

# Ontologia

Abbiamo visto che tassonomie e tesauri fissano una semantica. Per arricchire la semantica si deve passare a modelli concettuali e teorie logiche. Un modello concettuale è il modello di una particolare area di conoscenza o di attività, chiamata dominio, che rappresenta le entità del dominio, le relazioni che intercorrono fra queste, espresse sotto forma di attributi (proprietà) delle entità e dei valori che questi attributi possono avere. Inoltre possiamo dare delle regole che riguardano le classi, gli attributi e le loro relazioni.

Un modello concettuale è simile a quello che si usa nella programmazione orientata agli oggetti dove possiamo definire classi e sottoclassi.

Anche le teorie logiche possono essere impiegate per descrivere modelli concettuali. In questo caso si hanno assiomi e regole di inferenza. Da questi si possono dimostrare dei teoremi per far vedere che qualcosa è vero.

Per esempio si può dire che la relazione *è\_sottoclasse\_di* è transitiva. Che la relazione *è\_superclasse\_di* è la sua inversa, che due sottoclassi di un medesimo concetto sono disgiunte (cioè non hanno elementi in comune). La logica che si occupa di questo viene chiamata “logica descrittiva”.

Una definizione di ontologia:

- Un’ontologia è una *descrizione formale esplicita di un dominio* di interesse.
- **Descrizione**: una forma di rappresentazione della conoscenza
- **Formale**: simbolica e meccanizzabile
- **Esplicita**: elenchi estensionali di frammenti di conoscenza
- **Dominio**: ristretta ad un determinato sottoinsieme dello scibile, affrontato da un certo punto di vista.

Una ontologia descrive le parole comuni e i concetti (significati) usati per descrivere e rappresentare un’area di conoscenza (dominio).

Una ontologia può essere usata da persone, applicazioni, database etc. per condividere una conoscenza comune riguardo ad un certo dominio (educazione, medicina, riparazione di automobili etc.). L’ontologia include le definizioni dei concetti del dominio e delle loro relazioni in un modo usabile dal computer (ma anche comprensibile agli umani).

Quindi una ontologia è costituita da:

- Classi (concetti generali del dominio di interesse.)
- Relazioni tra queste classi
- Proprietà (attributi, slot, ruoli) assegnate a ciascun concetto, che ne descrivono vari tipi di attributi o proprietà
- Restrizioni sulle proprietà (facet, role restrictions). Impongono il tipo di dato sul valore che la proprietà può assumere.

Istanze

- A partire dalle classi dell’ontologia, è possibile definire delle istanze, che rappresentano specifici oggetti del mondo reale
- Le istanze ereditano attributi e relazioni dalle classi
- Knowledge base = Ontologia + insieme delle istanze delle classi

## Un esempio di ontologia.

Si vogliono descrivere i libri, le case editrici e le persone (l'esempio è illustrato con riferimento al tool per la costruzione di ontologie Protege <http://protege.stanford.edu>):

The screenshot shows the Protege-2000 interface with the title bar "Libri Protégé-2000 (G:\Documents and Settings\...)" and the file path "My Documents\Ontologie\WebSemantico\Libri\RDF\Libri". The menu bar includes Project, Edit, Window, Help. The toolbar has icons for New, Open, Save, and Delete. The tabs at the top are Classes, Slots, Forms, Instances, and Queries, with Classes selected. Below the tabs is a toolbar with Relationship, Super c..., V, C, and X buttons. On the left, a tree view shows the class hierarchy: :THING -> SYSTEM-CLASS -> Libri:casa\_editrice -> Libri:libro, which has subclasses Libri:manuale and Libri:romanzo, and a generalization Libri:persona. The main panel displays the details for the class "Libri:libro". The "Name" field contains "Libri:libro". The "Documentation" field is empty. The "Role" dropdown is set to "Concrete". In the "Template Slots" section, there is a table:

Name	Type	Cardinality
Libri:autore	Instance	required multiple
Libri:editore	Instance	required single
Libri:titolo	String	single

La classe libri ha due sottoclassi manuale e romanzo. La classe libro ha tre slot autore, editore e titolo. Ci sono delle restrizioni sugli slot. Autore deve essere una istanza di persona (una assunzione semplificativa) e deve esserci almeno 1 autore, ma anche più di uno (required multiple in protege). Editore ha come valore una istanza obbligatoria e unica di casa editrice. Titolo è un tipo base cioè striga ed è obbligatorio ed unico. Le sottoclassi manuale e romanzo ereditano gli stessi slot ma possono aggiungerne altri.

La classe romanzo eredita questi tre slot ma ne aggiunge uno, genere di tipo simbolo che può assumere solo i valori elencati (storico, verista ..) e di numerosità single (cioè solo 1).

Libri Protégé 2.1 (file:C:\Documents and Settings\Administrator\Desktop\OntologieWebSemantico\LibriRDF\Libri.ppr...)

