Ontology

Lab Seminar
Jeong Hankyo

Topics

Ontology

S-100
Ontology

Global
Ontology

Ontology

Definition of ontology

- Neches, 1991,
  "An ontology defiens the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vacabulary."

- Gruber, 1993,
  "An ontology is an explicit specification of a conceptualization."

- Borst, 1997,
  "Ontologies are defined as a formal specification of a shared conceptualization."

- Studer et al., 1998
  "a formal, explicit specification of a shared conceptualization"

Ontology

Definition of ontology

- Neches, 1991,
  "An ontology defiens the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vacabulary."

- Gruber, 1993,
  "An ontology is an explicit specification of a conceptualization."

- Borst, 1997,
  "Ontologies are defined as a formal specification of a shared conceptualization."

- Studer et al., 1998
  *"a **formal**, explicit specification of a shared conceptualization"*

ontology should be machine-readable.

Ontology

Definition of ontology

- Neches, 1991,
  "An ontology defiens the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vacabulary."

- Gruber, 1993,
  "An ontology is an explicit specification of a conceptualization."

- Borst, 1997,
  "Ontologies are defined as a formal specification of a shared conceptualization."

Means that the type of concepts used, and the restrictions on their use are explicitly defined.

- Studer et al., 1998
  "a formal, ***explicit*** *specification of a shared conceptualization"*

Ontology

Definition of ontology

- Neches, 1991,
  "An ontology defiens the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vacabulary."

- Gruber, 1993,
  "An ontology is an explicit specification of a conceptualization."

- Borst, 1997,
  "Ontologies are defined as a formal specification of a shared conceptualization."

- Studer et al., 1998
  *"a formal, explicit specification of a **shared** conceptualization"*

consensual knowledge, that is, it is not the privilege of some individual, but accepted by a group.

Ontology

Definition of ontology

- Neches, 1991,
  "An ontology defiens the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vacabulary."

- Gruber, 1993,
  "An ontology is an explicit specification of a conceptualization."

- Borst, 1997,
  "Ontologies are defined as a formal specification of a shared conceptualization."

- Studer et al., 1998
  *"a formal, explicit specification of a shared **conceptualization**"*

an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon

Ontology

Component of ontology

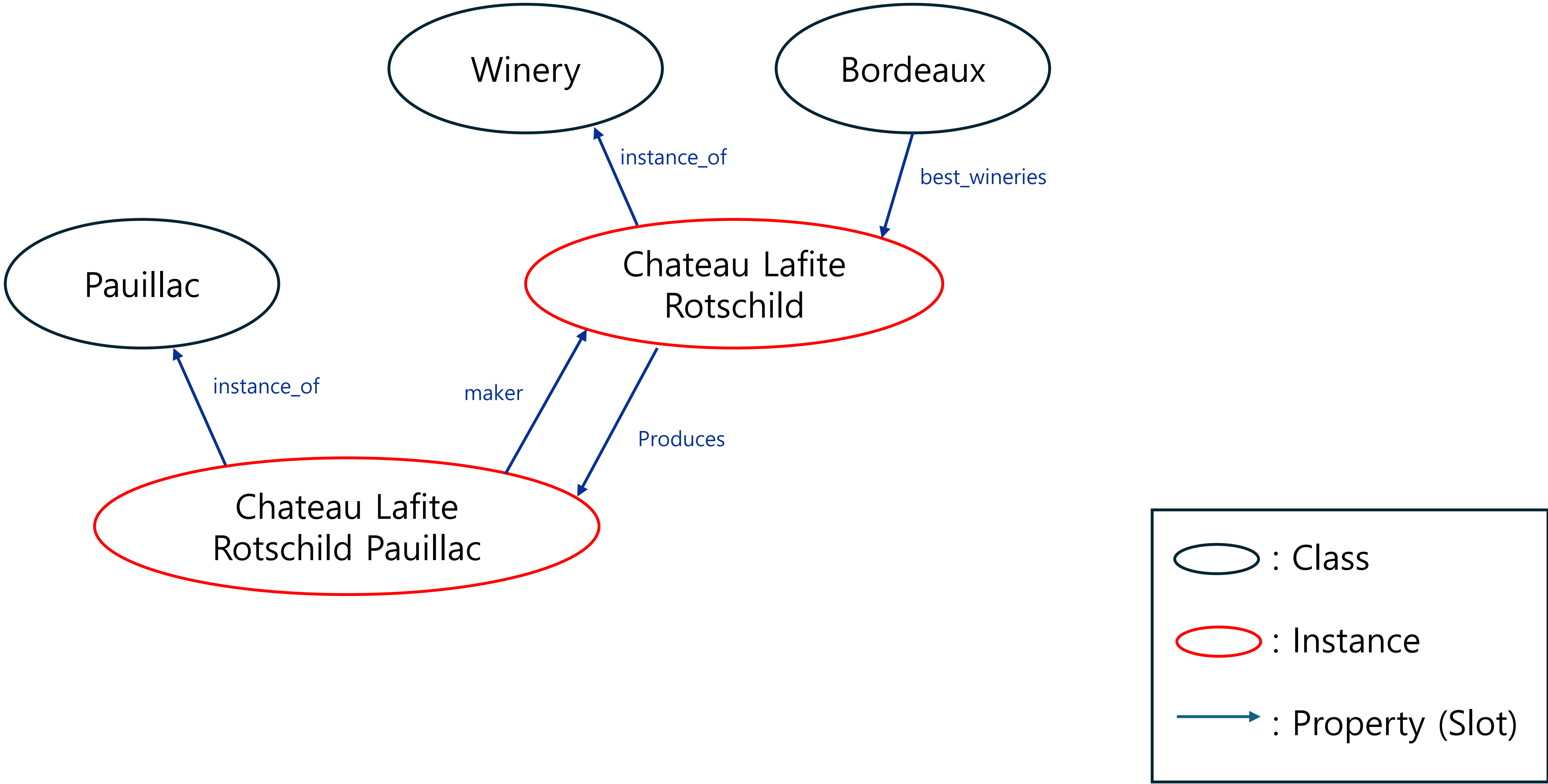
- Class (= concepts)
  - Describe concepts in the domain. (Ex, a class of wines represents all wines)
  - Specific wines are instances of this class.
  - A class can have subclass (Ex, red wines, white wines, rese wines …)

- Slot (= roles or properties)
  - Describe of classes and instances
  - *'Chateau Lafite Rothschild Pauillac'* wine has a full body
  - *'Chateau Lafite Rothschild Pauillac'* wine produced by the *'Chateau Lafite Rothschild'* winery.
  - Body: full, maker: *'Chateau Lafite Rothschild'* winery

- Restrictions (= facets)
  - Constraints applied to slots.
  - (Ex, the maker slot must have exactly one instances of the class winery → minCardinality=1

- Instances
  - Concrete examples of a class.

