

# Introduzione alle Ontologie

---

di Valeriano Sandrucci

# Ontologie

---

## PARTE I

# Definizioni

---

- In Filosofia: “L'**Ontologia**, branca fondamentale della metafisica, è lo studio dell'essere in quanto tale, nonché delle sue categorie fondamentali”
- In informatica: “L'**Ontologia** è una esplicita specificazione di una concettualizzazione”
- **N.B.** Con lo stesso nome ci si riferisce a cose diverse

# Definizioni

---

- **Ontologia:** insieme di tecnologie per la modellazione di conoscenza
- **Modello Ontologico:** modello che definisce in modo esplicito le relazioni e la semantica delle entità di un dominio.
- **Linguaggio Ontologico:** linguaggio per la descrizione di un modello ontologico

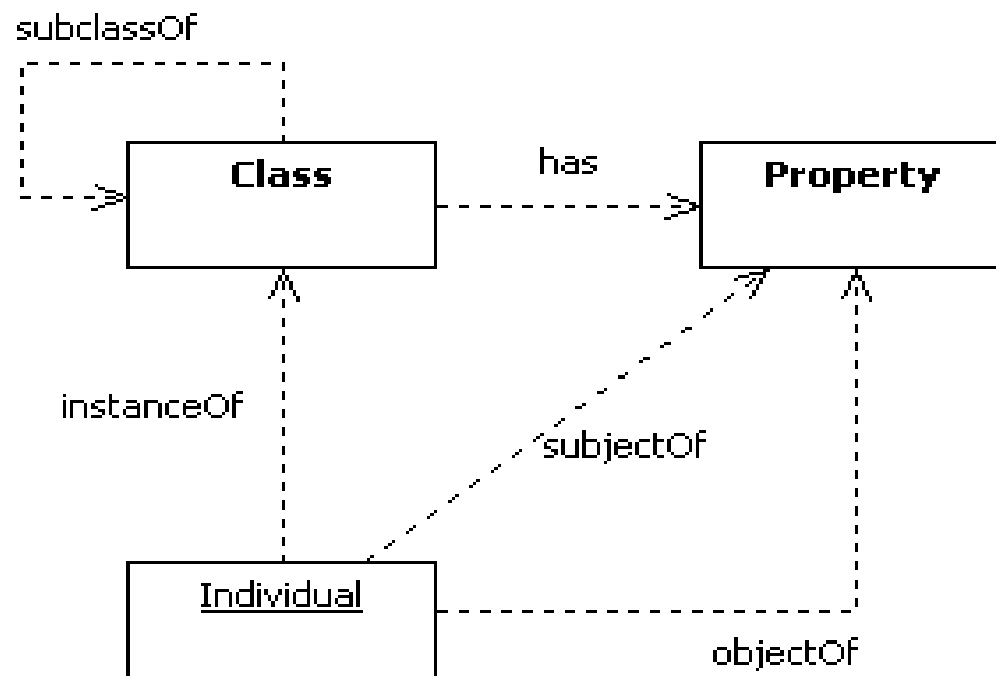
# Chi contribuisce alle Ontologie

---

- **Intelligenza Artificiale:** espressività linguaggi di modellazione, meccanismi di inferenza
- **Telematica:** URI, XML, carattere distribuito
- **Ingegneria del software:** architetture software, tecniche di progettazione
- **Altre fonti:** Linguistica Computazionale, Informatica Teorica, Matematica...

# Modello Ontologico

---



- Un modello Ontologico:
  - Si compone di: **Classi**, **Attributi**, **Relazioni** ed **Individui**
  - E' simile ad un modello ad oggetti

# Modello Ontologico

---

- **Classi:** insiemi, collezioni o tipi di oggetti
- **Attributi:** proprietà, caratteristiche o parametri che gli oggetti possono avere e condividere (Literals)
- **Relazioni:** modi in cui gli oggetti possono essere messi in relazione gli uni con gli altri
- **Individui:** istanze del modello, sono gli elementi di base (“ground level objects”)

# Termini ricorrenti

---

- **Sussunzione:** is-superclass-of. La relazione di sussunzione è usata per creare una gerarchia, o un grafo aciclico, di classi.
- **Meronimia:** part-of. Relazione che serve a descrivere come i concetti si combinano insieme per formarne di nuovi.
- **Vocabolario Controllato:** lista di termini esplicitamente enumerati. Tutti i termini devono avere un'unica definizione e non essere ambigui.
- **Tassonomia:** è una collezione di termini di un vocabolario controllato organizzati in una struttura gerarchica attraverso relazioni di specializzazione.
- **Thesaurus:** è una rete di termini di un vocabolario controllato. Questo significa che un thesaurus adopera relazioni associative in aggiunta a quelle di specializzazione.



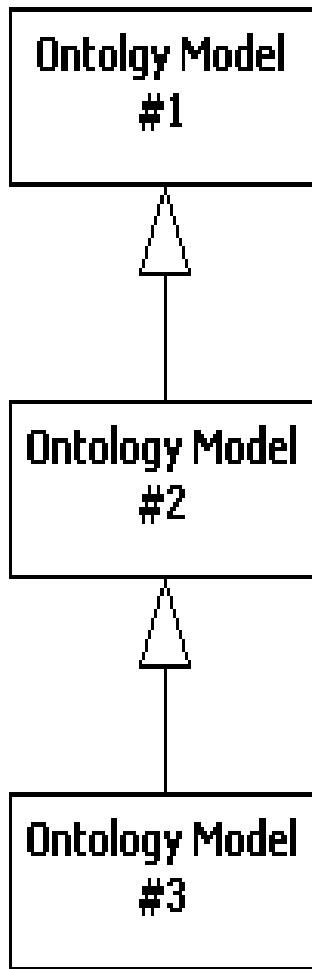
# Operazioni Tipiche

---

- Costruire un modello ontologico estendendone uno esistente
- Eseguire il mapping tra i concetti di un modello ontologico con quelli di un altro
- Unire in un unico modello le informazioni provenienti da sorgenti diverse

# Tipi di Ontologie

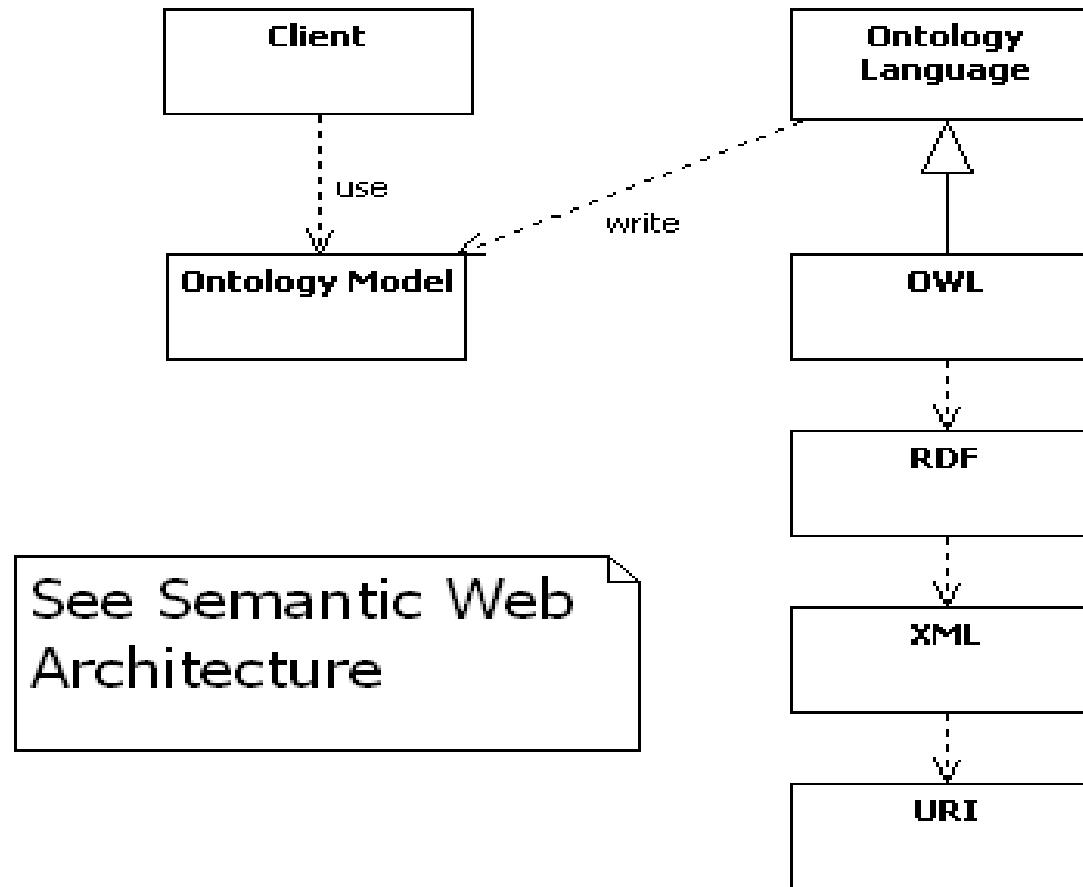
---



- **Foundation Ontology:** modello ontologico relativo a quei concetti che si possono applicare a domini diversi (es. Dublin Core, OpenCyc, DOLCE)
- **Domain Ontology:** modello ontologico che descrive i concetti di un dominio specifico
- **Specialized Ontology:** estensione di un ontologia di dominio con scopi molto specifici

# Vista Componenti

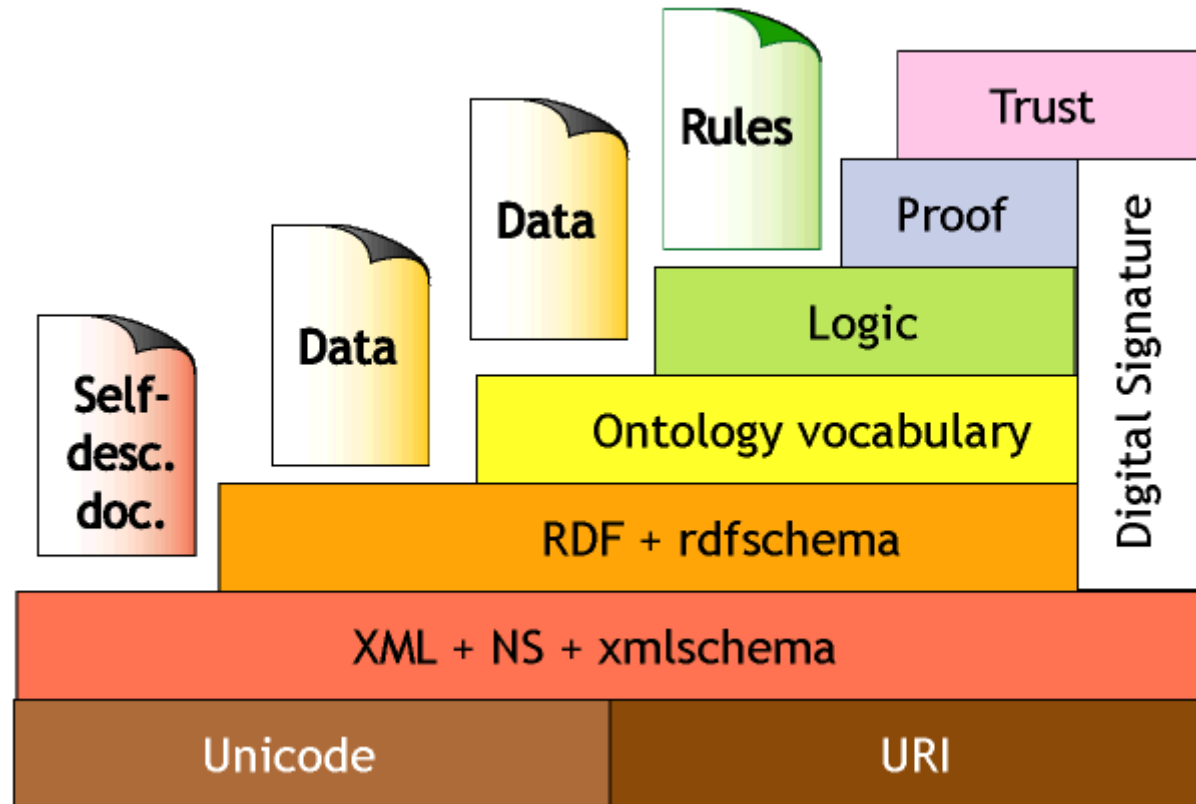
---



- **Semantic Web:** evoluzione del web attuale

# Semantic Web Architecture

---



# URI

---

- **URI:** Universal Resource Identifier
  - Gli URI si dividono in: URL ed URN
- **URL:** Universal Resource Locator
  - es.: <http://www.ing.unifi.it/didattica.htm>
- **URN:** Universal Resource Name
  - es.: <urn:isbn:0451450523>

**N.B.** La sintassi degli URI è definita da RFC 3986

# XML

---

- XML sta per eXtensible Markup Language
- Viene gestito dal W3C
- Nasce nel 1998 da SGML
- E' un linguaggio di markup estendibile: si possono definire nuovi tag.
  - **es.** `<saluto>Hello World!</saluto>`

# Tecnologie legate a XML

---

- **DTD (*Document Type Definition*)**: linguaggio che specifica le caratteristiche del tipo di documento attraverso una serie di "regole grammaticali".
- **XML Schema**: come la DTD, serve a definire la *grammatica* del documento XML.
- **XLink**: serve a collegare in modo completo due documenti XML; permette di creare link multi-direzionali e semanticamente avanzati.
- **XSL (*eXtensible Stylesheet Language*)**: è il linguaggio con cui si descrive il foglio di stile di un documento XML. La sua versione estesa è l'XSLT (la T sta per *Trasformations*).