Project Edit Window Help

A R

Classes Slots Forms Instances Queries Ontowiz

Relationship Supercl... V C X

THING A

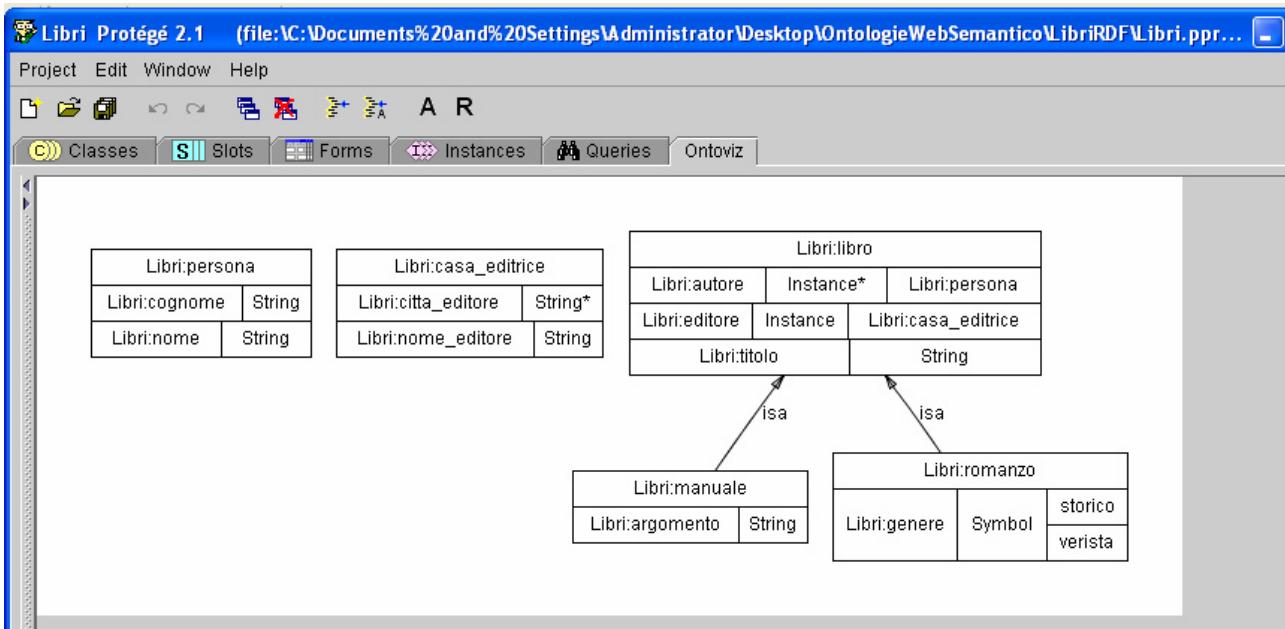
- SYSTEM-CLASS A
- Libri:casa\_editrice
- Libri:libro
  - Libri:manuale
  - Libri:romanzo
- Libri:persona

**Libri:romanzo (type:=STANDARD-CLASS)**

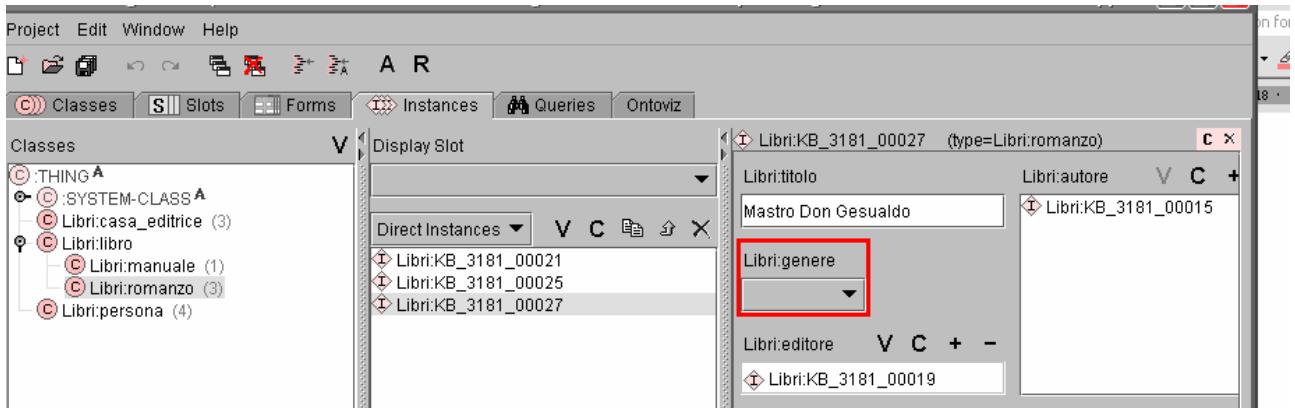
Name	Documentation	Constraints	
Libri:romanzo			
Role	Concrete	V C + -	
Template Slots			
Name	Type	Cardinality	Other Facets
S Libri:genere	Symbol	single	allowed-values={storico,verista}
S Libri:autore	Instance	required multiple	classes={Libri:persona}
S Libri:editore	Instance	required single	classes={Libri:casa_editrice}
S Libri:titolo	String	single	

In maniera analoga possiamo definire le classi persona e casa\_editrice.

Quella che segue è una immagine della struttura delle classi ottenuta con ontowiz (un plugin di Protege). Sotto il nome della classe (conceitto) si vedono gli slot (proprietà) ed i valori che possono assumere.

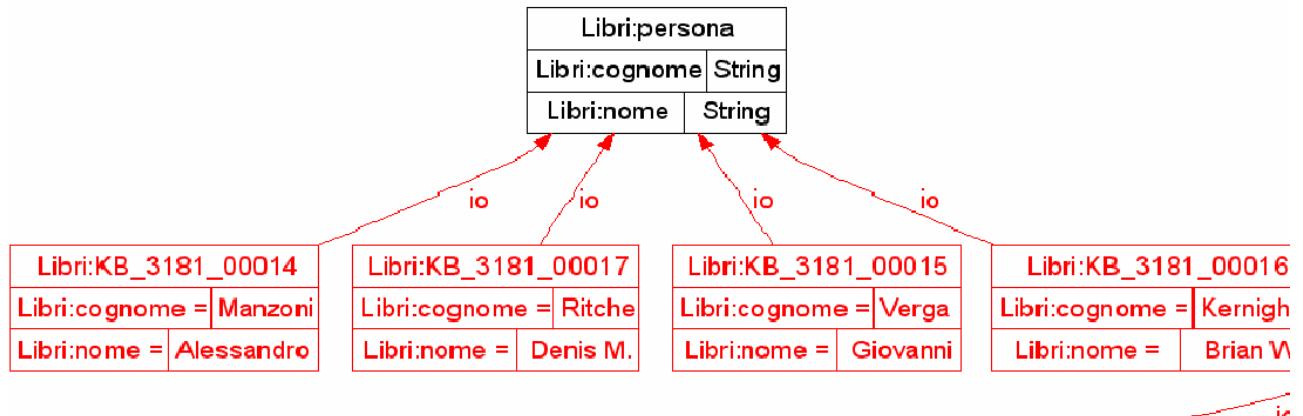


Definita l'ontologia possiamo passare a definire le istanze delle varie classi.