Ontology



Winery
Bordeaux
instance_of
best_wineries
Pauillac
Chateau Lafite Rotschild
instance_of
maker
Produces
Chateau Lafite Rotschild Pauillac
: Class
: Instance
: Property (Slot)

Knowledge-Engineerring Methodology

Determine the domatin and scope of the ontology

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Knowledge-Enginerring Methodology

Determine the domatin and scope of the ontology

- What is the domain that the ontology will cover? : wine and food
- For what we are going to use the ontology? :
- For what types of questions the information in the ontology should provide answers? :
- Who will use and maintain the ontology?

Knowledge-Engineerring Methodology

Determine the domatin and scope of the ontology

- What is the domain that the ontology will cover? : wine and food
- For what we are going to use the ontology? : suggest good combinations of wines and food.
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Knowledge-Engineerring Methodology

Determine the domatin and scope of the ontology

- What is the domain that the ontology will cover? : wine and food
- For what we are going to use the ontology? : suggest good combinations of wines and food
- For what types of questions the information in the ontology should provide answers?[1] :
- Who will use and maintain the ontology?

1. What wine characteristics should I consider when choosing a wine?
2. Is Bordeaux a red or white wine?
3. Does Cabernet Sauvignon go well with seafood?
4. What is the best choice of wine for grilled meat? …

1) Competency question
Gruninger, Fox 1995, "Methodology for the Design and Evaluation of Ontologies"

Knowledge-Enginerring Methodology

Determine the domatin and scope of the ontology

- What is the domain that the ontology will cover? : wine and food
- For what we are going to use the ontology? :suggest good combinations of wines and food
- For what types of questions the information in the ontology should provide answers? :

1. What wine characteristics should I consider when choosing a wine?
2. Is Bordeaux a red or white wine?
3. Does Cabernet Sauvignon go well with seafood?
4. What is the best choice of wine for grilled meat? …

- Who will use and maintain the ontology? :

  Users: restaurant customers, wine recommendation systems, NLP applications

  Maintainers: sommeliers, ontology engineers, wine retailers, NLP developers

Knowledge-Enginerring Methodology

Develop an ontology

1. Define the classes and the class hierarchy
2. Define the properties of classes – slots
3. Define the facets of the slots
4. Create instances

Knowledge-Enginerring Methodology

1. Define the classes and the class hierarchy

- Wine
- Food

Knowledge-Engineerring Methodology

1. Define the classes and the class hierarchy

- Wine
  - White wine
  - Red wine
  - Rose wine

- Food

Knowledge-Enginerring Methodology

1. Define the classes and the class hierarchy

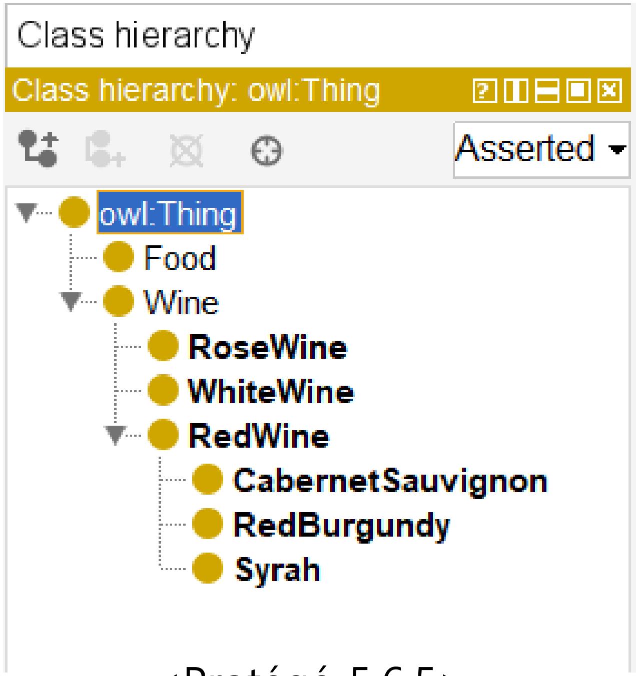
- Wine
  - White wine
  - Red wine
    - Syrah
    - Red Burgundy
    - Cabernet Sauvignon
  - Rose wine

- Food

Knowledge-Engînerring Methodology

1. Define the classes and the class hierarchy

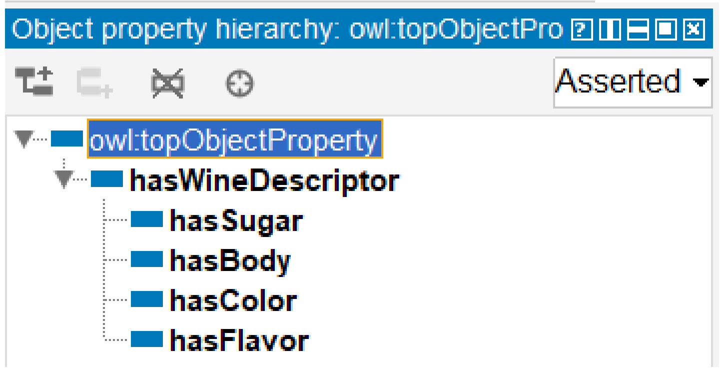
- Wine
    - White wine
    - Red wine
        - Syrah
        - Red Burgundy
        - Cabernet Sauvignon
    - Rose wine
- Food



<Protégé 5.6.5>

Knowledge-Enginerring Methodology

2. Define the properties of classes – slots



- Intrinsic: color, body, flavor, …
- Extrinsic: name, area, …
- Part-of: e.g.,
  MerlotGrape partOf ChateauMargaux2009
- Relationship:
  Wine hasMaker Winery
  Wine madeFromGrape Grape

Knowledge-Enginerring Methodology

3. Define the facets of the slots

Slot cardinality

- how many values a slot can have.