# Tecnologie legate a XML

---

- **XPath:** è una tecnologia legata ad XSL ed XSLT. Serve per recuperare e gestire i nodi di un documento XML.
- **XPointer:** serve ad identificare univocamente precise porzioni di un documento XML.
- **XQuery:** è un linguaggio di query concepito per essere applicabile a qualsiasi sorta di documento XML, e si basa sull'utilizzo di XPath per la specificazione di percorsi all'interno di documenti.
- **Altro:** SAX, DOM, SVG,...



# RDF

---

- RDF sta per Resource Description Framework
- Viene gestito dal W3C
- RDF è un framework per la descrizione della conoscenza
- Un modello RDF può essere serializzato in XML

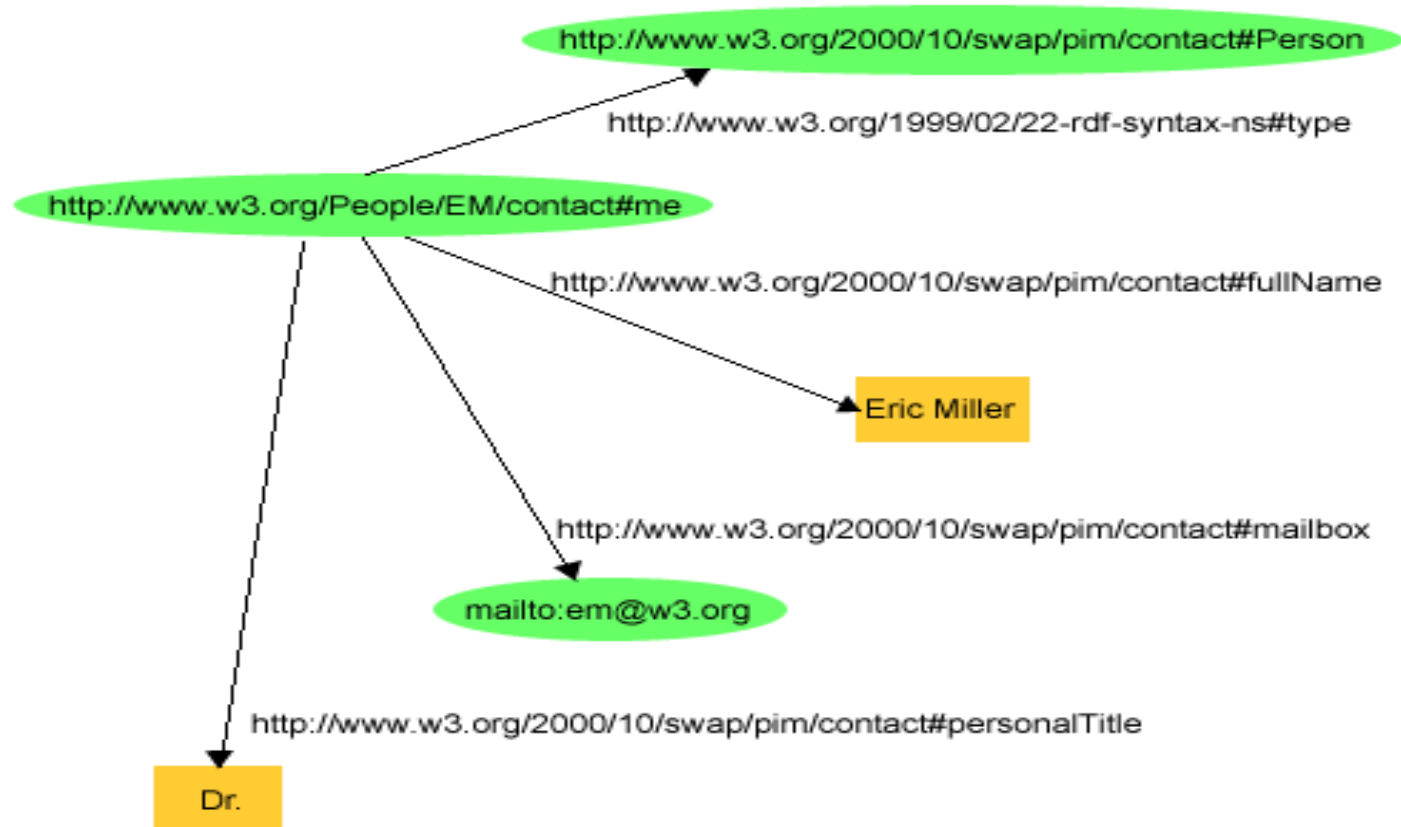
# RDF

---

- Qualunque cosa descritta da RDF è detta risorsa
  - Ogni risorsa è identificata da un URI
- L'unità base per rappresentare un'informazione in RDF è lo statement.
  - Uno statement è una tripla del tipo: Soggetto – Predicato – Oggetto
  - Soggetto, Predicato ed Oggetto sono risorse
  - Oggetto può essere anche un tipo primitivo

# RDF

---



- Un modello RDF è rappresentabile con un grafo orientato i cui nodi sono risorse o tipi primitivi ed i cui archi rappresentano proprietà

# RDF

---

- RDF definisce un modello semplice per descrivere le relazioni tra le risorse. Ma non fornisce alcuno strumento per attribuire semantica alle relazioni.
- **Esempio di serializzazione XML:**

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:au="http://description.org/schema/">
  <rdf:Description
    about="http://www.book.it/Promessi_Sposi">
    <au:author>Alessandro_Manzoni</au:author>
  </rdf:Description>
</rdf:RDF>
```

# OWL

---

- OWL sta per Ontology Web Language
- Viene gestito dal W3C
- OWL è un linguaggio per la descrizione di modelli ontologici. Esso fornisce tre sotto-linguaggi dal potere espressivo via via crescente: OWL Lite, OWL DL, OWL Full
- Attribuire semantica ad un modello RDF

# OWL

---

- **Class (Thing, Nothing)**

```
<owl:Class rdf:ID="Winery"/>
```

```
<owl:Class rdf:ID="Region"/>
```

```
<owl:Class rdf:ID="ConsumableThing"/>
```

- **rdfs:subClassOf**

```
<owl:Class rdf:ID="Wine">
```

```
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
```

```
  <rdfs:label xml:lang="en">wine</rdfs:label>
```

```
  <rdfs:label xml:lang="fr">vin</rdfs:label>
```

```
  ...
```

```
</owl:Class>
```

# OWL

---

- **rdf:Property, rdfs:subPropertyOf, rdfs:domain, rdfs:range**

```
<owl:ObjectProperty rdf:ID="madeFromGrape">  
  <rdfs:domain rdf:resource="#Wine"/>  
  <rdfs:range rdf:resource="#WineGrape"/>  
</owl:ObjectProperty>
```

- **Individual**

```
<Wine rdf:ID="MikesFavoriteWine">
```

- **equivalentClass, disjointWith**

```
<owl:Class rdf:ID="Wine"/>  
  <owl:equivalentClass rdf:resource="&vin;Wine"/>  
</owl:Class>
```

# OWL

---

- **equivalentProperty**
- **SameAs, differentFrom**

```
<Wine rdf:ID="MikesFavoriteWine">  
  <owl:sameAs rdf:resource="#StGenevieveTexasWhite" />  
</Wine>
```

- **AllDifferent, distinctMembers**

```
<owl:AllDifferent>  
  <owl:distinctMembers rdf:parseType="Collection">  
    <vin:WineColor rdf:about="#Red" />  
    <vin:WineColor rdf:about="#White" />  
    <vin:WineColor rdf:about="#Rose" />  
  </owl:distinctMembers>  
</owl:AllDifferent>
```



# OWL

---

- **Restriction, onProperty, allValuesFrom, someValuesFrom**

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

# OWL

---

- **MinCardinality, maxCardinality, cardinality**

```
<owl:Class rdf:ID="Vintage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasVintageYear"/>
      <owl:cardinality rdf:datatype="&xsd;
nonNegativeInteger"> 1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

# OWL

---

- **oneOf**

```
<owl:Class rdf:ID="WineColor">
  <rdfs:subClassOf rdf:resource="#WineDescriptor"/>
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#White"/>
    <owl:Thing rdf:about="#Rose"/>
    <owl:Thing rdf:about="#Red"/>
  </owl:oneOf>
</owl:Class>
```

# OWL

---

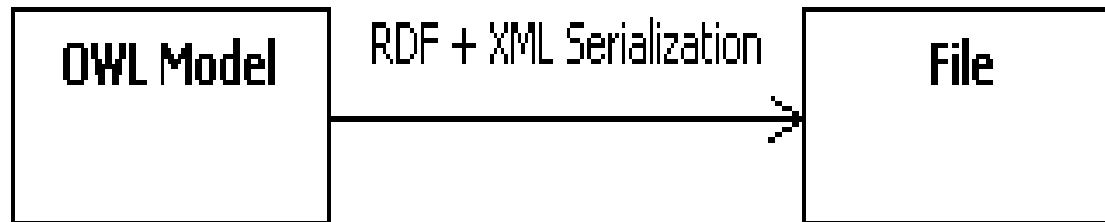
- **unionOf, complementOf, intersectionOf**

```
<owl:Class rdf:ID="Fruit">  
  <owl:unionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#SweetFruit" />  
    <owl:Class rdf:about="#NonSweetFruit" />  
  </owl:unionOf>  
</owl:Class>
```

- **hasValue**

# Archiviazione di OWL

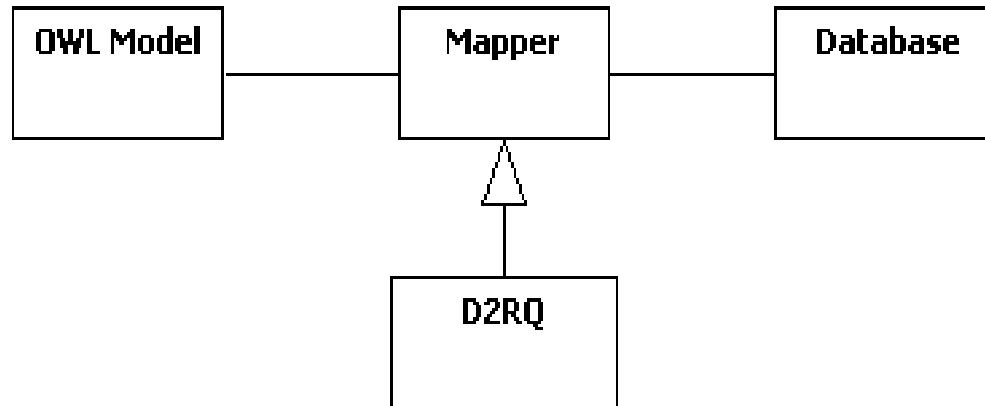
---



- Archiviazione in un file
- Ogni modello ontologico può essere descritto con un sufficiente numero di triple RDF che serializzate in XML possono venire archiviate in un file.
- Tale soluzione ha il pregio di essere semplice e coerente con la descrizione del Semantic Web attraverso una pila di strumenti che ricorda una pila protocollare.