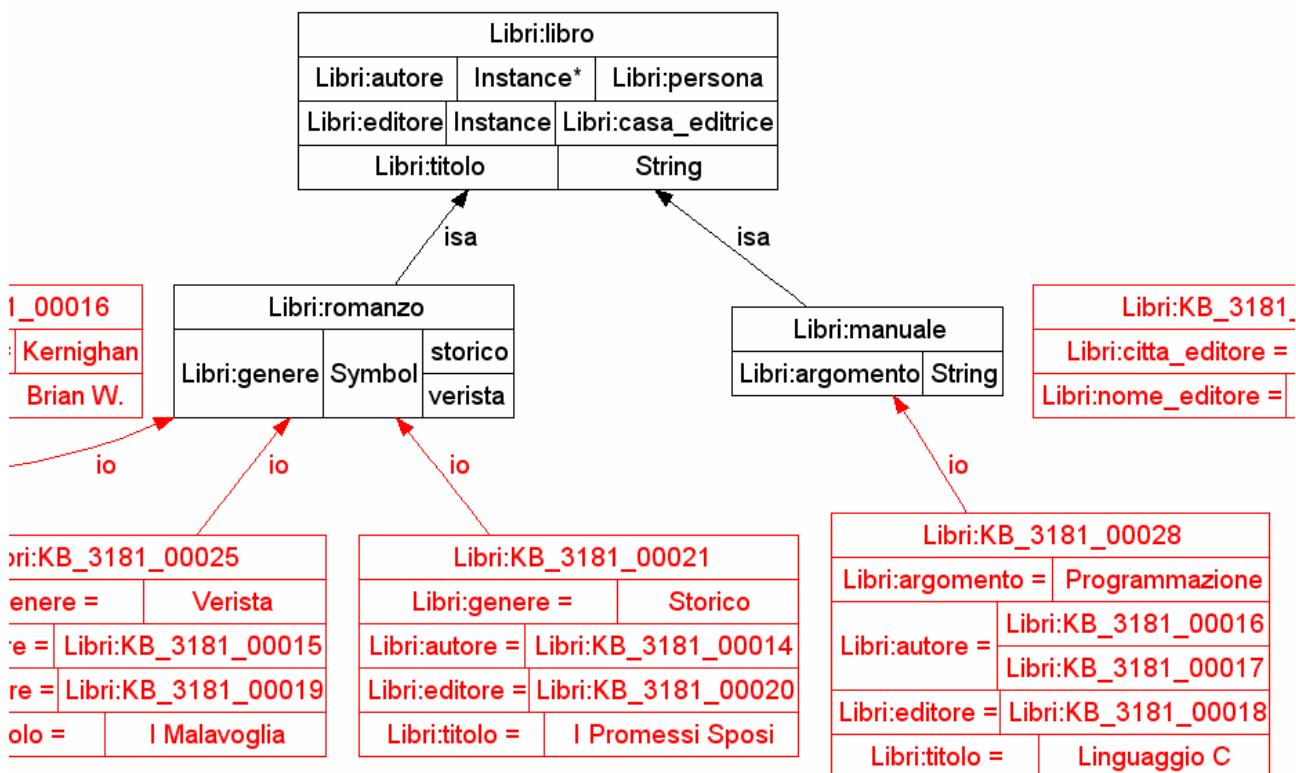


In questo esempio vediamo le istanze di romanzo.

Possiamo vedere le istanze in modo grafico.

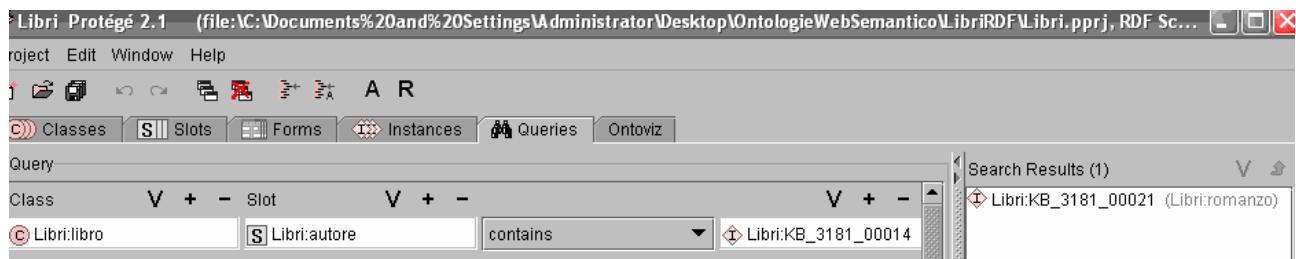


Le istanze di persona.



e le istanze di libri. Analogamente si possono vedere le istanze di casa editrice.

A questo punto abbiamo una base di conoscenza e possiamo fare delle interrogazioni (o query).



Sempre usando le Protege possiamo fare delle query.

Vogliamo cercare un libro il cui autore sia Manzoni. Come possiamo vedere dalle figure precedenti Libri:KB\_3181\_00014 è una istanza di persona che corrisponde a Manzoni. Il risultato Libri:KB\_3181\_00021 corrisponde a I Promessi Sposi. (purtroppo dobbiamo usare gli identificatori che Protege assegna automaticamente alle istanze).

E' opportuno notare che per rispondere bisogna fare una inferenza. Infatti I Promessi Sposi sono una istanza di romanzo, non di libro. Ma sapendo che romanzo è una sottoclasse di libro possiamo anche dire che sono una istanza libro. Protege segnala nella risposta che l'istanza trovata appartiene alla classe romanzo.

Potremmo chiederci come sono rappresentate le ontologie. Questo esempio è stato salvato come RDF/RDFS.

Le istanze sono rappresentate nella notazione RDF. La risorsa **KB\_3181\_00021** ha come genere “storico”, come titolo “I Promessi Sposi” come autore la risorsa **KB\_3181\_00014** che corrisponde a una persona (Manzoni).

