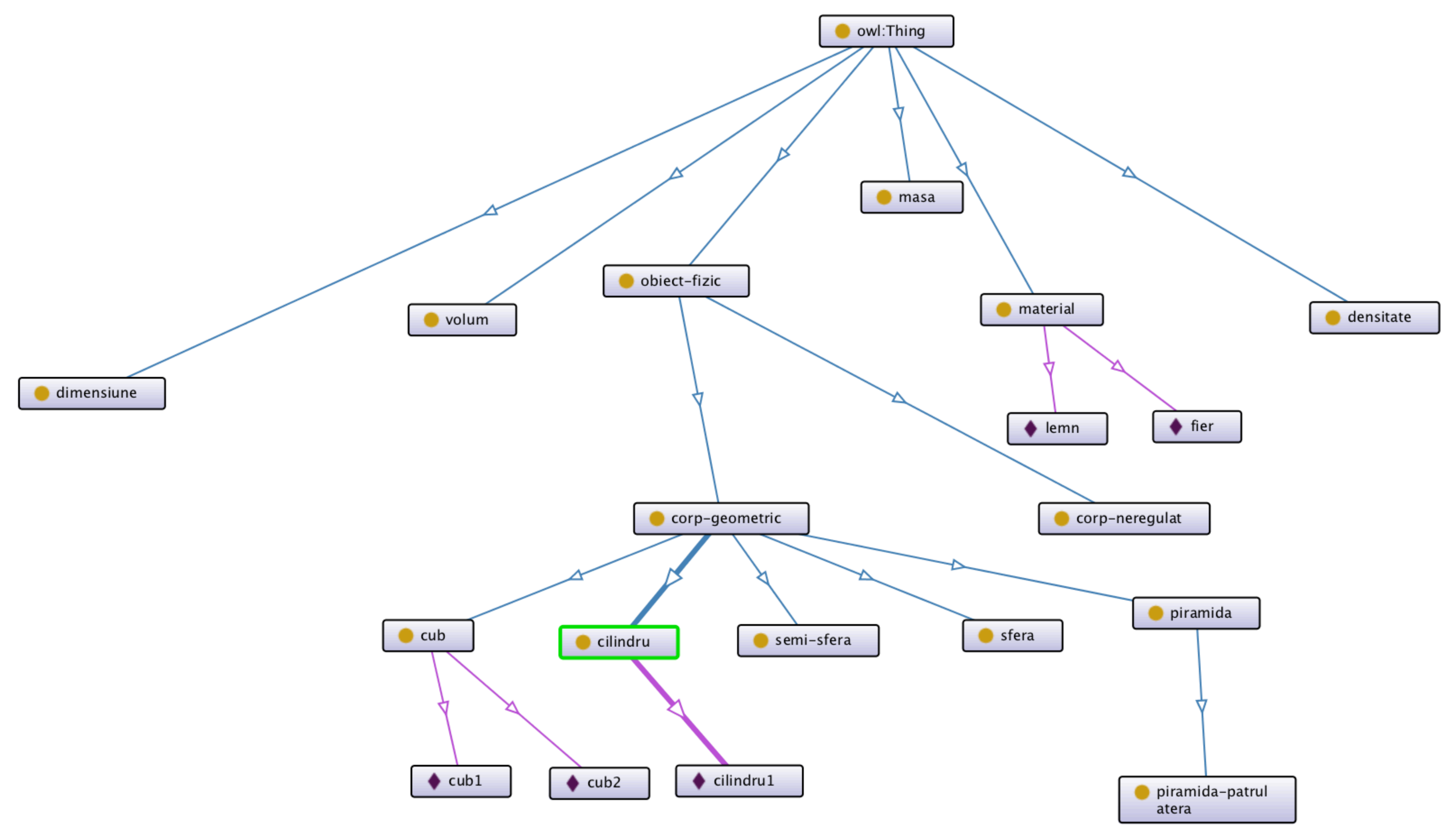
Individuals

- The unique name convention does not work in Protégé => the same entity may have more names
- The identity must be stated explicitly