# Archiviazione di OWL

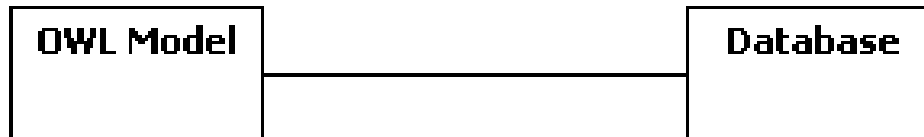
---



- Archiviazione in un database relazionale.
- Se le informazioni si trovano in una base di dati relazionale è possibile costruire delle classi di mapping che mostrino l'informazione in forma ontologica.
- D2RQ è una libreria Java che rende semplice l'operazione di mapping

# Archiviazione di OWL

---



- Archiviazione in un contenitore RDF.
- L'archiviazione di un modello ontologico in un file è interessante ma può risultare inefficiente. E' possibile archiviare le triple RDF in un database relazionale esplicitamente progettato per questo scopo.
- Jena e Sesame sono due librerie Java che permettono la gestione di un contenitore RDF in un database relazionale.

# Protégé

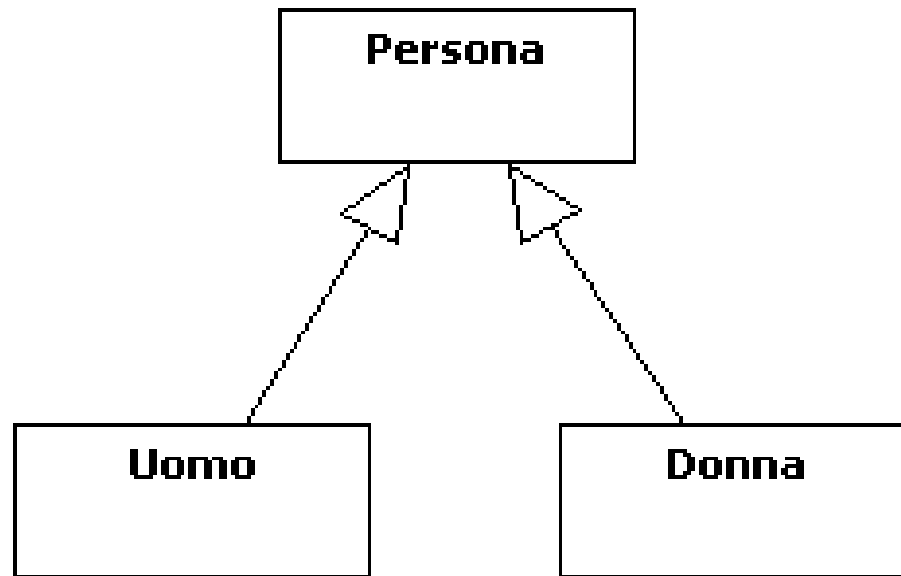
---

- Editor per OWL
- Gestito dall'Università di Stanford
- Open-source
- Organizzato a Plug-in



# Rappresentare OWL con UML

---



- La relazione di ereditarietà di OWL si modella in maniera naturale con UML
- Notare che le classi di OWL non possono contenere metodi

# Rappresentare OWL con UML

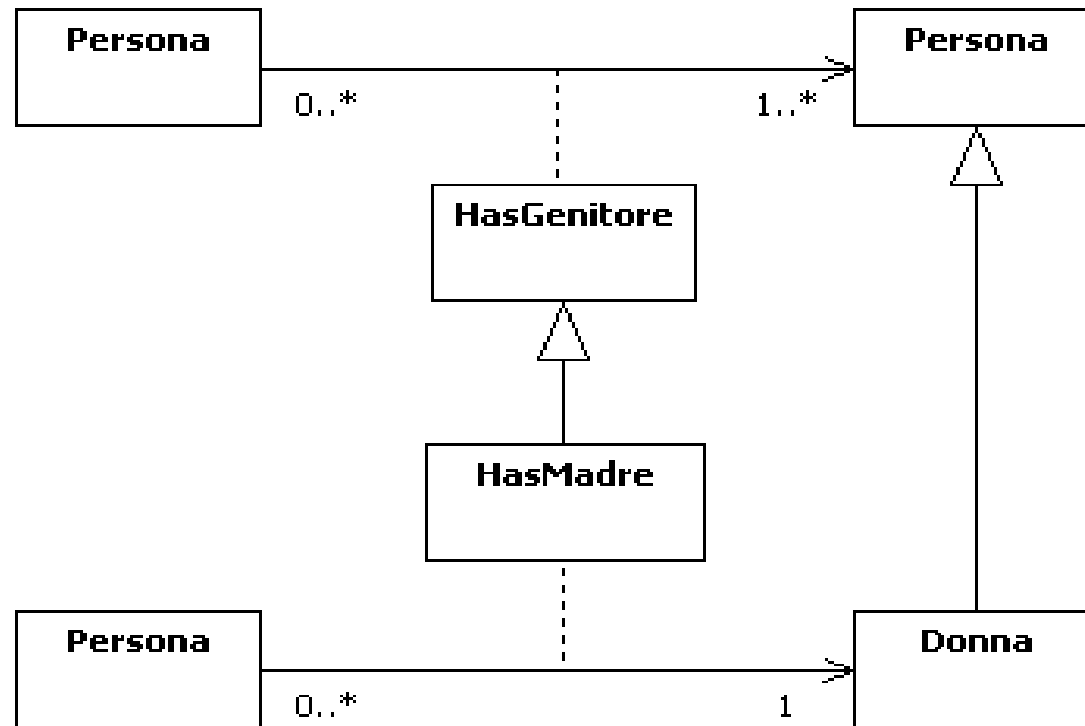
---



- Per rappresentare la relazione HasGenitore esistente tra due persone di OWL posso adoperare un link di associazione
- Se si adoperano associazioni orientate le relazioni sono più chiare

# Rappresentare OWL con UML

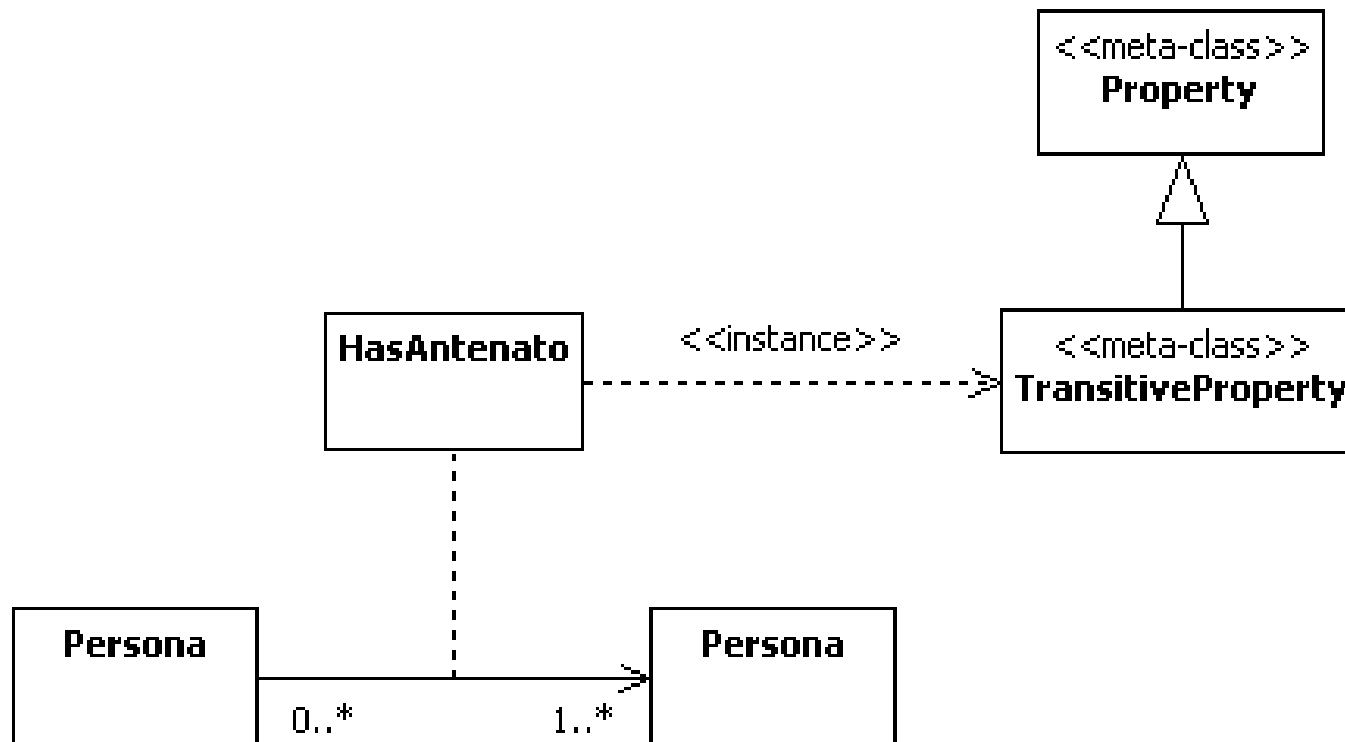
---



- Per evidenziare relazioni tra proprietà di OWL si ricorre alle classi di associazione

# Rappresentare OWL con UML

---



- Le classi e le proprietà del modello OWL sono in realtà istanze del meta-modello stabilito da OWL stesso

# Rappresentare OWL con UML

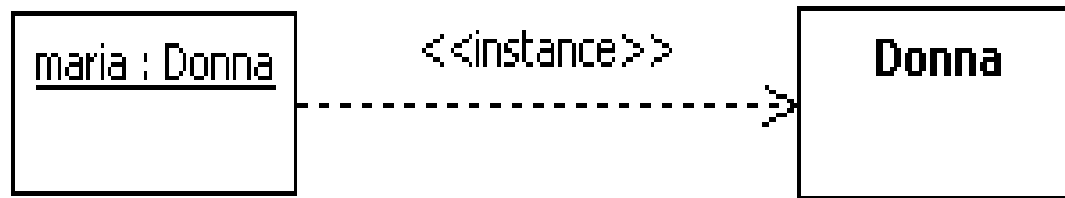
---



- I costrutti di OWL: disjoint, intersectionOf, unionOf, complementOf, equivalentTo, etc... posso essere rappresentati con vincoli di UML
- In modo equivalente ci si può comportare con i costrutti: sameAs, differentForm, etc... relativi alle istanze

# Rappresentare OWL con UML

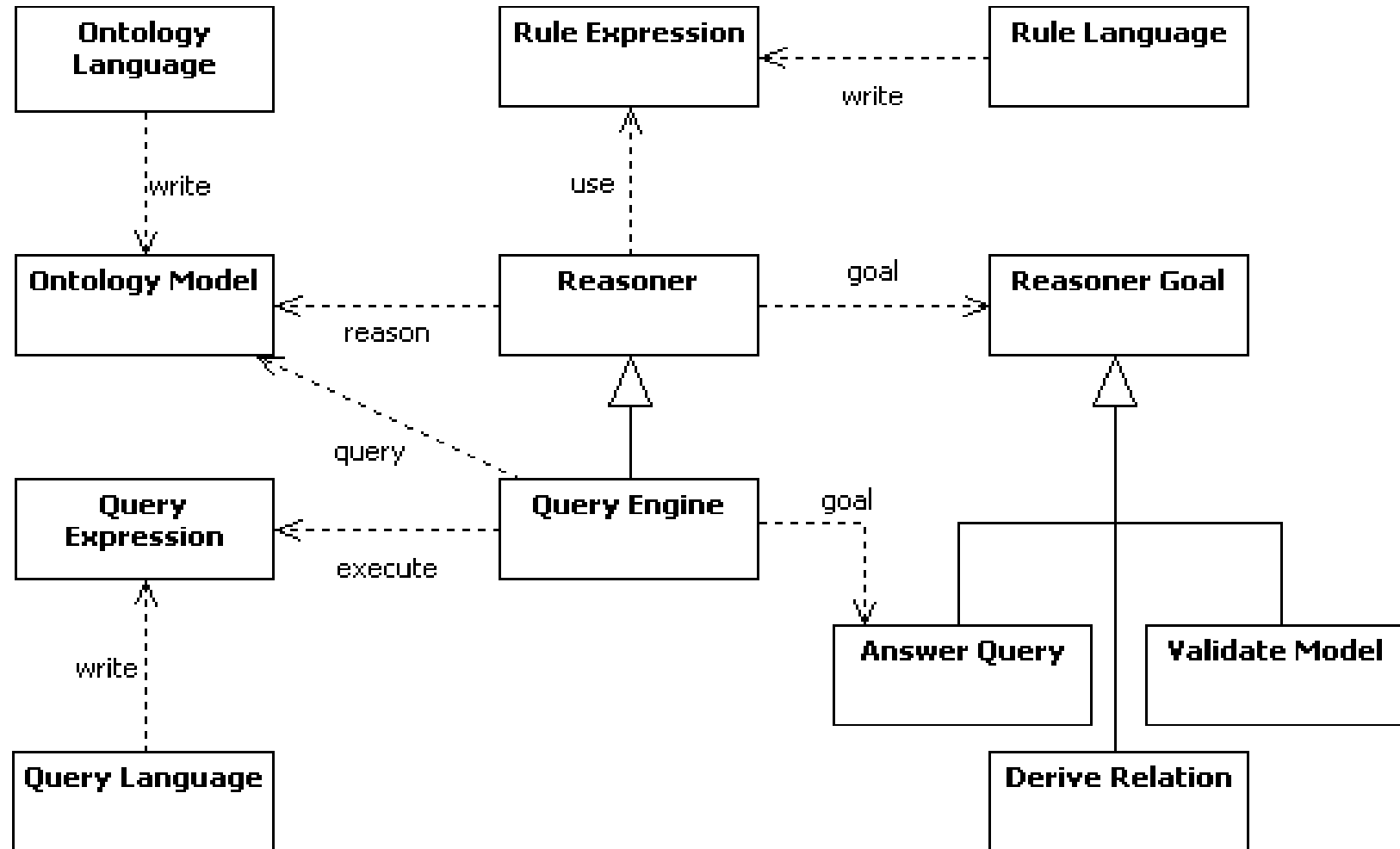
---



- Per mostrare il legame tra oggetti e classi che li definiscono si possono usare link di dipendenza
- Notare che un oggetto OWL può essere istanza di più classi contemporaneamente