In realtà la risorsa si chiama [http://protege.stanford.edu/Libri#KB\\_3181\\_00021](http://protege.stanford.edu/Libri#KB_3181_00021), cioè si fa riferimento al namespace: //protege.stanford.edu/Libri

```
<Libri:romanzo rdf:about="http://protege.stanford.edu/Libri#KB_3181_00021"
  Libri:genere="Storico" Libri:titolo="I Promessi Sposi"
  rdfs:label="Libri:KB_3181_00021">
<Libri:autore rdf:resource="http://protege.stanford.edu/Libri#KB_3181_00014" />
<Libri:editore rdf:resource="http://protege.stanford.edu/Libri#KB_3181_00020" />
</Libri:romanzo>
```

```
<Libri:persona rdf:about="http://protege.stanford.edu/Libri#KB_3181_00014"
  Libri:cognome="Manzoni" Libri:nome="Alessandro" rdfs:label="Libri:KB_3181_00014" />
```

L’ontologia è invece rappresentata in un file RDFS (Schema RDF).

```
<rdfs:Class rdf:about="http://protege.stanford.edu/Libri#libro" rdfs:label="Libri:libro">
<rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
  19990303#Resource" />
</rdfs:Class>
<rdfs:Class rdf:about="http://protege.stanford.edu/Libri#manuale"
  rdfs:label="Libri:manuale">
<rdfs:subClassOf rdf:resource="http://protege.stanford.edu/Libri#libro" />
</rdfs:Class>
<rdfs:Class rdf:about="http://protege.stanford.edu/Libri#romanzo"
  rdfs:label="Libri:romanzo">
<rdfs:subClassOf rdf:resource="http://protege.stanford.edu/Libri#libro" />
</rdfs:Class>
```

Libro, manuale e romanzo sono classi. Libro è sottoclasse di Risorsa (Il concetto più generale) mentre manuale e romanzo sono sottoclasse di libro.

```
<rdf:Property rdf:about="http://protege.stanford.edu/Libri#autore" a:minCardinality="1"
  rdfs:label="Libri:autore">
<rdfs:domain rdf:resource="http://protege.stanford.edu/Libri#libro" />
<rdfs:range rdf:resource="http://protege.stanford.edu/Libri#persona" />
</rdf:Property>
```

Autore è una “proprietà” con una cardinalità mini 1 (obbligatorio). Il dominio della Proprietà è “libro” (cioè è uno slot della classe libro). Il codominio è la classe “persona” (cioè il valore che la proprietà può assumere è una istanza della classe “persona”).

Visto che l’ontologia è rappresentata come RDF/RDFS possiamo fare delle query utilizzando RDQL che come abbiamo visto è il linguaggio di i

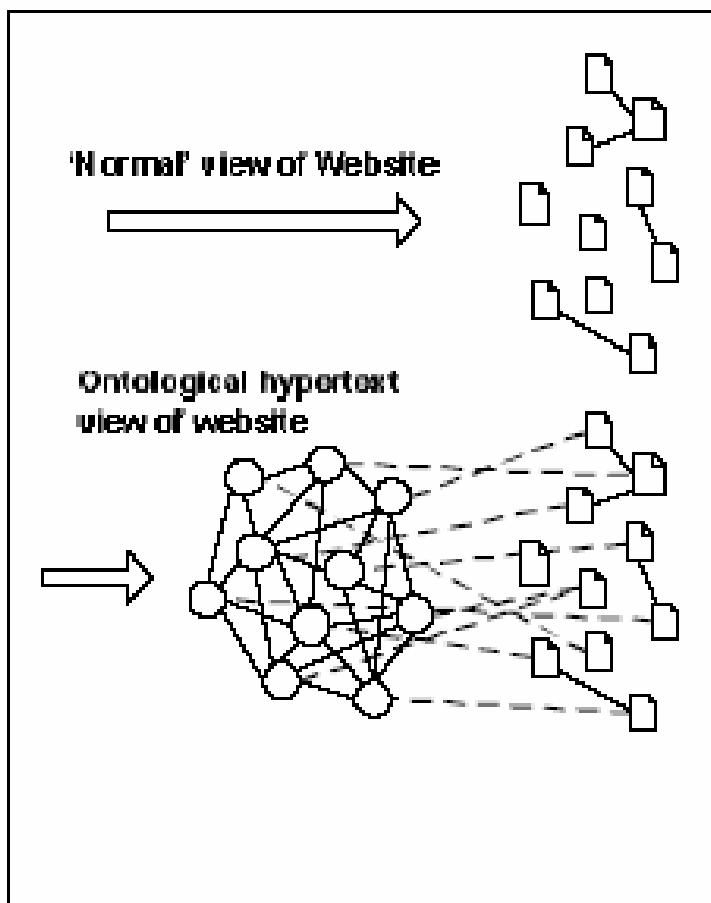
La query vista prima (trova un libro il cui autore sia Manzoni potrebbe essere scritta nel linguaggio RDQL in questo modo:

```
SELECT ?x
FROM <libri.rdf>
WHERE (?x,<rdf:type>,<Libri:libro>),
      (?x,<Libri:autore>,<?y>),
      (?y,<Libri:cognomen>,<?z>)
AND ?z=="Manzoni"
USING Libri for http://protege.stanford.edu/Libri
      rdf for ...
```

Si cerca una istanza `?x` di libro, questa istanza `?x` ha un autore `?y`? (che sarà una persona). La persona `?y`

## Annotazioni

- Un'annotazione è una relazione tra un'istanza di un oggetto ed una classe dell'ontologia
- È rappresentabile mediante una tripla <oggetto, tipo, classe>
- L'oggetto non deve necessariamente essere un'istanza di una classe dell'ontologia
- L'annotazione può avere delle proprietà (attributi)
- Un singolo oggetto può possedere diverse annotazioni
- Es. metadati che fanno riferimento a tassonomie o dizionari esterni.



Le pagine web sono annotate mediante l'ontologia. C'è una struttura del sito web dove le pagine sono legate fra loro dai link ipertestuali (che non hanno una struttura o una interpretazione semantica diretta).

Mediane le annotazioni possiamo stabilire legami fra le pagine web e i nodi di una ontologia di dominio. Usando l'ontologia possiamo navigare il nostro sito muovendoci fra i nodi dell'ontologia.

# Progettare Ontologie

## Definire un'ontologia

Per definire una ontologia dobbiamo:

- Definire le classi
- Organizzare le classi in una gerarchia tassonomica (sottoclassi-superclassi)
- Definire le proprietà e descrivere i valori leciti per ciascuna di esse
- Attribuire i valori alle proprietà per tutte le istanze create.

### 1. Elencare i termini più importanti

Si può iniziare scrivendo una lista di tutti i termini del dominio su cui vogliamo fare affermazioni.

1. Cerchiamo di rispondere alle seguenti domande:
2. Quali sono i termini di cui vogliamo parlare?
3. Quali sono le proprietà ad essi associate?
4. Cosa vogliamo dire a proposito di essi?