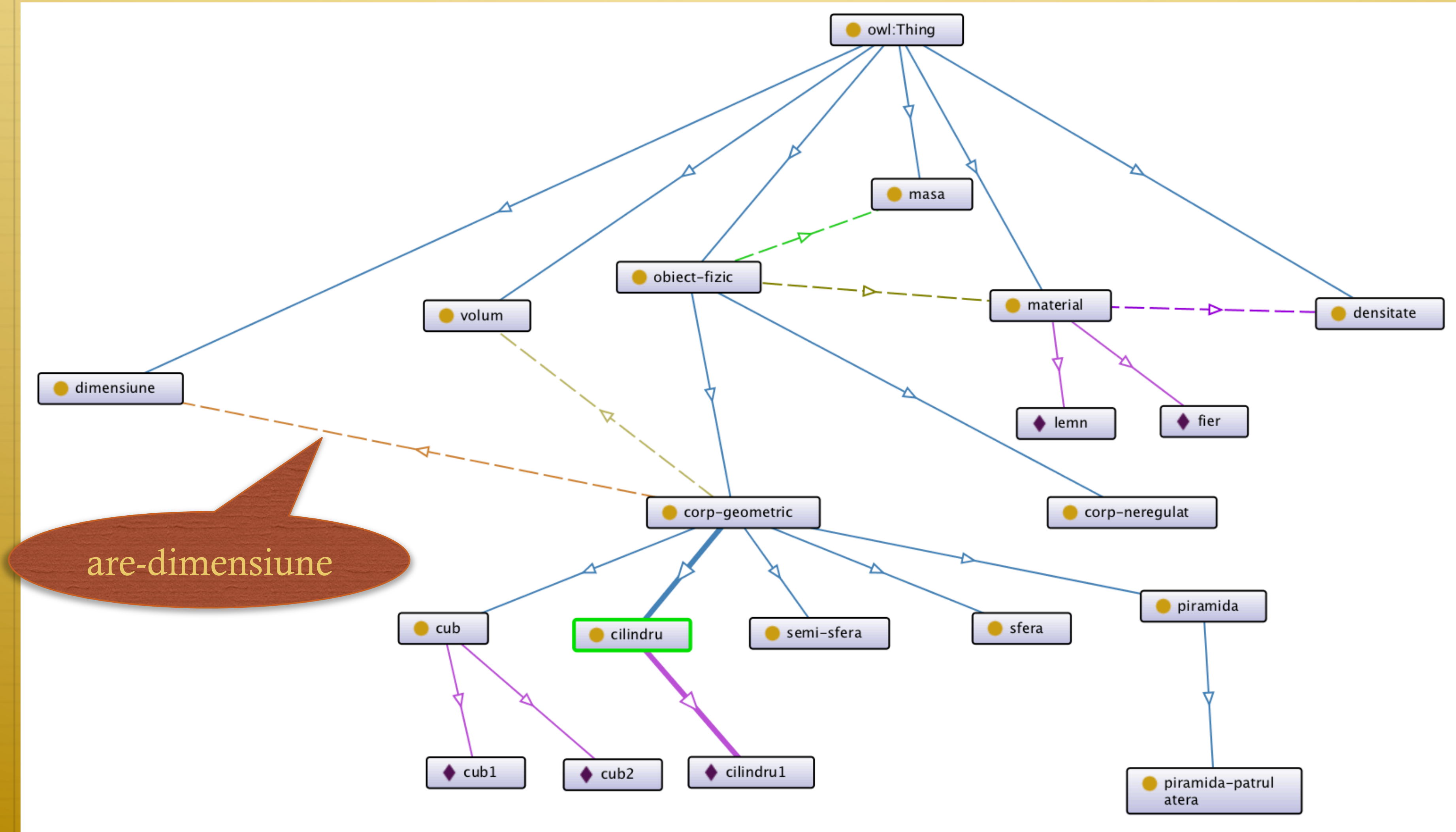
Classes and subclasses – an example



Classes and individuals

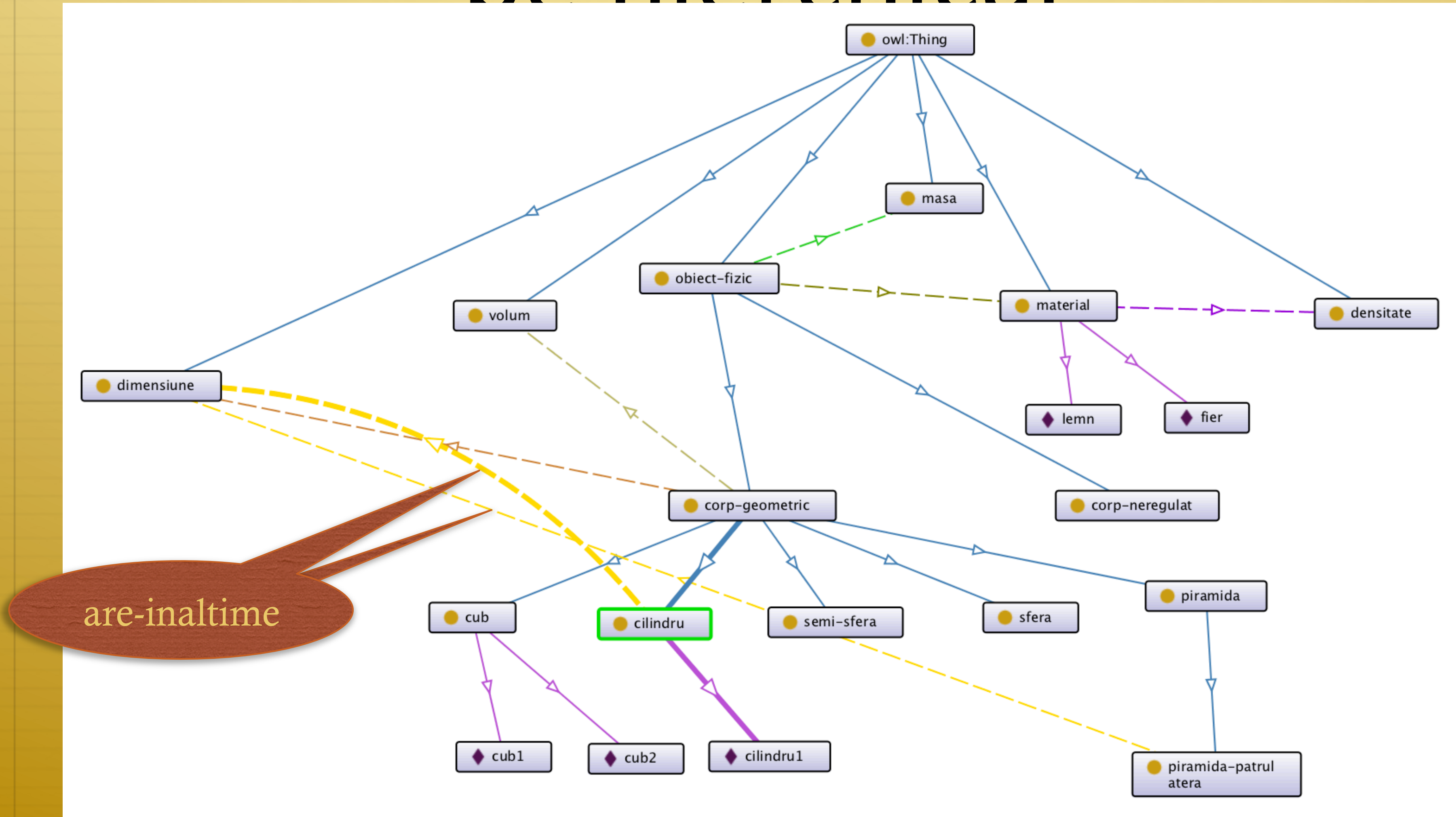