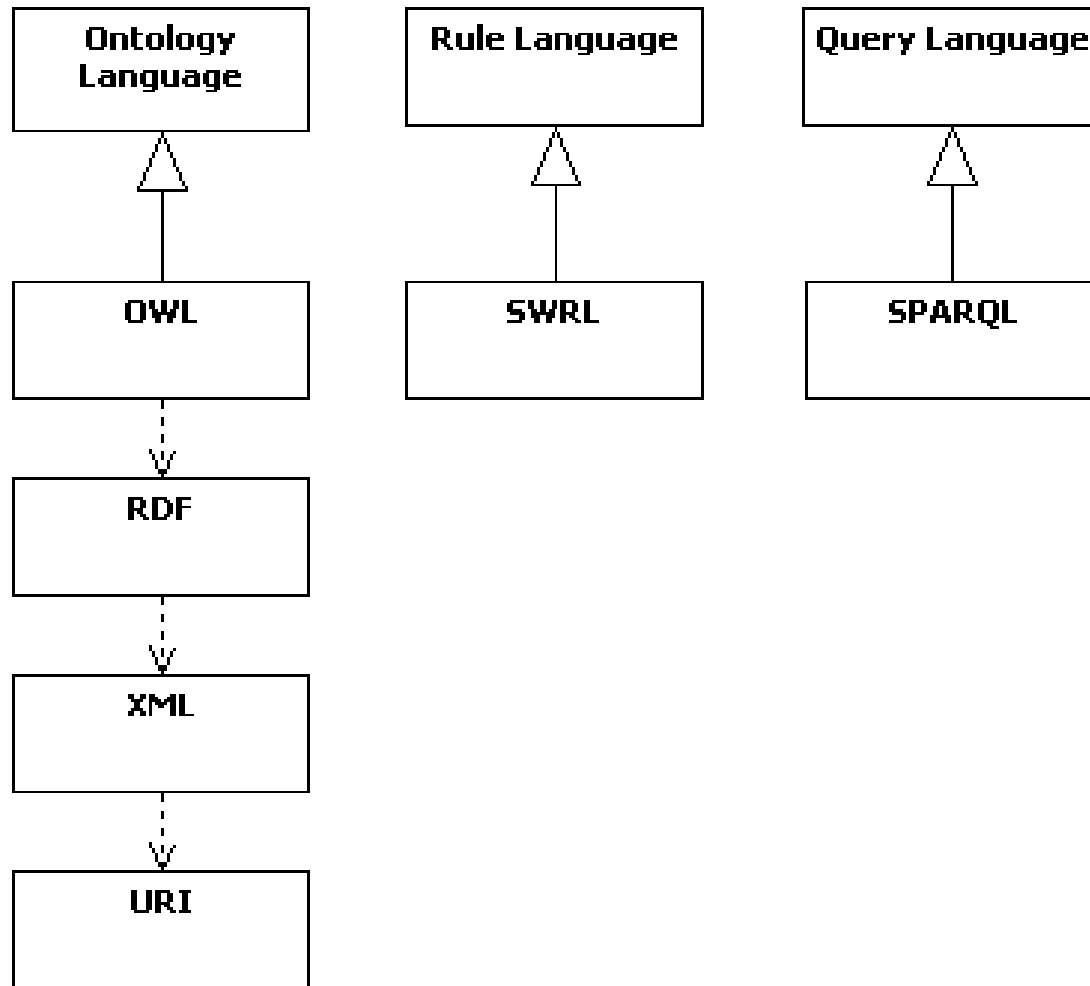
# Vista Componenti

---



# Vista Componenti

---





# SPARQL

---

- SPARQL è un Query Language for RDF
- Viene gestito dal W3C
- E' un linguaggio di interrogazione che ricorda vagamente SQL
  - Sono parole chiave del linguaggio: SELECT, WHERE, ORDER BY, FILTER, UNION, etc...

# SPARQL

---

- **Esempio n. 1**

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?book ?title
WHERE { ?book dc:title ?title }
```

- **Esempio n. 2**

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price.
        FILTER (?price < 30.5) .
        ?x dc:title ?title. }
```

# SWRL

---

- SWRL sta per Semantic Web Rule Language
- Viene gestito dal W3C
- Serve per definire regole di inferenza
- Le espressioni SWRL sono adoperate dai Reasoner per fare inferenza su modelli OWL
- Le espressioni SWRL sono archiviate all'interno di modelli OWL

# SWRL

---

- Per SWRL viene definita una sintassi astratta ed una sintassi in OWL
- Sono regole SWRL espressioni del tipo (gli esempi non usano la sintassi formale):
  - `hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) => hasUncle(?x1,?x3)`
  - `Artist(?x) & artistStyle(?x,?y) & Style(?y) & creator(?z,?x) => style/period(?z,?y)`

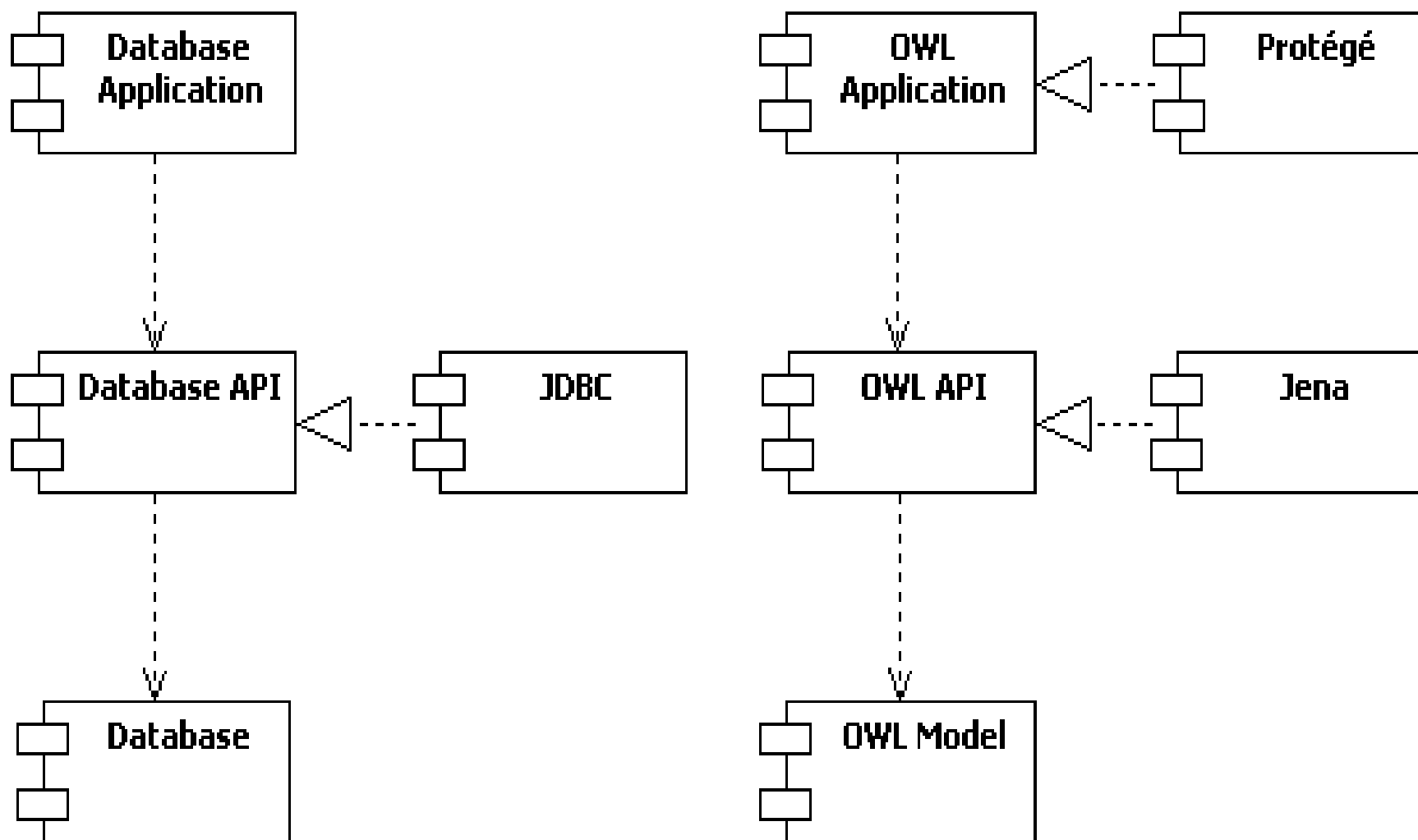
# Riassumendo

---

- OWL è un linguaggio che serve per modellare conoscenza. Su un modello ontologico si possono eseguire operazioni quali:
  - Validazione
  - Interrogazione
  - Deduzione di nuova conoscenza
- OWL non è un linguaggio di programmazione
  - Non permette di definire metodi
  - Non esistono costrutti per la gestione della visibilità

# Architettura Standard

---



# Qualità positive

---

- Per nativa vocazione le ontologie permettono la rappresentazione esplicita di modelli semantici che combinano la non-ambiguità necessaria alla specifica tecnica con la comprensibilità capace di colmare la distanza tra tecnici e stakeholders
- Le ontologie abilitano l'applicazione di tecniche e strumenti on-the-shelf capaci di ragionamento finalizzato ad una varietà di obiettivi tra cui ricerca di informazione, inferenza e deduzione di nuova conoscenza, validazione di modelli

# Qualità positive

---

- Le ontologie si adattano bene ad un contesto distribuito, permettendo la creazione di modelli per riuso, composizione e riconciliazione di frammenti sviluppati in modo concorrente e distribuito
- Le ontologie hanno la capacità di modellare domini che evolvono nel tempo, superando molte complessità che invece discendono dall'uso esclusivo di tecnologie di rappresentazione convenzionale. Questo a sua volta favorisce un approccio incrementale allo sviluppo capace di accompagnare processi di evoluzione verso modelli condivisi