### 2. Definire le classi e la gerarchia

Ci sono diversi metodi per sviluppare l'organizzazione gerarchica dell'ontologia:

1. **Top-down:** si parte dalla definizione dei concetti più generali e si specializzano tali definizioni in concetti più specifici.
2. **Bottom-up:** si parte dalla definizione delle classi più specifiche, e si cerca di raggrupparle in gruppi concettuali più astratti.

Quasi sempre si segue una combinazione dei metodi Top-down e Bottom-up.

### 3. Definire le proprietà delle classi

Le proprietà degli oggetti verranno modellate nelle classi relative:

1. Proprietà "intrinsiche" properties (il gusto di un vino)
2. Proprietà "estrinsiche" (il nome di un vino)
3. Parti costituenti, se l'oggetto è strutturato (le parti del corpo)
4. Relazioni ad altri soggetti (costruttore di ...)

Gli attributi sono ereditati dalle sottoclassi.

### 4. Definire i vincoli sulle proprietà

Le proprietà (slot) possono avere diverse "facet": tipo dati, valori permessi, numero di valori, ecc.:

1. Cardinality: quanti valori posso avere in uno slot
2. Value type: tipo di dato
3. Domain: a quali classi si applica questo slot
4. Range: classi permesse per gli slot definibili sulle istanze.

### 5. Creare le istanze

1. L'ultimo passo di sviluppo consiste nella creazione delle istanze delle classi presenti nell'ontologia.
2. A questo punto spesso si scoprono errori di modellazione nei passi precedenti.

## Riuso di Ontologie

Conviene sempre analizzare se esistono ontologie disponibili che possano essere adottate, estese, raffinate, adattate, nel nostro dominio e obiettivo particolare.

Alcuni esempi:

- Ontolingua ontology library <http://www.ksl.stanford.edu/software/ontolingua>
- DAML ontology library <http://www.daml.org/ontologies>
- Ontologie OWL <http://protege.stanford.edu/plugins/owl/ontologies.html>
- Sito OWL <http://www.w3.org/2004/OWL/>

Nel campo dell'e-learning ci sono riferimenti specifici:

[http://protege.stanford.edu/ontologies/dublincore/dublin\\_core.html](http://protege.stanford.edu/ontologies/dublincore/dublin_core.html)

Una ontologia che descrive i metadati del [Dublin Core](#) (che sono anche una parte dei metadati di IEEE LOM e SCORM)

“Learner” <http://www.learninglab.de/~dolog/learnerrdfbindings/> Una ontologia che descrive le caratteristiche di un learner create nell’ambito del progetto [Elena](#). Descrive vari aspetti del profilo studente in accordo con standard LSTC IEEE e IMS.

Lo scopo è quello di realizzare sistemi adattabili al profilo dello studente.

Ci sono vari linguaggi per la rappresentazione di ontologie.

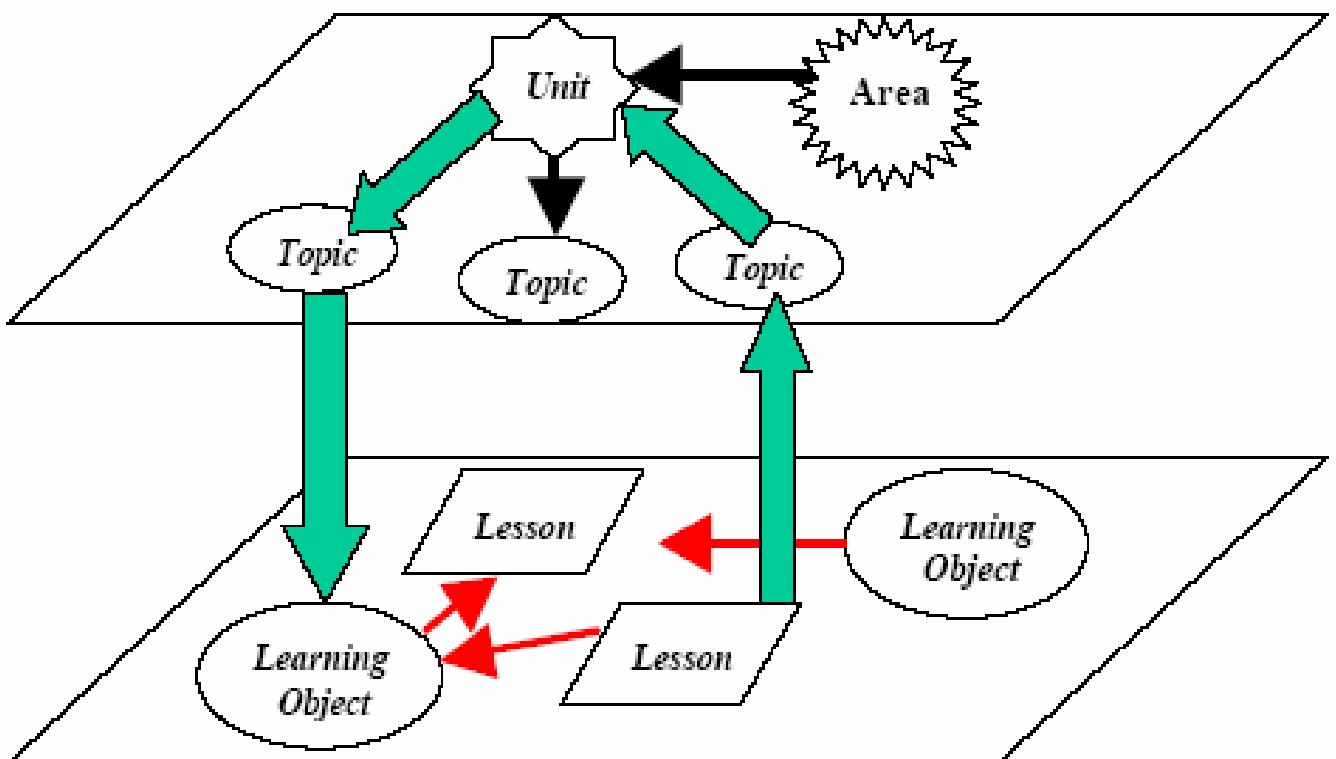
OWL Ontology Web Language è un linguaggio per la descrizione di ontologie proposto dal consorzio W3C.

Per un tutorial su OWL: [ISWC03 tutorial on OWL](#) by Sean Bechhofer, Ian Horrocks and Peter Patel-Schneider.