Properties

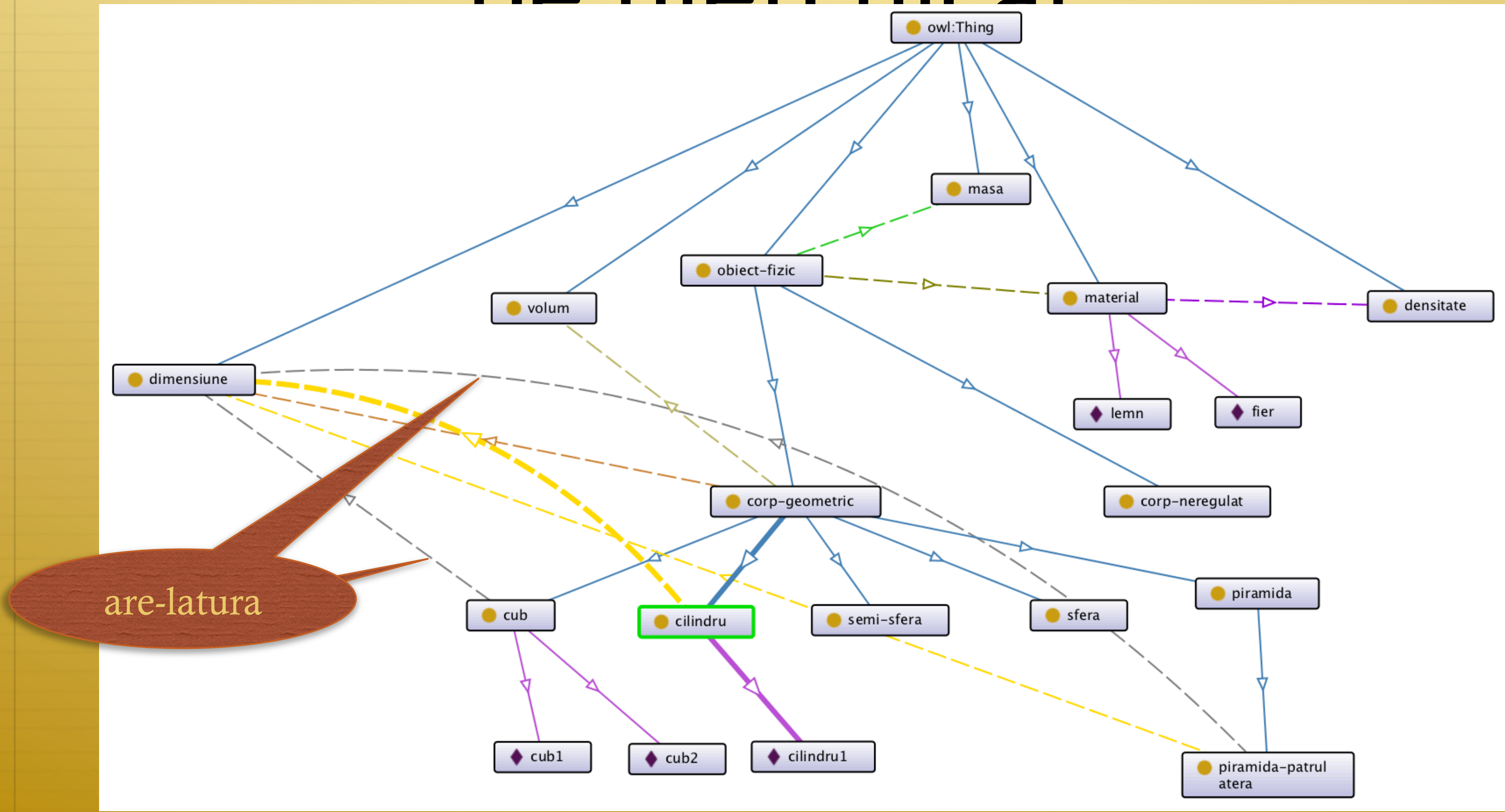


owl:Thing
masa
obiect-fizic
material
densitate
volum
dimensiune
lemn
fier
corp-geometric
corp-neregulat
are-dimensiune
cub
cilindru
semi-sfera
sfera
piramida
cub1
cub2
cilindru1
piramida-patrulatera