# Ontologie

---

## PARTE II

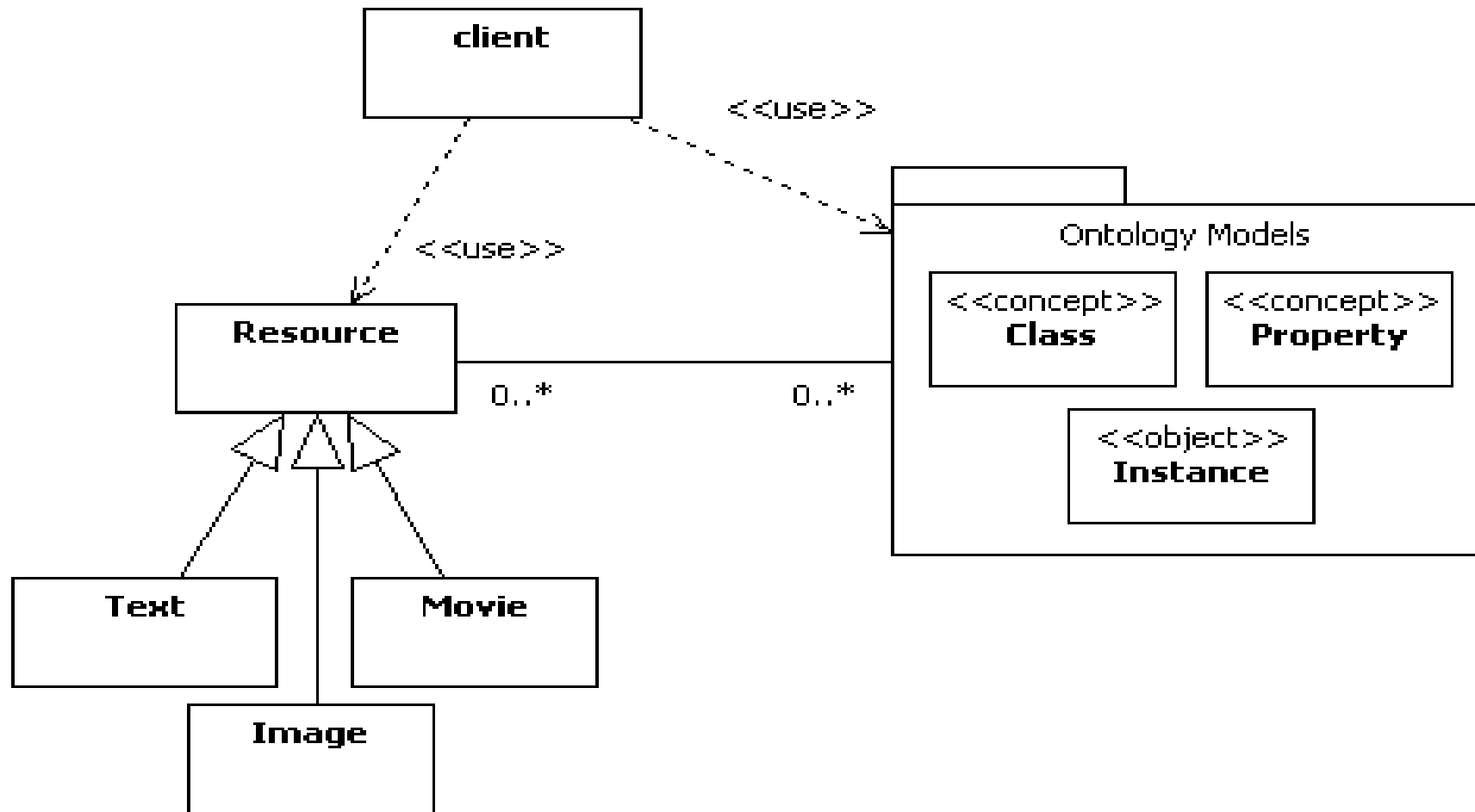
# Semantic Web

---

- Evoluzione del Web attuale
- L'informazione ha precisa caratterizzazione semantica
- Consente applicazioni oggi impensabili
  - Ricerche semantiche
  - Agenti intelligenti

# Semantic Web

---



# Per il Semantic Web occorre

---

- Modelli Ontologici Condivisi
- Risorse Annotate Semanticamente
  - Le nuove risorse potrebbero avere dati semantici al loro interno
  - Le vecchie risorse dovrebbero essere rigenerate o elaborate in qualche modo
  - Le basi di dati potrebbero essere fatte corrispondere ad equivalenti modelli ontologici attraverso opportuno mapping

# RDFa

---

- RDFa è una estensione di XHTML proposta dal W3C
- RDFa permette di annotare XHTML con informazioni semantiche.
- RDFa generalizza gli attributi degli elementi meta e link in modo da poterli adoperare su ogni elemento di un documento XHTML. Gli attributi sono:
  - **about** - URI della risorsa cui si riferiscono i metadata
  - **rel, rev, href** – relazioni con altre risorse
  - **property** – proprietà per il contenuto di un elemento
  - **content** – contenuto di un elemento, usato con property
  - **datatype** – datatype di content

# XHTML non annotato

---

```
<html>
  <head><title>Jo's Blog</title></head>
  <body>
...
  <p>
    I'm giving a talk at the XTech Conference about web widgets, on May 8th at
10am.
  </p>
...
  <p class="contactinfo">
    My name is Jo Smith. I'm a distinguished web engineer
    at
    <a href="http://example.org">
      Example.org
    </a>.
    You can contact me
    <a href="mailto:jo@example.org">
      via email
    </a>.
  </p>
...
  </body>
</html>
```

# XHTML con RDFa

---

```
<html xmlns:cal="http://www.w3.org/2002/12/cal/ical#"
      xmlns:contact="http://www.w3.org/2001/vcard-rdf/3.0#">
...
  <p role="cal:Vevent">
    I'm giving
    <meta property="cal:summary">
      a talk at the XTech Conference about web widgets
    </meta>,
    on
    <meta property="cal:dtstart" content="20060508T1000-0500">
      May 8th at 10am
    </meta>.
  </p>
...
  <p class="contactinfo" about="http://example.org/staff/jo">
    My name is
    <meta property="contact:fn">
      Jo Smith
    </meta>.
    I'm a
    <meta property="contact:title">
      distinguished web engineer
    </meta>
    at
    <a rel="contact:org" href="http://example.org">
      Example.org
    </a>.
    You can contact me
    <a rel="contact:email" href="mailto:jo@example.org">
      via email
    </a>
```

# Annotazione risorse esistenti

---

- Mapping: per sorgenti strutturate
- Tecniche di Intelligenza Artificiale e DataMining
  - Reti Neurali
  - Text Mining
  - NLP (Natural Language Processing)
- Soluzioni Ad Hoc.



# OWL-S

---

- Semantic Markup for Web Services
- OWL-S è un'ontologia OWL standardizzata
- Adoperando OWL-S posso descrivere le caratteristiche di un Web Service.
- Si possono costruire applicazioni che basano il loro funzionamento su dati OWL-S al fine di:
  - Trovare il servizio desiderato da un utente
  - Eseguire un particolare servizio rispettando il protocollo di messaggi previsto
  - Comporre servizi diversi per raggiungere un obiettivo

# Ontologie

---

## Parte III

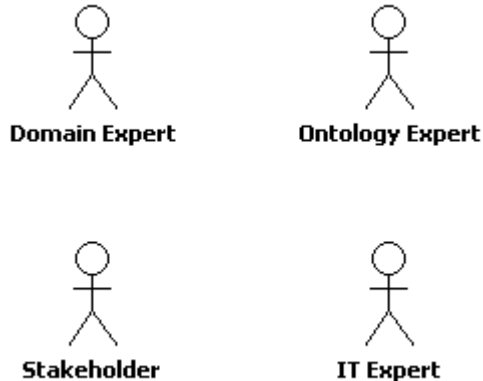
# Muddy

---

- Muddy è un'applicazione web-based progettata per consentire la fruizione e la condivisione di dati in forma ontologica.
  - Gli utenti conoscono il tipo di tecnologia adoperata per modellare i dati.
  - Gli utenti possono condividere modelli ontologici arbitrari.
- L'applicazione permette di:
  - Navigare tra le informazioni seguendo link semantici.
  - Eseguire operazioni di ricerca semantica sulla base di conoscenza.
  - Contribuire nuova conoscenza.
  - Esprimere giudizi di gradimento riguardo al materiale consultato.

# Contesto Applicativo

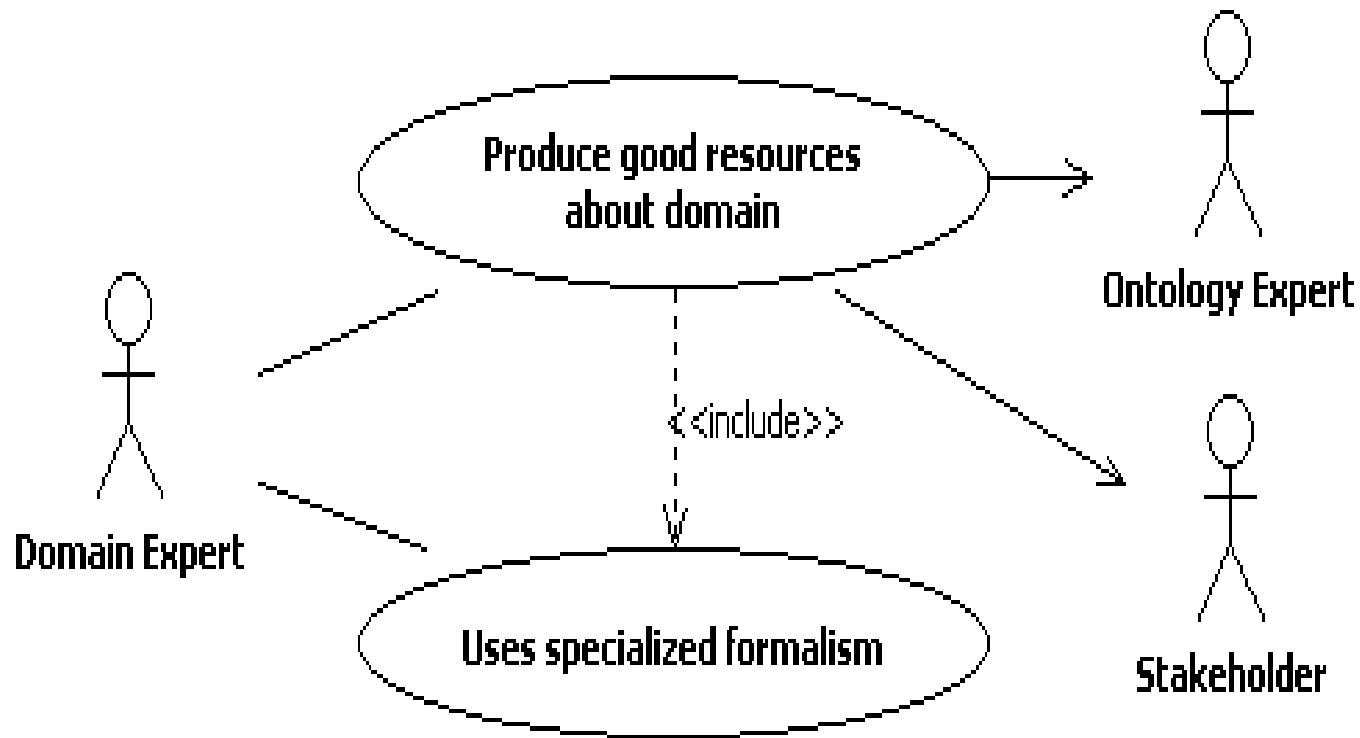
---



- **Domain Expert:** è colui il quale conosce approfonditamente l'argomento oggetto di studio
- **Ontology Expert:** è colui che conosce bene il formalismo delle ontologie
- **Stakeholder:** è una persona interessata al dominio oggetto di studio.
- **IT Expert:** il tecnico informatico.