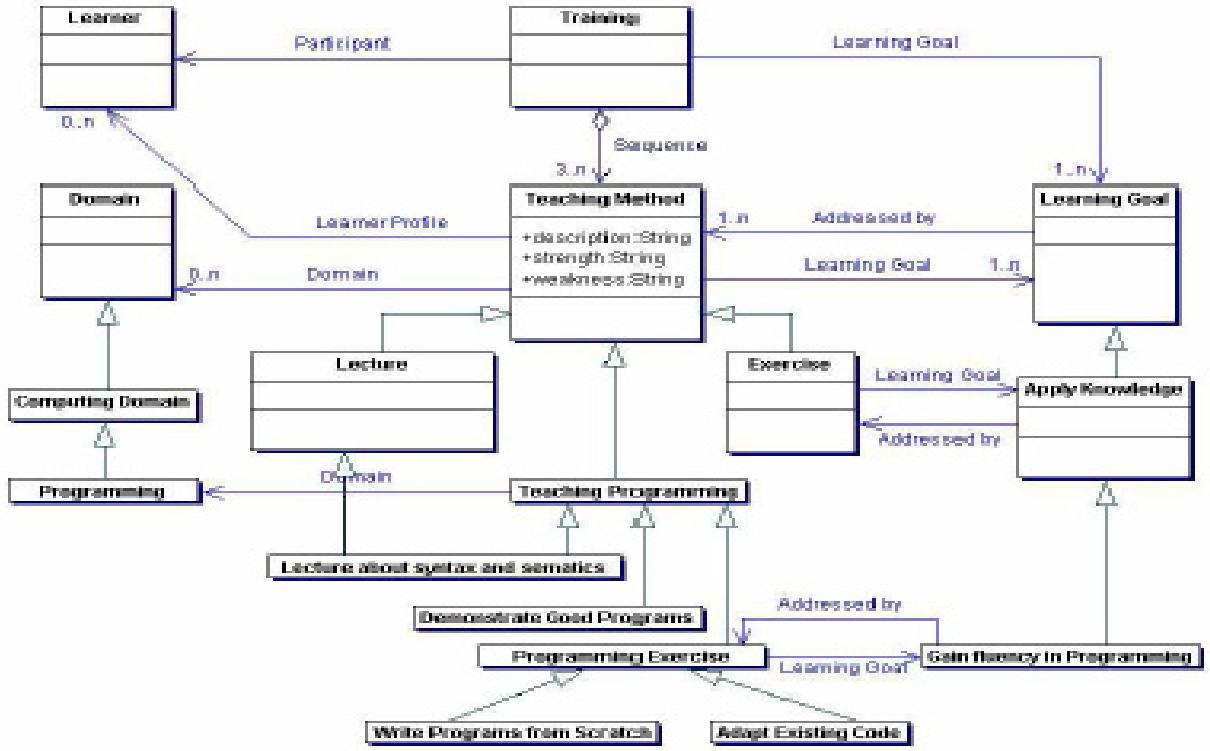
## Ontologie ed e-learning

- Descrivere un dominio di conoscenza
  - Argomenti di una certa materia e loro relazioni (prerequisiti etc).
- Descrivere un processo di apprendimento
  - Le varie attività: lezioni, esercitazioni, verifiche
  - Descrivere i vari ruoli: studenti, tutor...
  - Descrivere situazioni es: lavoro collaborativo

Esempio. Possiamo costruire una ontologia (Unità e topic della figura) che organizzi un dominio di conoscenza e annotare degli learning Objects o lezioni rispetto a questo dominio.



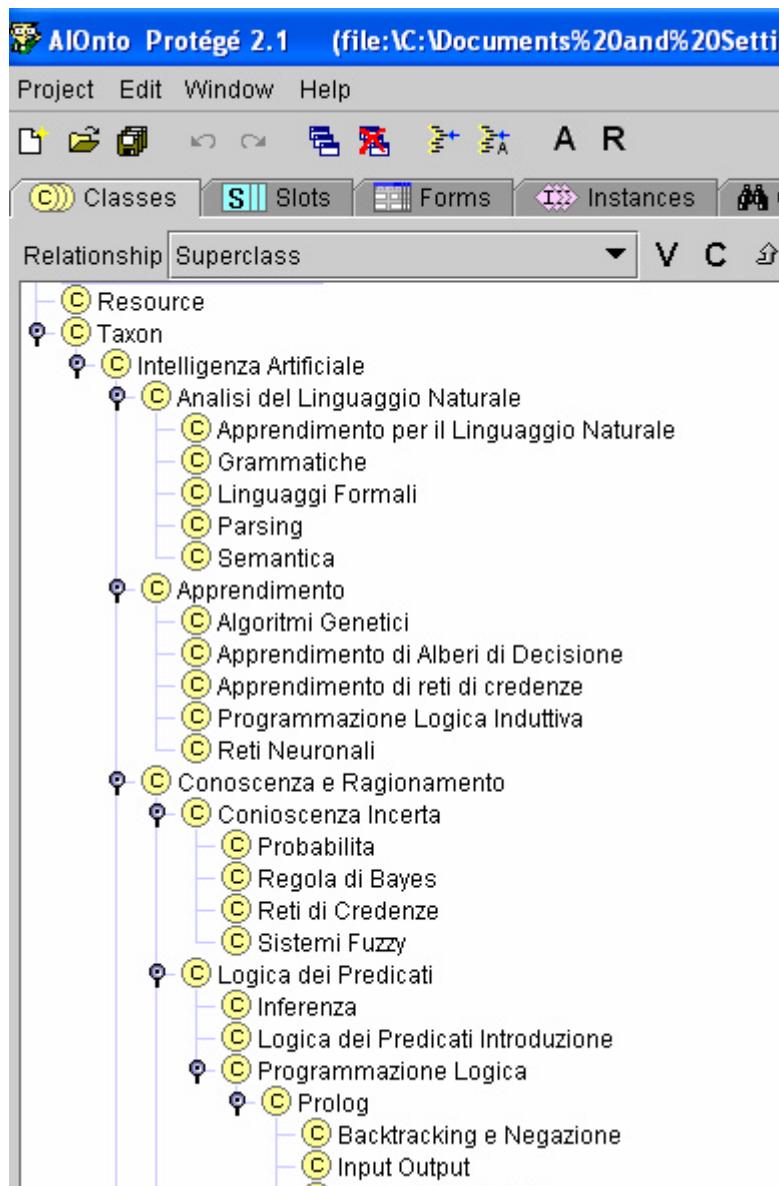
Esempio Ontologia a supporto di attività didattiche. (Qui l'ontologia viene rappresentata graficamente usando UML, Unified Modelling Language, che viene usato in vari ambiti quale la progettazione di programmi orientati agli oggetti)



Alcuni esempi.