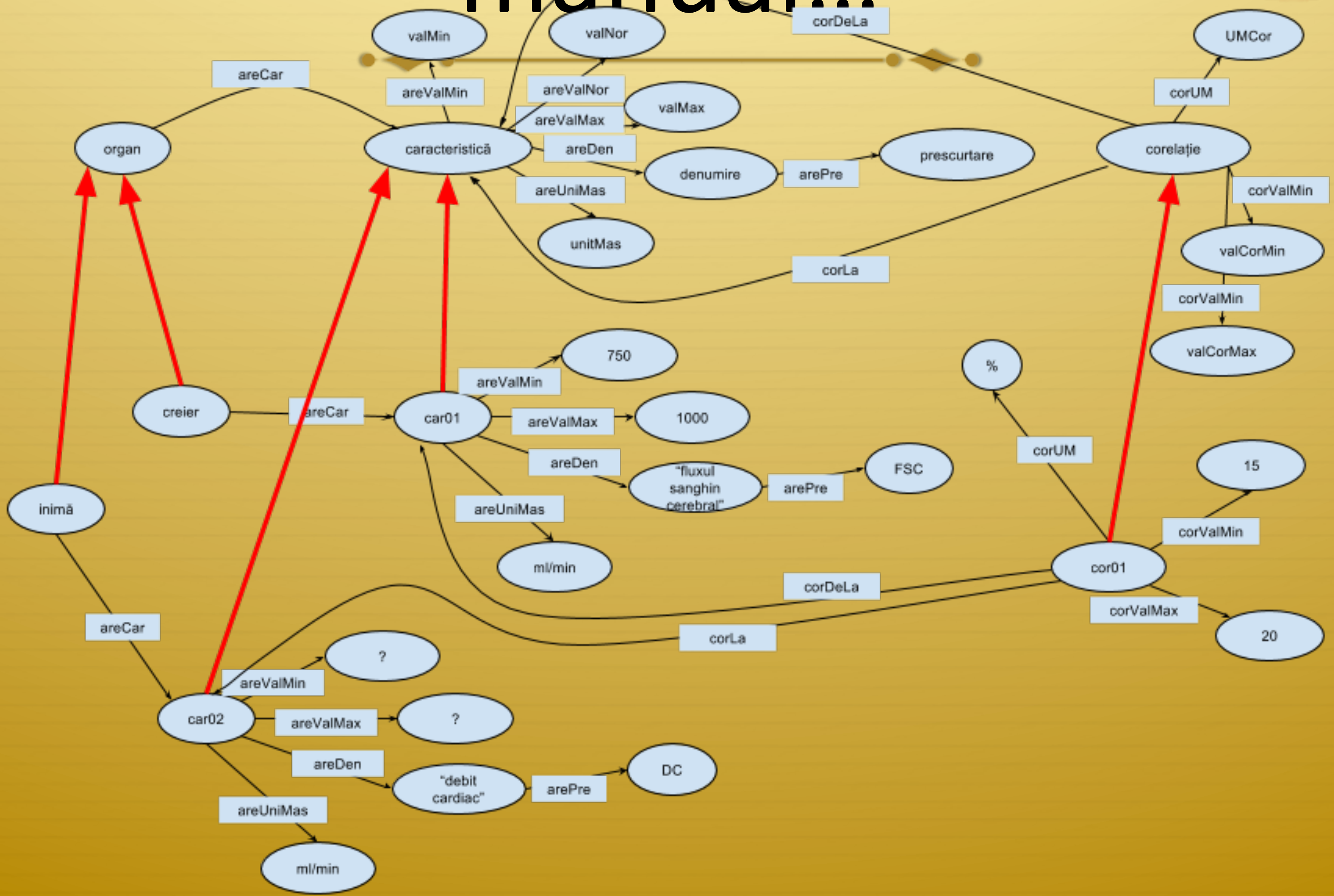
Properties may themselves be hierchical



Properties may themselves be hierchical



A sentence from the manual…



OBO

- The Open Biomedical Ontologies (OBO) Foundry (now The Open Biological and Biomedical Ontologies (OBO) Foundry) is a collaborative experiment involving developers of science-based ontologies. (Smith et al., 2007)
- The Foundry is concerned with establishing a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain. The Foundry approach has been adopted by the Neuroscience Information Framework (NIF) Standard and by the cROP (Common Reference Ontologies for Plants) initiatives.