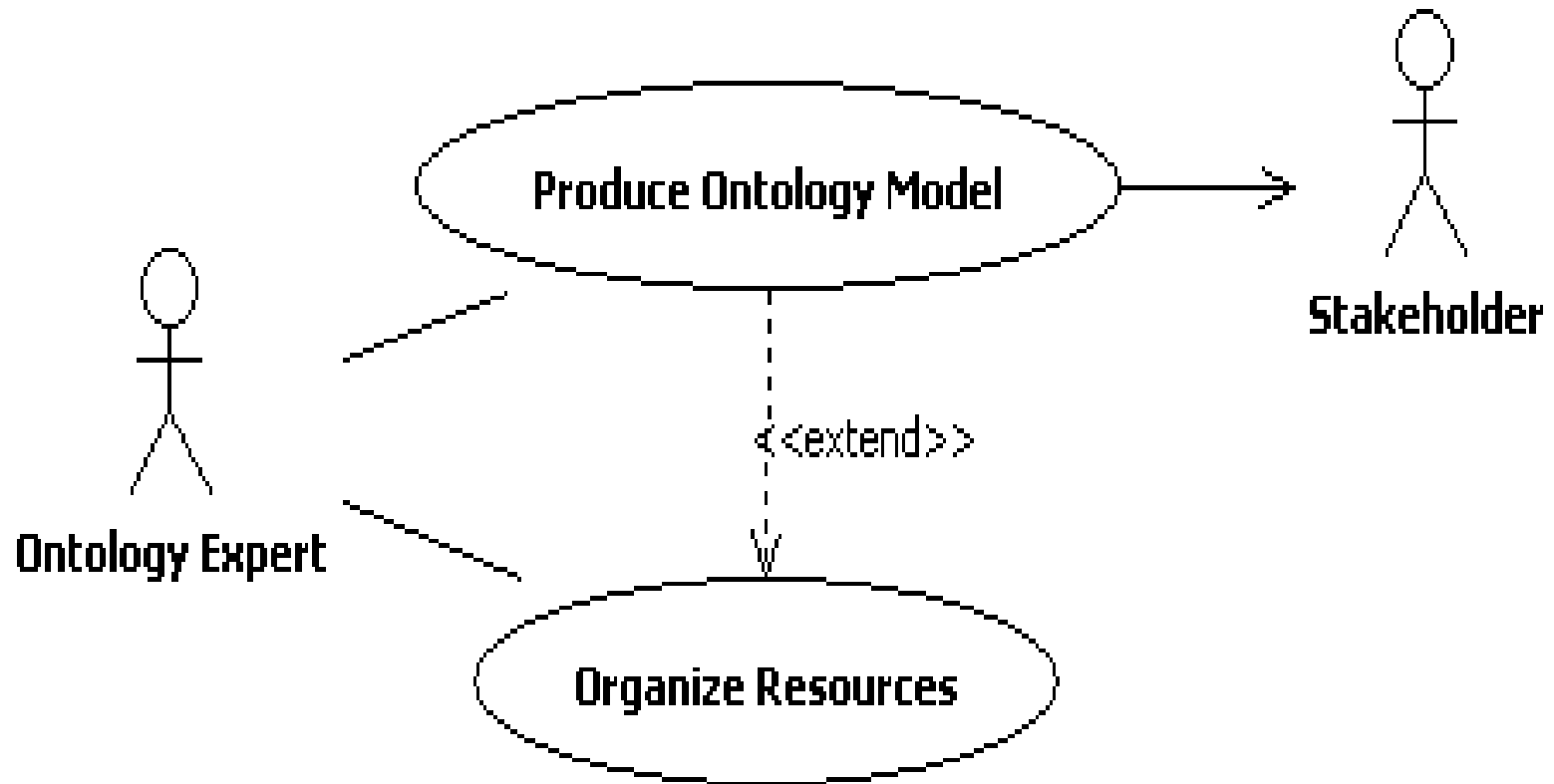
# Contesto Applicativo

---



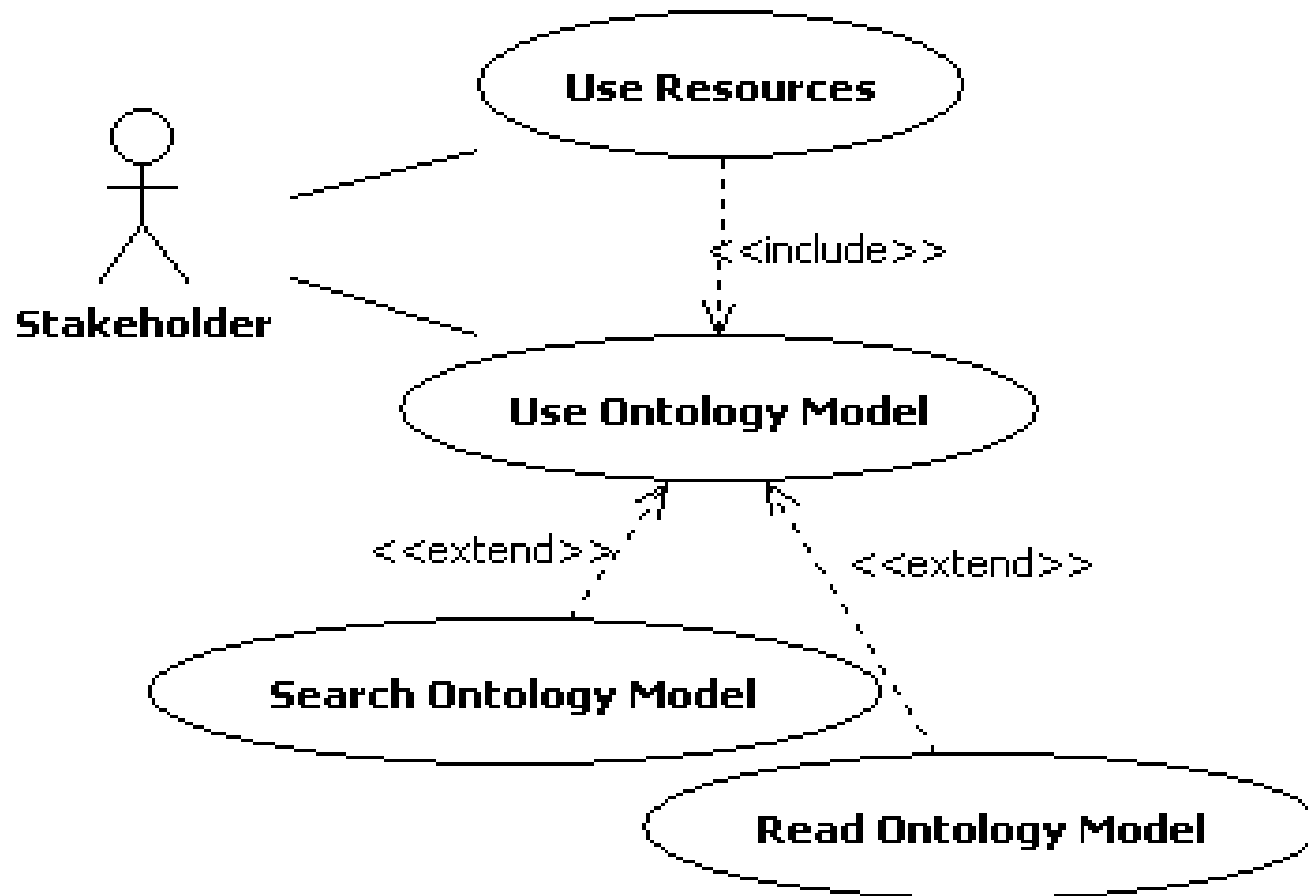
# Contesto Applicativo

---



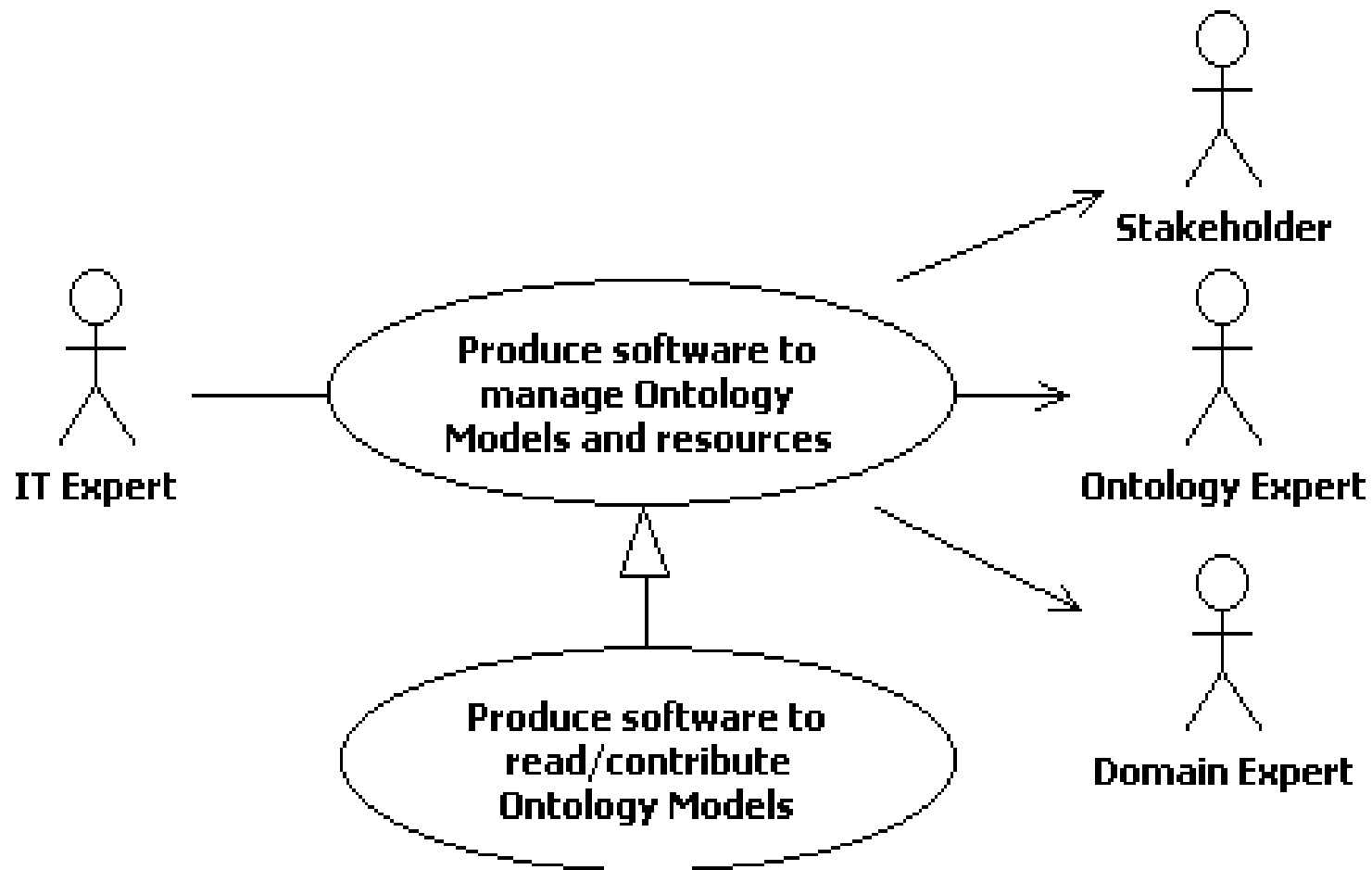
# Contesto Applicativo

---



# Contesto Applicativo

---





# Muddy – Ricerca 1

In verde il numero di sotto-classi contenute

In azzurro il numero di istanze contenute

Categoria (classe) corrente

Elenco categorie attraversate

Elenco sotto-categorie per la classe corrente

Elenco risultati

I risultati in blu sono le istanze del modello

I risultati in verde sono la definizione delle classi del modello

Home Find Directory Downloads Uploads User data Logout

MixDesign

MixDesign

Activity (11) Concept (3) Procedure (6,2) Tool (21)

Risultati 1 - 6 su 6

[A mix design protocol](#)  
The mix design protocol aims at defining a guided procedure to select the appropriate mix of earth t...

[Activity](#)

[Concept](#)

[MixDesign](#)

[Procedure](#)

[Tool](#)

Internet 100%

# Muddy – Ricerca 2

Bottone per l'esecuzione della ricerca

Testo da cercare

Risultato ricerca

Il risultato della ricerca viene paginato



# Muddy - Consultazione

Link per lasciare un feedback  
relativo alla risorsa

Nome della risorsa

Categorie cui appartiene  
la risorsa

Descrizione della risorsa

The screenshot shows a web browser window with the URL <http://www.stlab.dsi.unifi.it/muddy/pages/resource/resourceView.jsf>. The page content includes:

- Navigation links: [Home](#), [Find](#), [Directory](#), and [Logout](#).
- Resource title: **Maison de l'Ouarsenis**.
- Breadcrumbs: [Architecture](#) > [Traditionnelle](#) > [Méditerranéenne](#) > [Espace Méditerranéen](#) > [Algérie](#).
- Description: "Habitat dispersé qui s'accroche à flanc de montagne, adossé à la pente. Les maisons de cette typologie sont éparpillées par petits groupes qui correspondent chacun à une famille étendue. Une clôture délimite un espace d'étendue variable devant et autour de la maison qui est le siège des activités domestiques."
- Architectural drawing: A technical drawing of a traditional house with a pitched roof and wooden frame.
- Metadata tags: **typologie\_analytique** (Culture) > [Aspis](#) > [Milieu Rural](#) > [Niveau économique](#) Bas > [Usage](#) Permanent > [Origine](#) Evolutif > [Implantation](#) Épars > [Caractère](#) Fixe > [Morphologie](#) Complexe.
- External references: **riferimento\_esterno** > [IMG Ouarsenis](#) > [PDF](#).
- Location: **Algérie** > [TYPOLOGIES ARCHITECTURALES](#).
- Feedback link: [Feedback](#).