(Questi esempi e quello dei libri sono contenuti in EsempiOntologie.zip)

Una tassonomia per una materia (in questo caso l'Intelligenza Artificiale). Le varie sottoclassi corrispondono a vari argomenti. In questo caso la relazione fra classi e sottoclassi corrisponde a è\_parte\_di. Le classi non hanno proprietà associate.



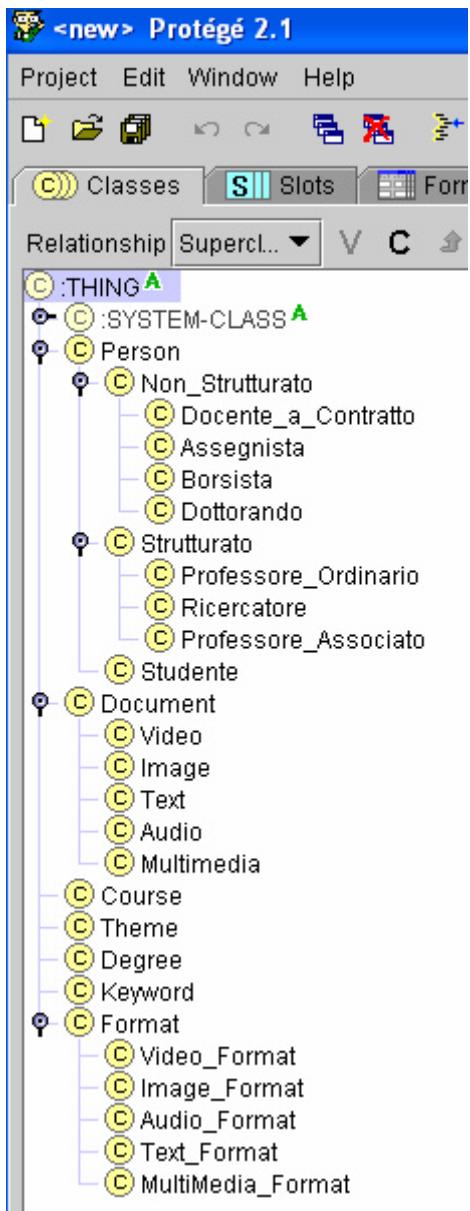
Si è poi creata una classe Resource che descrive delle risorse didattiche con alcune proprietà (metadati) quali Title, Requires (Prerequisiti), etc.

**C Resource (type=:STANDARD-CLASS)**

Name	Documentation		
Resource	Una risorsa didattica: un Learning Object.		
Role	Concrete		
Template Slots			
Name	Type	Cardinality	
S InteractivityType	String	multiple	
S Requires	Instance	multiple	cla
S Subject	Instance	multiple	cla
S Title	String	multiple	
S TypicalLearningTime	String	multiple	

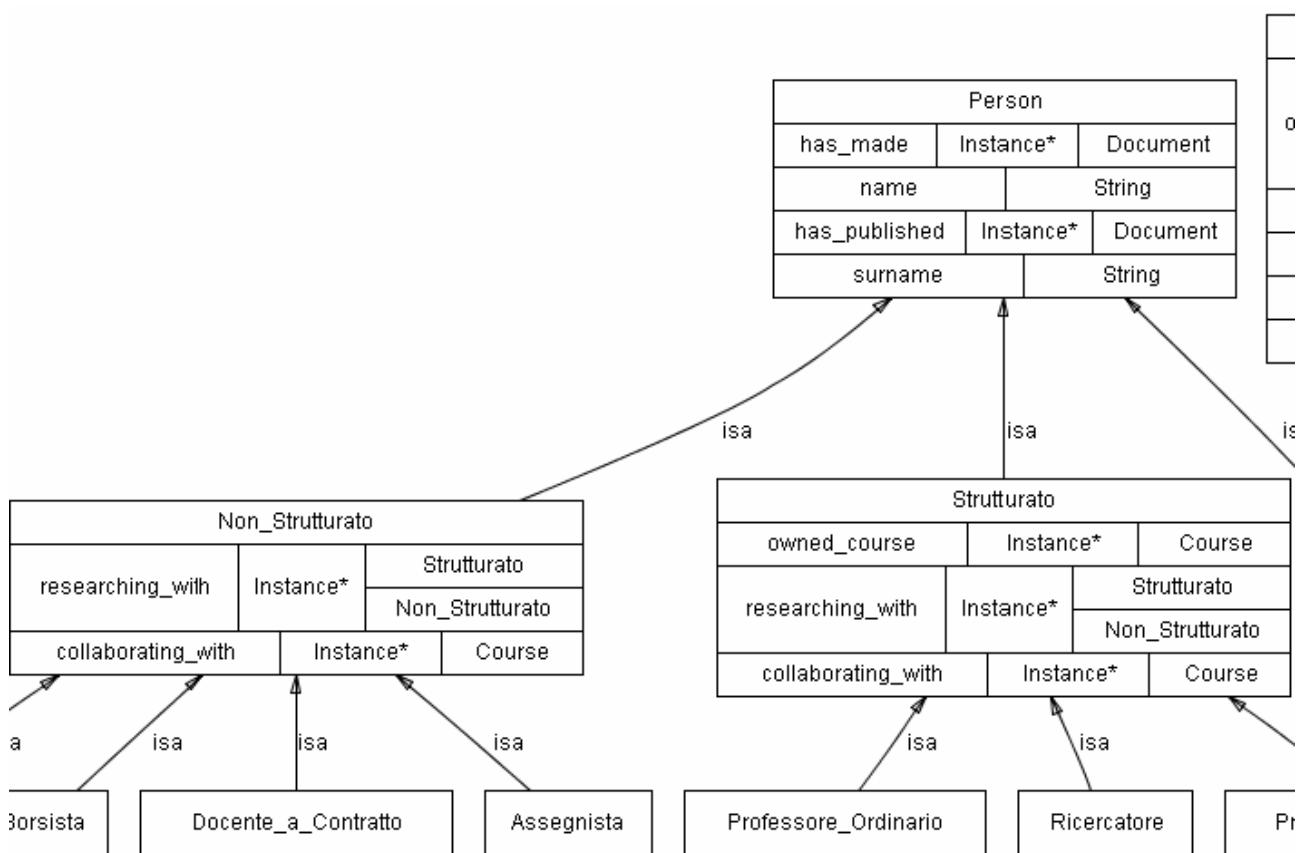
Potete vedere i dettagli nei file AiOnto.rdf e AiOnto.rdfs

Un secondo esempio riguarda un archivio di dati multimediali che possono essere utilizzati come parte di un corso di Sistemi Operativi.