Slot-value type

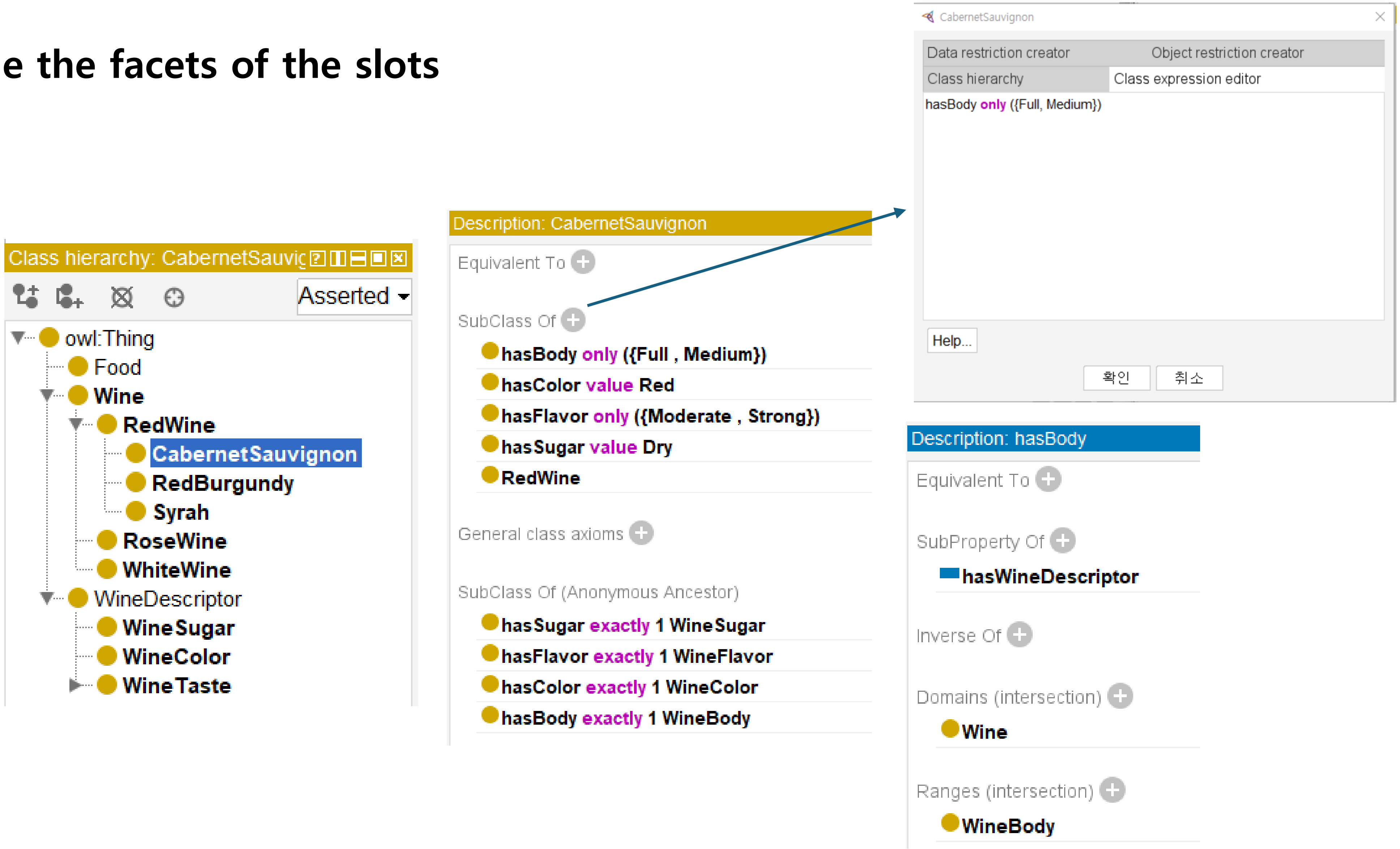
- String:
- Number: Float, Integer,…
- Boolean: true, false
- Enumerated: specify a list of specific allowed values for the slot.
- Instance

Domain and range of a slot

- Domain: the class the slot is attached to
- Range: the type of values it can take

Knowledge-Engінerring Methodology

3. Define the facets of the slots



Knowledge-Engineerring Methodology

4. Create instances

(1) Choosing a calss (Beaujolais)

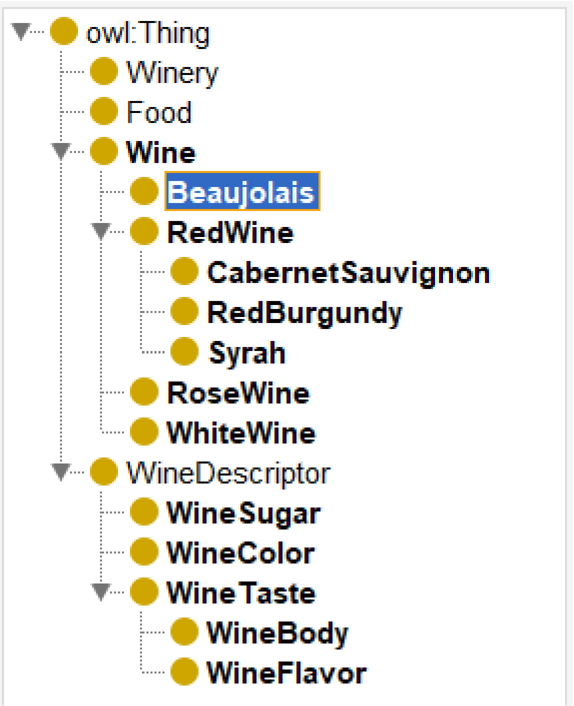
(2) Creating an individual instance of that class (Chateau_Morgon_Beaujolais)