Link relativi alle risorse  
collegate

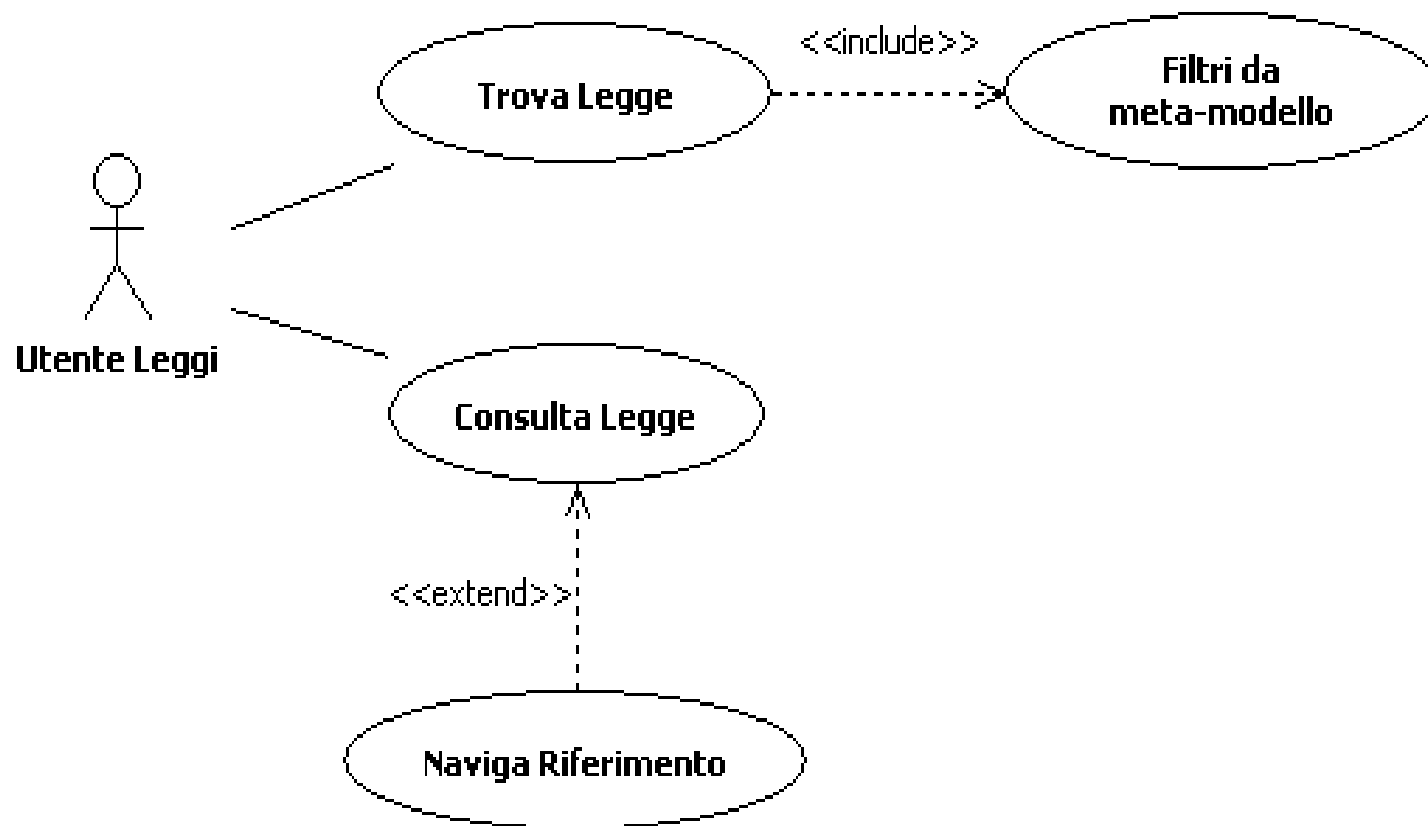
# Sperimentazione in Ambito Giuridico

---

- **Oggetto della sperimentazione:**
  - Modellazione di Leggi
  - Modellazione di Concetti Legali
  - Mapping tra sistemi giuridici diversi

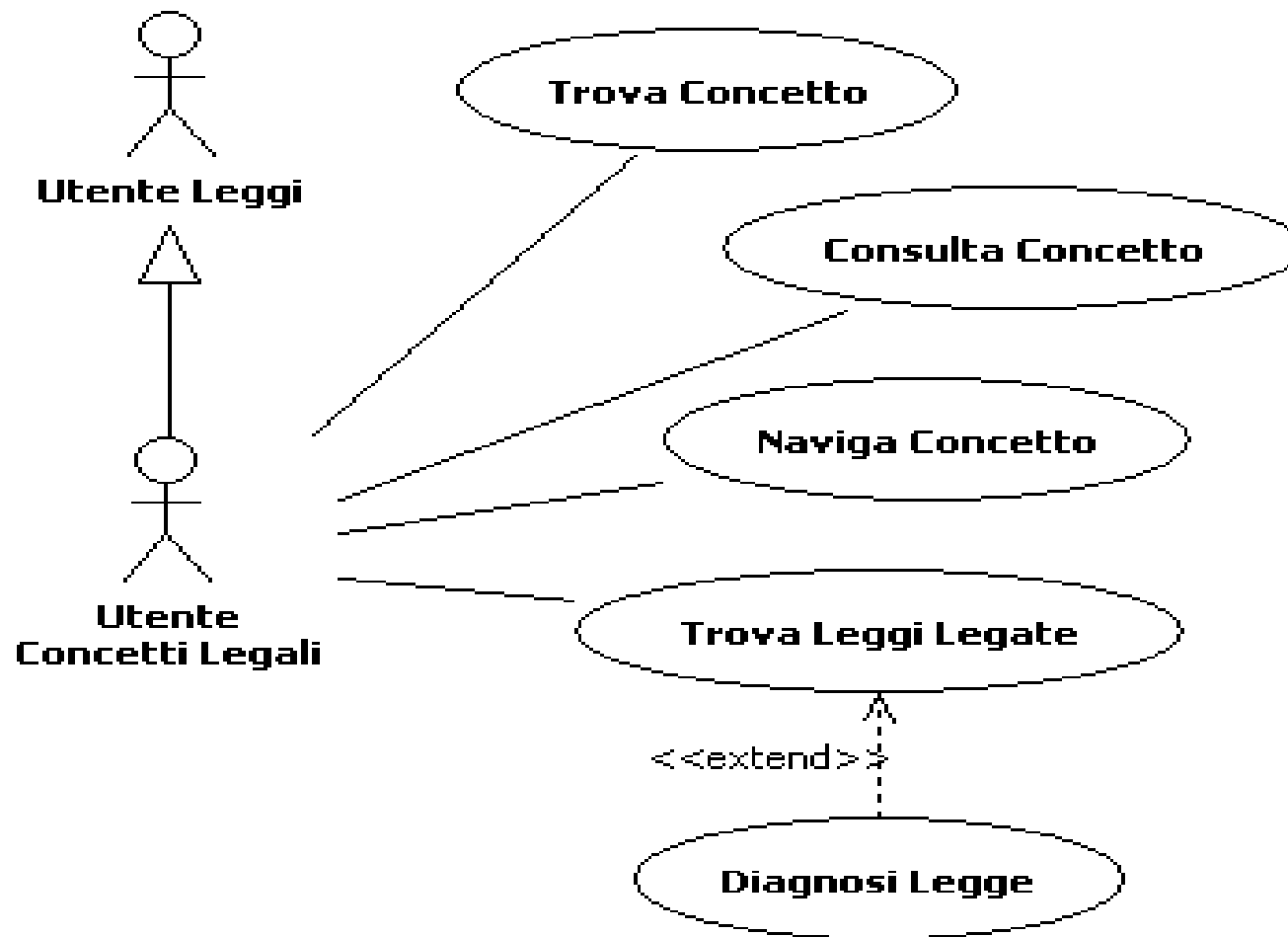
# Sperimentazione in Ambito Giuridico

---



# Sperimentazione in Ambito Giuridico

---



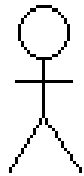
# Sperimentazione in Ambito Giuridico

---

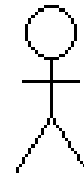
- **Utilità dell'avere un modello ontologico**
  - Confrontare sistemi giuridici diversi
  - Valutare l'impatto di una norma e più in generale la consistenza di un insieme di leggi
  - Determinare quali norme si applicano a certi comportamenti o procedure (es. come faccio a rinnovare il passaporto?)

# E-learn

---



**Professore**



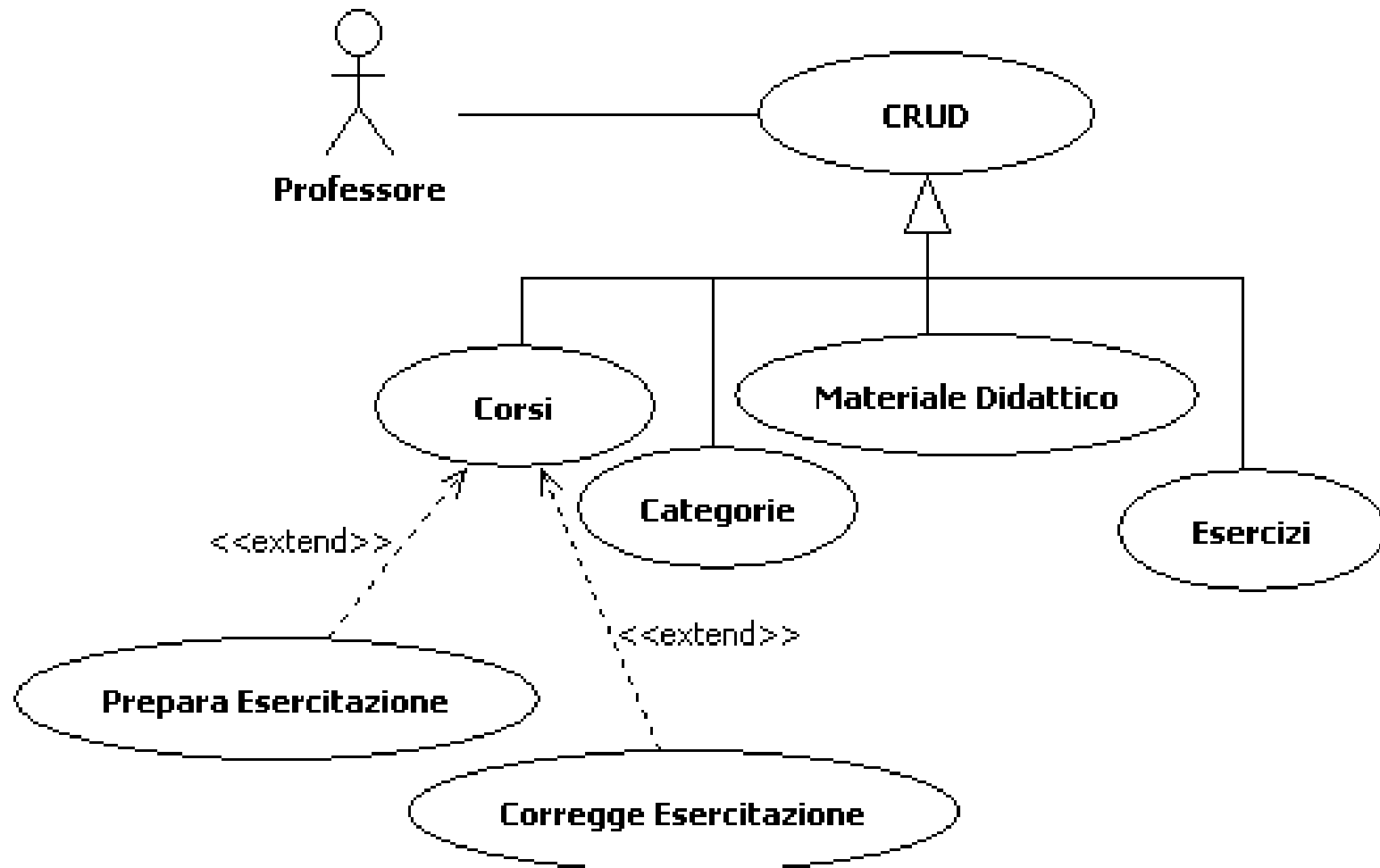
**Studente**

- **Professore:**
  - Fornisce e cataloga in nuovo materiale didattico
  - Gestisce corsi
  - Prepara e corregge esercitazioni
- **Studente:**
  - Consulta il materiale didattico
  - Richiede esercizi
  - Si iscrive ai corsi
  - Esegue esercitazione e controlla i risultati