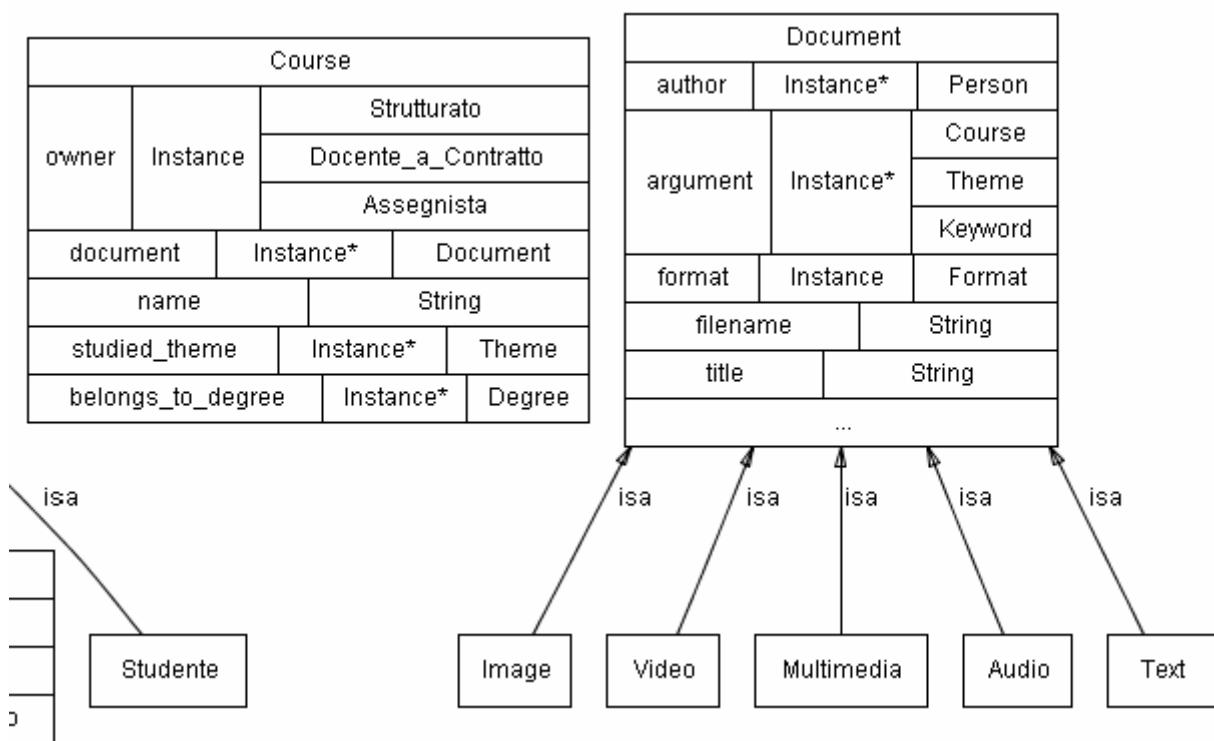


In questo caso vengono descritte le persone coinvolte (insegnanti e studenti), i corsi, i materiali multimediali, i corsi, la tassonomia della materia.

Per quanto riguarda le persone sono descritti vari tipi di persone che possono essere autori o publisher di risorse. Il personale strutturato o non strutturato collabora con qualcuno e ricerca con qualcuno. Il personale strutturato tiene dei corsi.



Sono descritti i corsi con i docenti e documenti che possono essere collegati.  
Per quanto riguarda i materiali multimediali possono essere testi, audio, video, materiali composti (come quelli descritti da file SMILE Video, audio, testo sincronizzati).



Infine vengono definite delle password per rappresentare il documento.

La tassonomia della materia è rappresentata dalla classe thema. I temi specifici sono rappresentati da istanze di tema legati dalla proprietà father (o la sua inversa son) che indica qual'è l'argomento più generale che lo contiene (o quali sono i sottoargomenti).

I documenti sono anche descritti da parole chiave.

Lo scopo di questa ontologia è quella di permettere di cercare documenti secondo vari criteri: per argomento, per corso, per autore, per parole chiave. La ricerca può essere resa flessibile percorrendo l'ontologia. Se cerco un certo argomento (theme) posso anche cercare argomenti fratelli nella tassonomia (che penso essere simili).

Se cerco un documento di un certo autore posso anche vedere i documenti descritti da persone che collaborano o fanno ricerca con lui.

Theme		
document	Instance*	Document
course	Instance*	Course
father	Instance*	Theme
son	Instance*	Theme
description	String*	

Degree

Keyword		
document	Instance*	Document
key	String	

## Uso di Protege e Sviluppo di Ontologie

Il documento [http://protege.stanford.edu/publications/ontology\\_development/ontology101.html](http://protege.stanford.edu/publications/ontology_development/ontology101.html) contiene una introduzione alle problematiche connesse allo sviluppo di ontologie. Illustra lo sviluppo di una ontologia relativa ai vini francesi.

Per le prime istruzioni sull'uso di Protege potete guardare  
<http://protege.stanford.edu/publications/GettingStarted.pdf>

Se volete aprire un file OWL fate Project/Import to Standard Interface/OWL File  
Quindi dare il nome del file OWL che contiene sia classi che istanze.

Per aprire i file RDF e RDFS nel pannello iniziale dire che si vuole un progetto RDF schema. Dare il comando Build. Dare quindi i nomi dei due file RDFS per le classi e RDF per le istanze. Potete lasciare il default di Protege per il namespace.