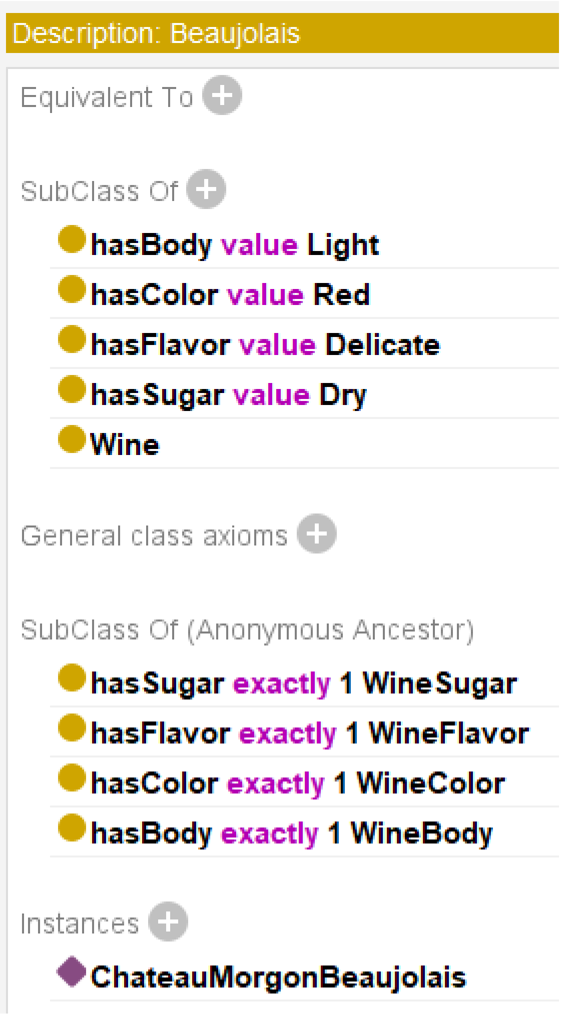
(3) filiing in the slot values.

- Body: Light
- Color: Red
- Flavor: Delicate
- Sugar: Dry
- …

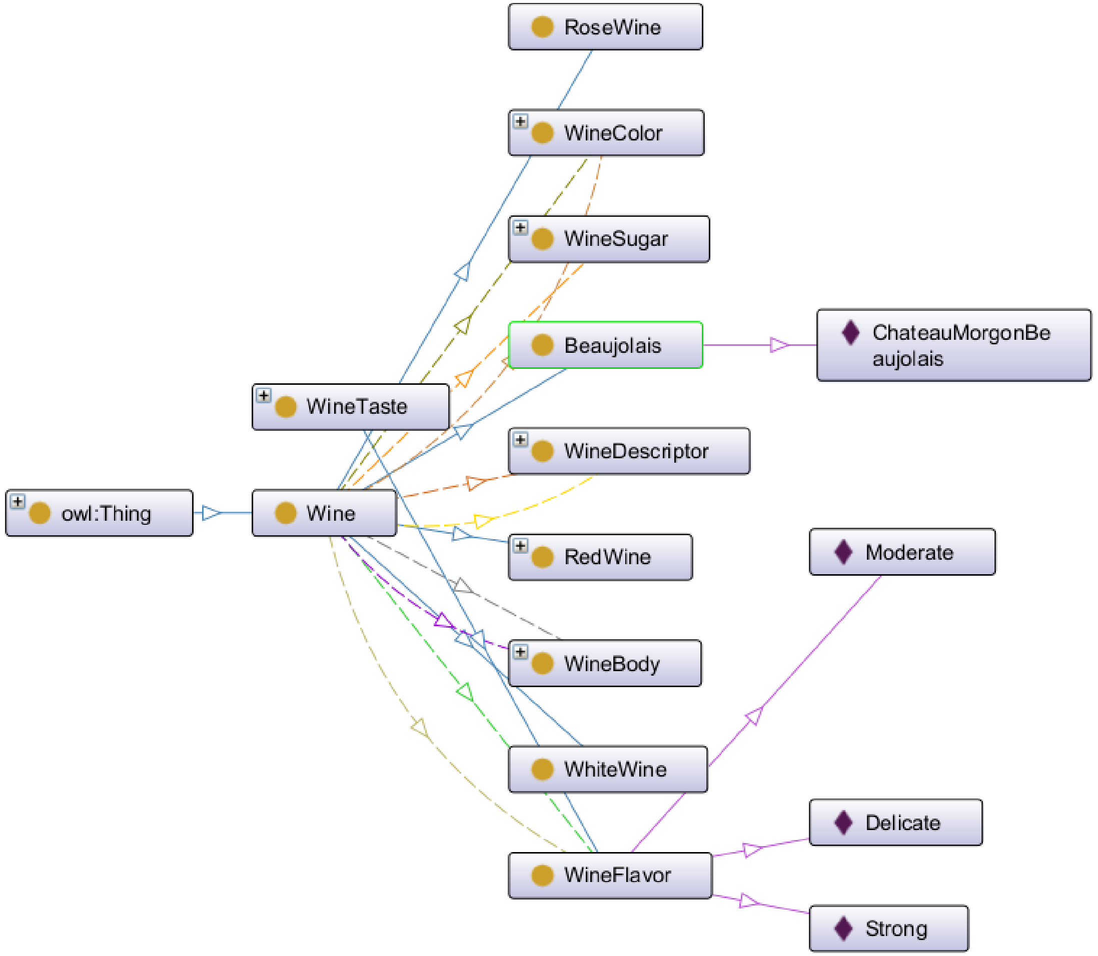
Knowledge-Enginerring Methodology

4. Create instances





Protégé

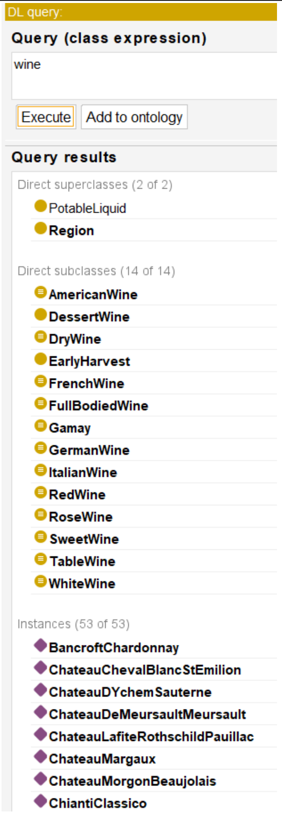


RoseWine
WineColor
WineSugar
Beaujolais
ChateauMorgonBe
aujolais
WineTaste
WineDescriptor
owl:Thing
Wine
RedWine
Moderate
WineBody
WhiteWine
Delicate
WineFlavor
Strong

Reasoner (DL Query / SPARQL Query)