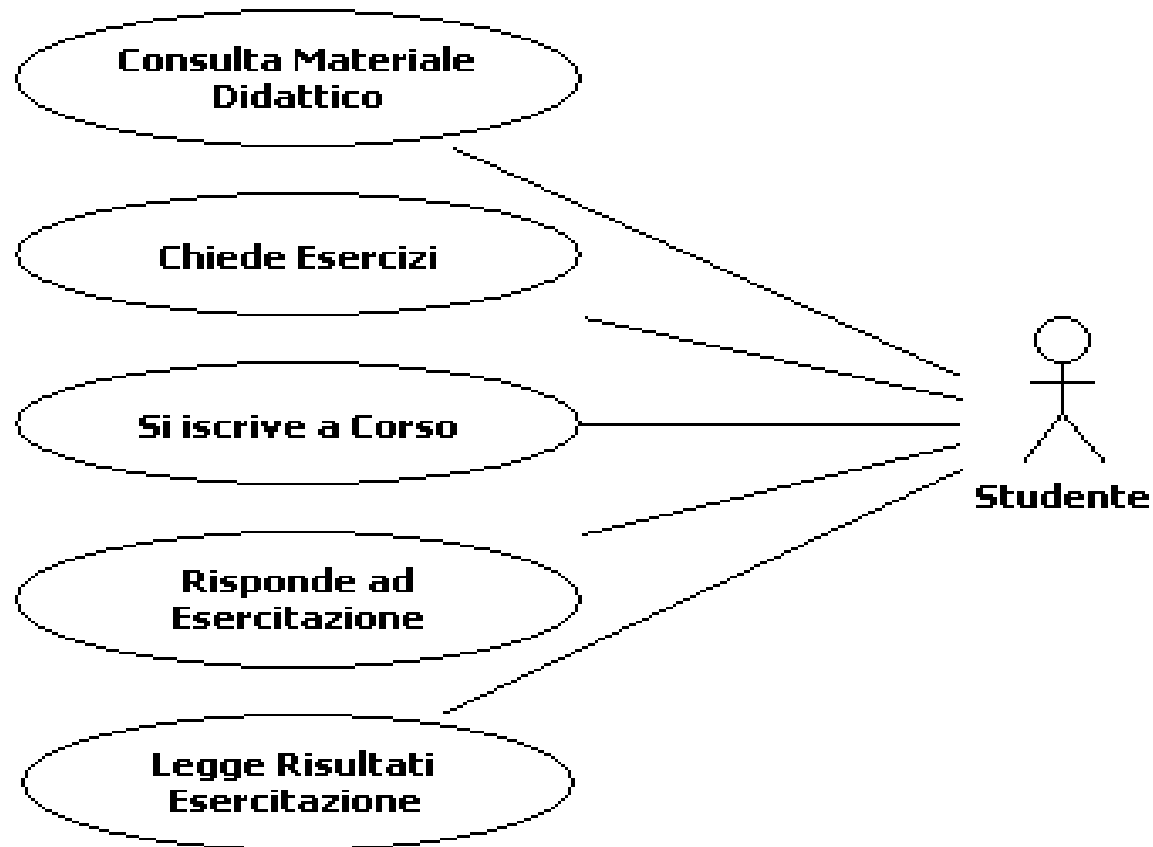
# E-Learn

---



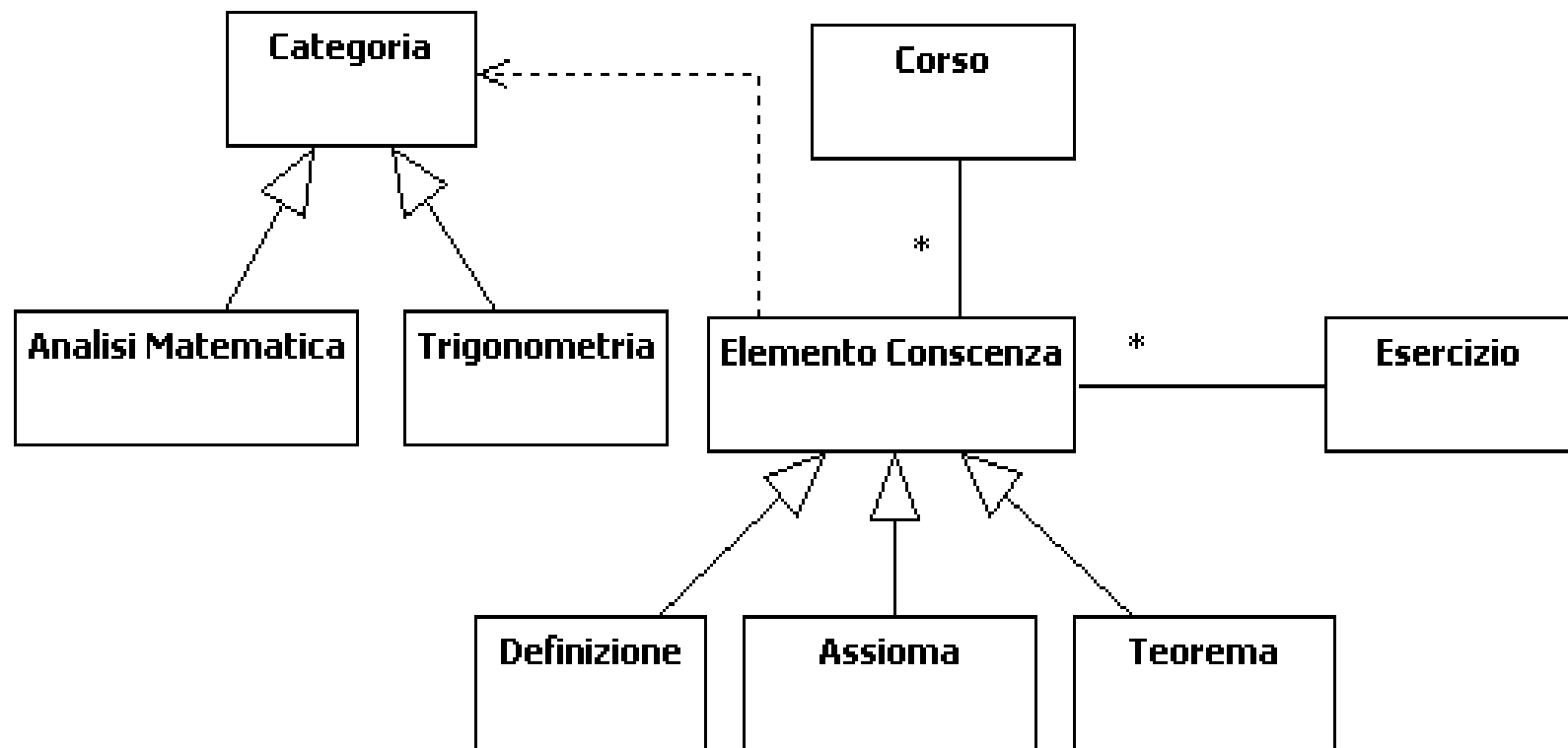
# E-Learn

---



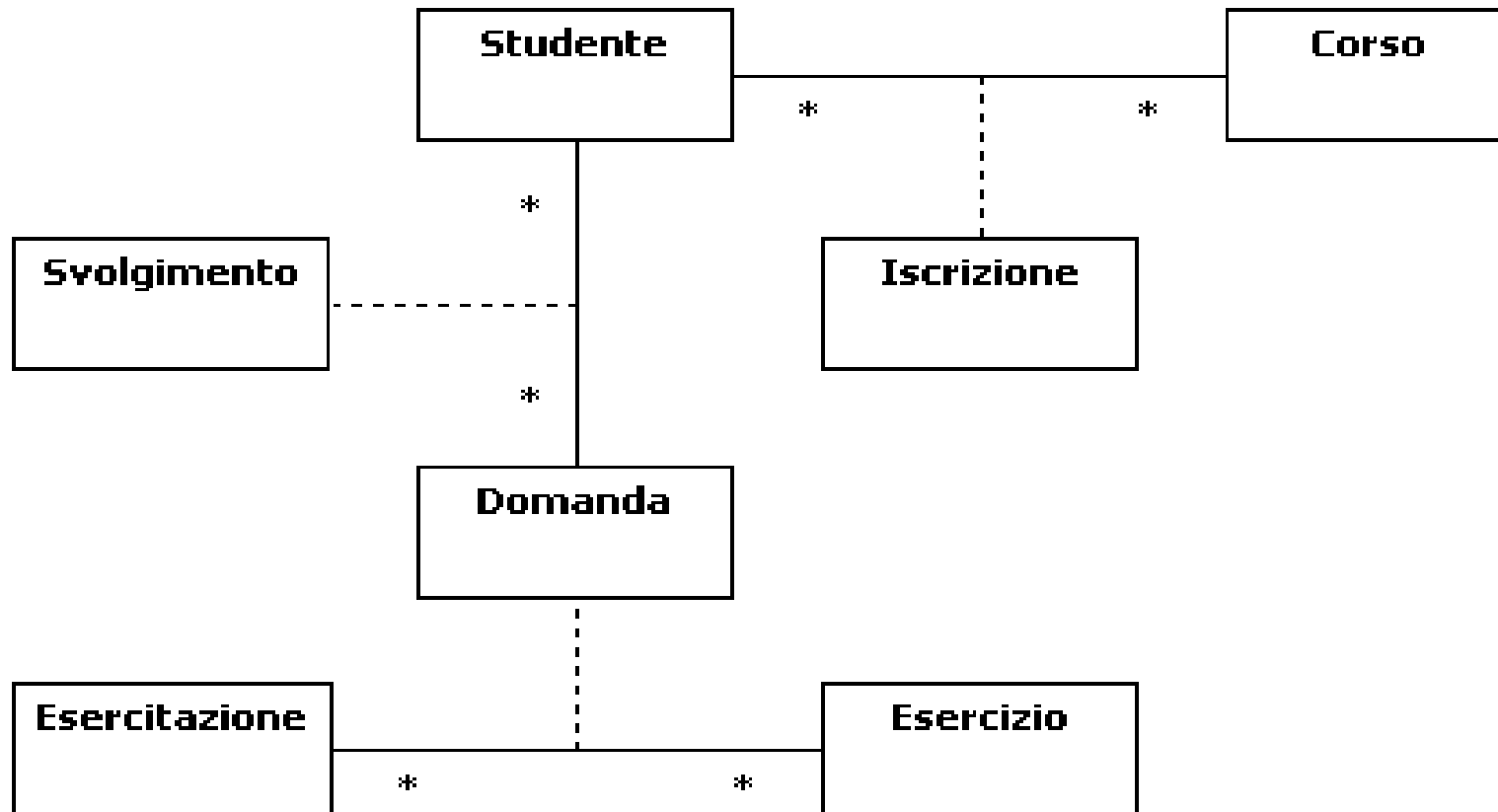
# E-Learn

---



# E-Learn

---



# E-Learn

---

- **Funzionalità possibili grazie ad inferenza:**
  - Calcolare l'insieme di esercizi necessario a verificare ogni argomento di un corso
  - Proporre ad uno studente l'insieme di esercizi che servono a verificare la conoscenza di quegli argomenti che in passato non sapeva bene
  - Determinare automaticamente l'ordine di precedenza dei corsi o comunque l'insieme di prerequisiti necessari

# Link Utili

---

- Protégé: <http://protege.stanford.edu/>
- OWL: <http://www.w3.org/TR/owl-features/>
- RDF: <http://www.w3.org/RDF/>
- SWRL: <http://www.w3.org/Submission/SWRL/>
- SPARQL: <http://www.w3.org/TR/rdf-sparql-query/>
- Jena Api: <http://jena.sourceforge.net/>
- Sesame Api: <http://www.openrdf.org/>
- OWL-S: <http://www.w3.org/Submission/OWL-S/>
- Semantic Web: <http://www.semwebcentral.org/>