1. **What wine characteristics should I consider when choosing a wine?**

DL Query (Description Logic)



SPARQL Query

Query:

```
SELECT DISTINCT ?property
WHERE {
  wine:Wine rdfs:subClassOf ?restriction .
  ?restriction owl:onProperty ?property .
}
```

Output:

hasBody
hasFlavor
locatedIn
hasSugar
hasColor
hasMaker
madeFromGrape

Reasoner (DL Query / SPARQL Query)

2. **Is Bordeaux a red or white wine?**

DL Query (Description Logic)

DL query:

Query (class expression)

Bordeaux

Execute | Add to ontology

Query results

Direct superclasses (1 of 1)

- **FrenchWine**

Direct subclasses (2 of 2)

- **RedBordeaux**
- **WhiteBordeaux**

Instances (4 of 4)

- **ChateauChevalBlancStEmilion**
- **ChateauDYchemSauterne**
- **ChateauLafiteRothschildPauillac**
- **ChateauMargaux**

SPARQL Query

Query:

```
SELECT ?instance ?type ?color ?flavor ?body
WHERE {
  ?instance rdf:type ?type .
  ?type rdfs:subClassOf* wine:Bordeaux .
  OPTIONAL { ?instance wine:hasColor ?color . }
  OPTIONAL { ?instance wine:hasFlavor ?flavor . }
  OPTIONAL { ?instance wine:hasBody ?body . }
}
```

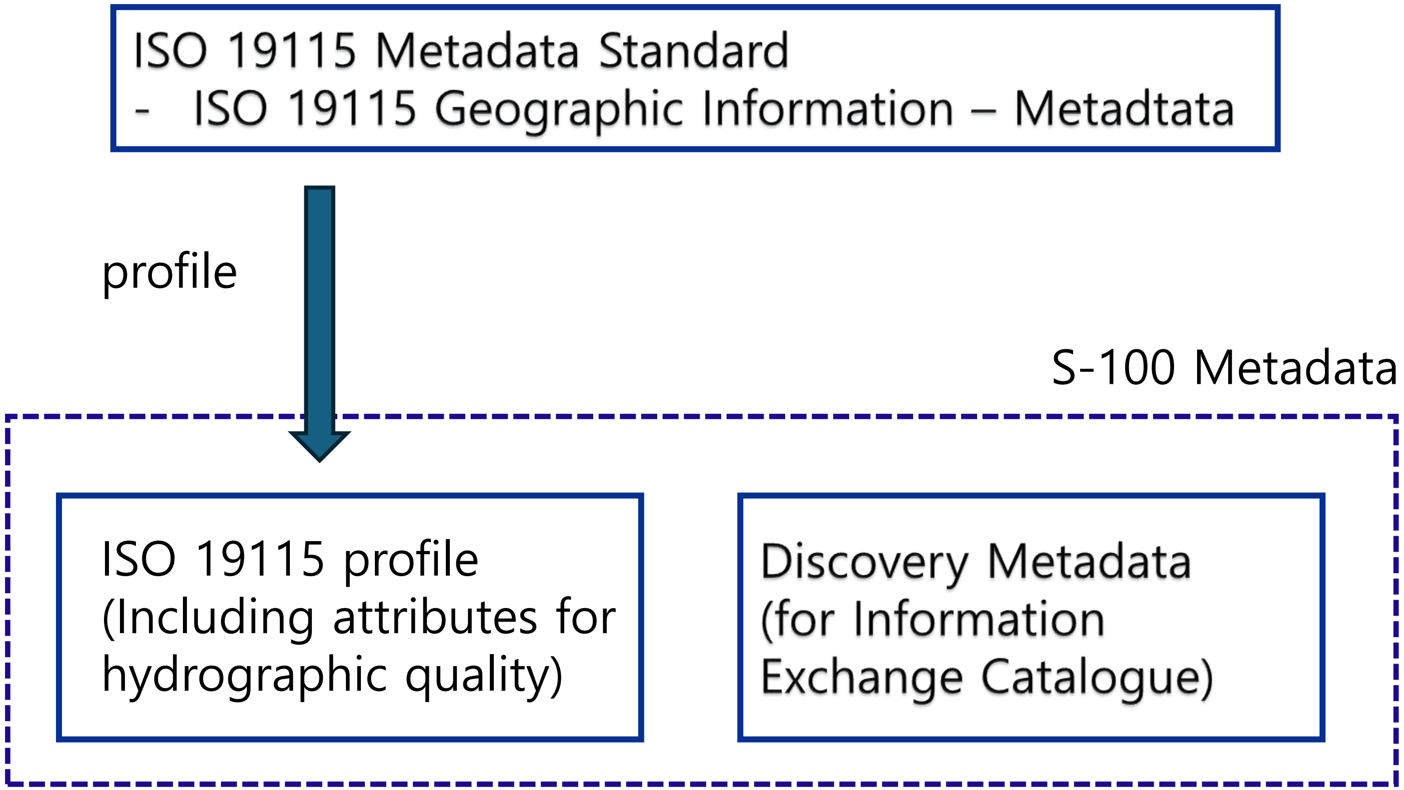
Output:

instance	type	color
ChateauDYchemSauterne	Sauternes	Red
ChateauChevalBlancStEmilion	StEmilion	Red
ChateauMargaux	Margaux	Red
ChateauLafiteRothschildPauillac	Pauillac	Red

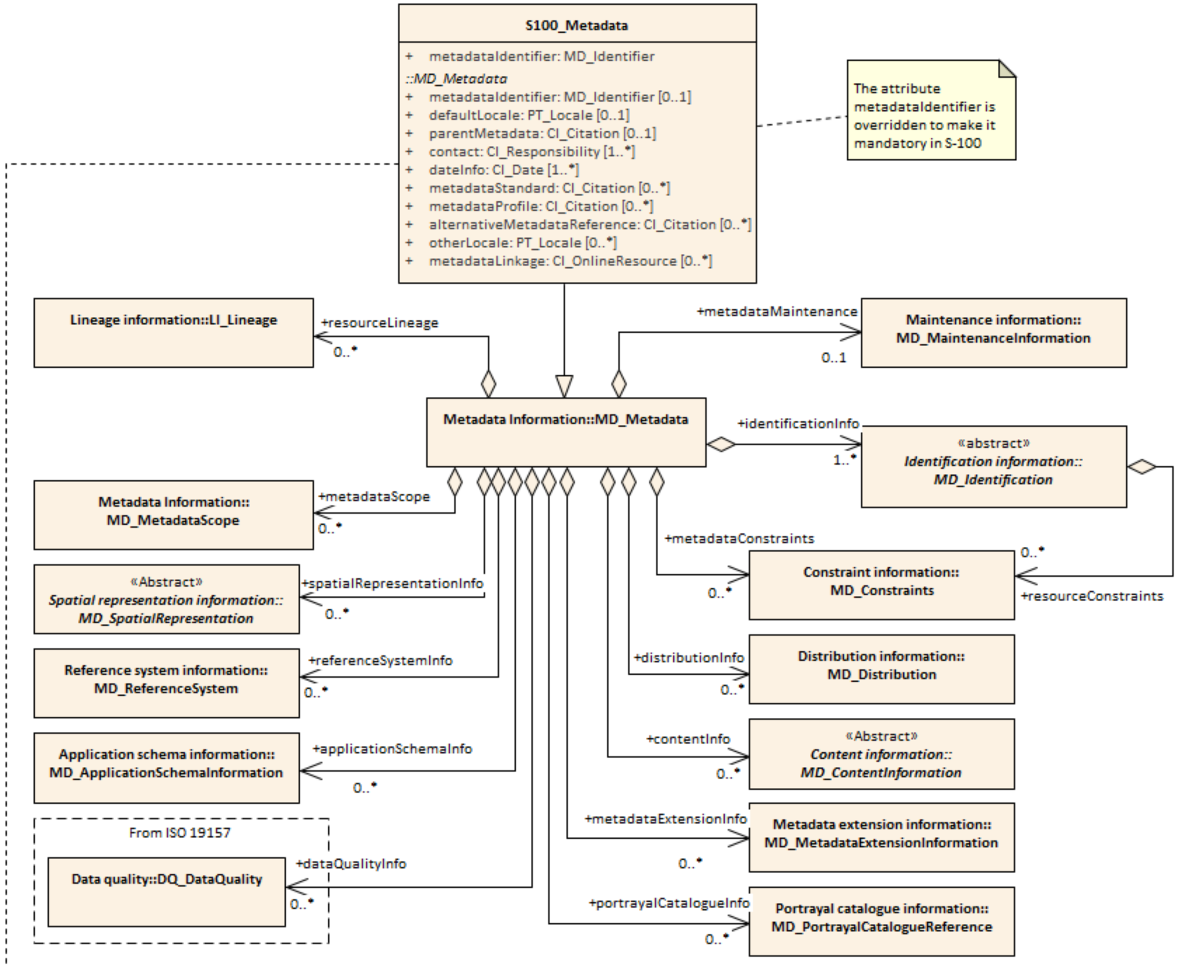
S-100, Ontology

Suhyun Park, 2010, “S-100 Metadata Conversion Design of the OWL-based Ontology”

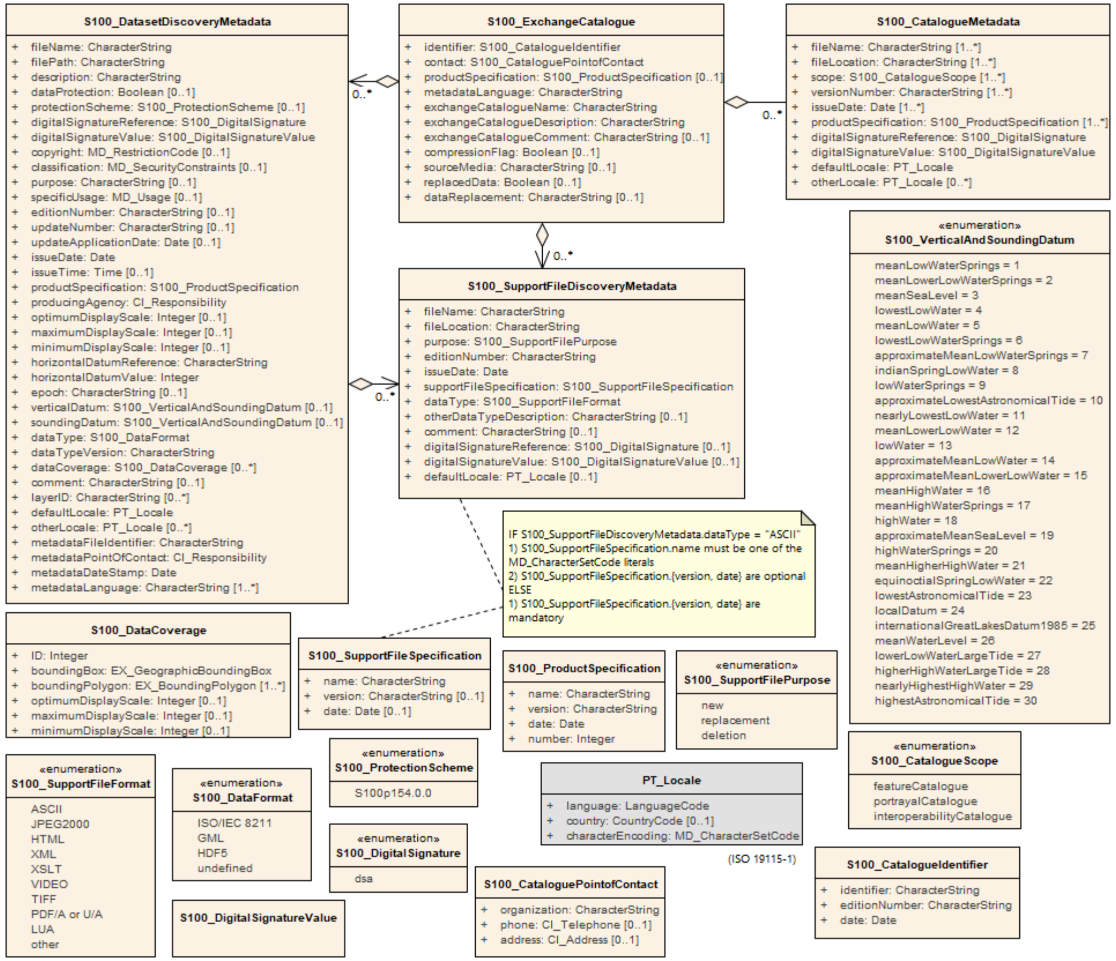
S-100 Metadata



S-100 Metadata



ISO 19115 profile metadata



Discovery metadata

S-100 Metadata

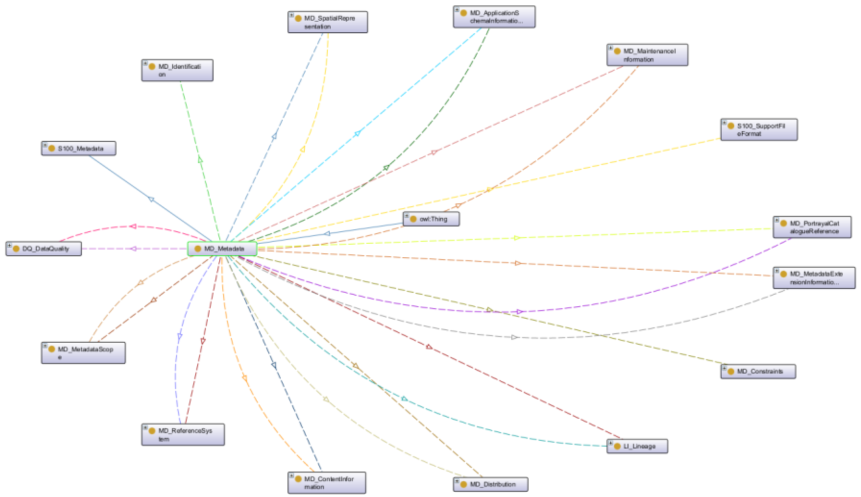
S100_CataloguePointofContact

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_CataloguePointOfContact	Contact details of the issuer of this exchange catalogue	-	-	-
Attribute	organization	The organization distributing this exchange catalogue	1	CharacterString	This could be an individual producer, value added reseller, etc.
Attribute	phone	The phone number of the organization	0..1	CI_Telephone	
Attribute	address	The address of the organization	0..1	CI_Address	

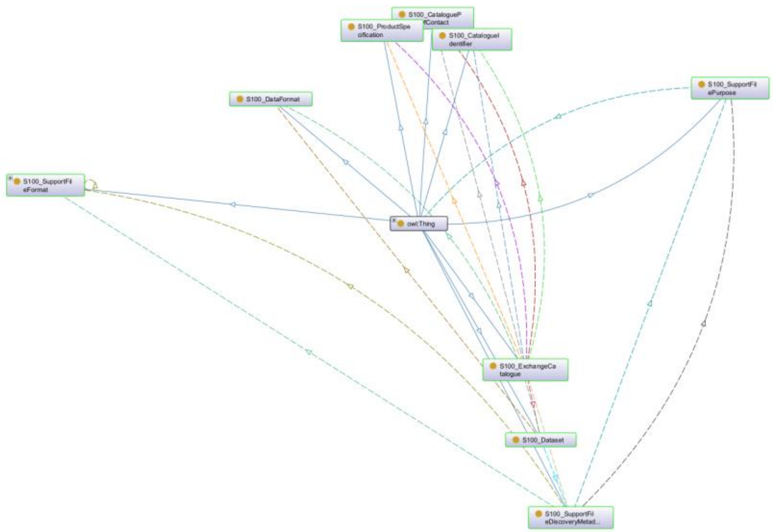
S100_Dataset

Role Name	Name	Description	Mult	Type	Remarks
Class	S100_Dataset		-	-	-
Role	composedOf	An exchange set is composed of 0 or more datasets	0..*	-	
Role	datasetCatalogue	Catalogue which is related to this dataset	0..*	-	

S-100 Metadata Ontology



ISO 19115 Metadata



S-100 Discovery Metadata

Global Ontology

Reza Asgari et al. 2015, “An ontology-based approach for integrating heterogeneous databases”

Problem

Traditional

DB1

Calculator_model	Calculator_type
M19	Programmer
N34	Scientific

DB2

Device_number	Category
M19	Programmer
N34	Scientific

- Traditional schema-based integration methods struggle when database fidels differ in naming. (e.g, “Calculator_model” vs “Device_number”)

- For integration, require manual semantic matching

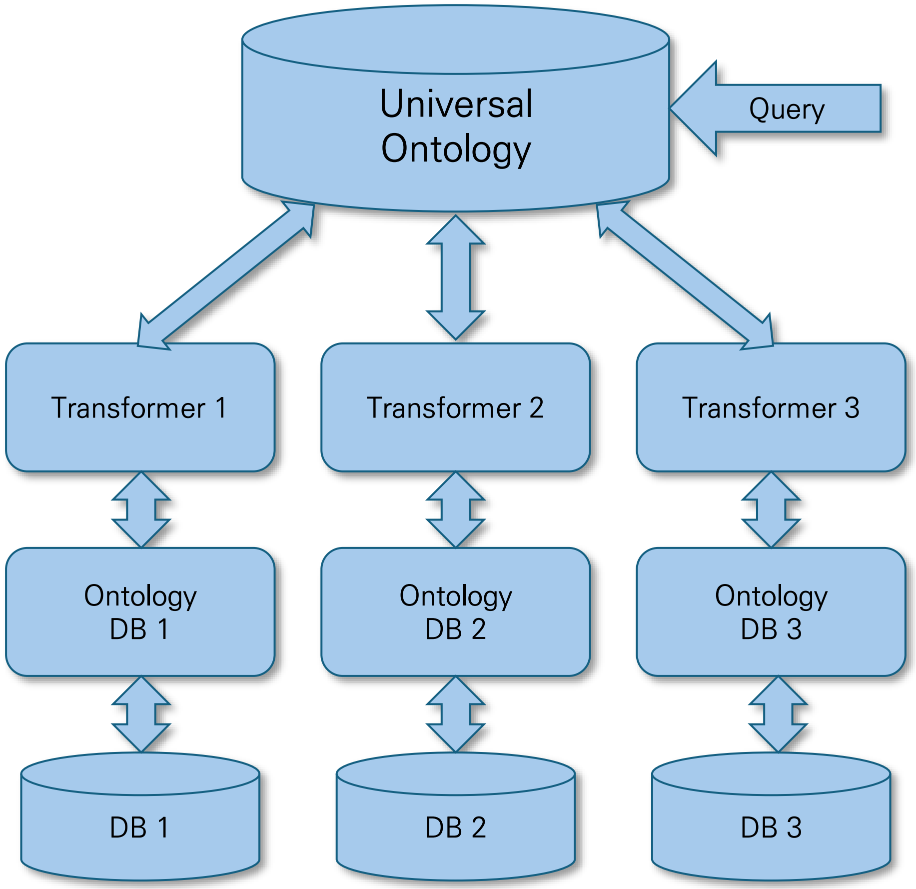
Problem

Proposal

DB_1	Relation	DB_2
Calculator_model	synOf	Device_number
Calculator_type	synOf	Category

- Explicitly defines semantic relationships, facilitating automatic and accurate integration.
- User can unified queries, and the system automatically matches fields semantically

Hybrid architecture



Ontology

The rules for constructing the ontology

- Deifne ontology "O" as: *O = {C, A, I, R}*
  C : a set of concepts.
  A : a set of attributes of concepts.
  I : a set of instances.
  R : a set of all relations that exist between *'C', 'A'* and *'I'.*

- *R = {isA, synOf, partOf, atr, val}*
  isA: Inheritance relations: 'CPU isA Processor'
  synOf: Synonyms relations: 'PC synOf PersonalComputer'
  partOf: Aggregation relations: 'Keyboard partOf Computer'
  atr: showing attributes of a concept: 'atr(CPU, company)'
  val: value of attribues: 'val(company, 'Intel', cpu_3034)

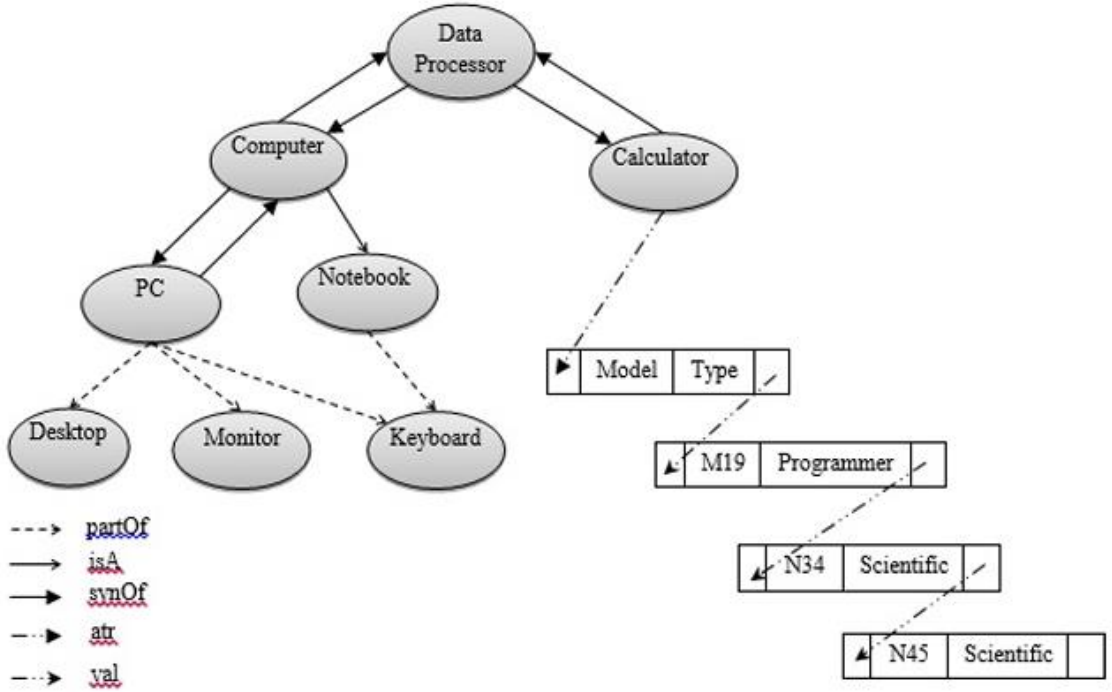
cpu_3034's value of company is "Intel"

Ontology

The rules for constructing the ontology

Type	Model	Name
LCD	S1227i	Desktop
Single processor	Dell	Notebook
Usb port	L11h23	Keyboard
Programmer	M19	Calculator
LED	121g2	Monitor
Scientific	N34	Calculator
Desktop pc	Intel	PC
Multi processor	IBM	Computer
Scientific	N45	Calculator

Table 1.
Sample table that contains product information.



Part of product ontology graph for Table 1.

Flow

Integration

- Constructing one ontology for each database.

- Constructing mapping function(here, called transformer) as an intermediate language and as a communication bridge among the local ontologies and the universe ontology

- Construct universal Ontology

Flow

Response to the user's query

Query: "Show me the IDs of all products where the Name is 'Laptop' and the Color is 'Silver.'

1. *The query is encoded from Universal ontology as:*

F = (c = Product,
a = {Name, Color},
i = {Name='Laptop', Color='Silver'},
s = {product_name ↔ name, color ↔ hue}, acc)

2. *After verigying acc, strip is away:*

F' = (c = Product,
a = {Name, Color},
i = {Name='Laptop', Color='Silver'},
s = {product_name ↔ name, color ↔ hue})

Flow

Response to the user's query

3. Transformer (mapping): convert F' into the SQL query.

Transformer	Semantic mapping	Sub-query
T1	Name ↔ product_name Color ↔ color	SELECT product_id FROM DB1 WHERE product_name='Laptop' AND color='Silver'
T2	Name ↔ name Color ↔ hue	SELECT id FROM DB2 WHERE name='Laptop' AND hue='Silver'

Flow

Response to the user's query

4. Local ontology → each DB → Return response

DB1

Product_id
101 102

DB2

id
201 202

5. Filtering and output F″

F''_{T1}={101,102}, F''_{T2} ={201,202}

6. The universal Ontology merges both sets into {101, 102, 201, 202} and final return to the user

Flow

Response to the user's query

4. Local ontology → each DB → Return response

DB1

Product_id
101 102

DB2

id
201 202

5. Filtering and output F″

F''_{T1}={101,102}, F''_{T2} ={201,202}

6. The universal Ontology merges both sets into {101, 102, 201, 202} and final return to the user

감사합니다.