

Protégé

Cos'è un'ontologia

Un'ontologia è

- una descrizione formale di concetti in un dominio (**classi**)
- le proprietà di ciascun concetto (**slot**)
- le restrizioni sugli slot (**facets**)

ONTOLOGIA + UN INSIEME DI ISTANZE DI CLASSI = BASE DI CONOSCENZA

A cosa serve un'ontologia

Un'ontologia consente di definire:

- Un vocabolario
- I concetti di base interpretabili da una "macchina"
- Le relazioni fra i concetti



ESPORTABILITA'

(rende un sistema indipendente dalla applicazione e quindi esportabile in altri domini)

MODELLAZIONE

(definizione della conoscenza del dominio, utile per analisi requisiti e la definizione use case in fase progettuale)

INTEROPERABILITA'

(due sistemi possono fare riferimento ad una stessa ontologia anche se utilizzano formati diversi di rappresentazione della conoscenza)

Esempio di utilizzo

WEB	Ontologie
Informazioni in diversi formati	La modellazione, potrebbe facilitare il processo di ricerca su fonti eterogenee.
Mancanza di una struttura unica	Sono XML-based, quindi consentono la descrizione dei contenuti in maniera strutturata.

Protégé: un editor per la creazione di ontologie

Caratteristiche:

- Piattaforma open-source
- Può esportare le ontologie in vari formati: RDF(S), XML Schema e OWL
- E' basato su Java
- E' estendibile (esistono numerose API e plug-in)
- Dispone di numerosi ambienti plug-and-play che consentono un rapido sviluppo delle applicazioni
- Sviluppato dall'università di Stanford (<http://smi.stanford.edu/projects/protege/>)

Protégé API

Protégé fornisce un'interfaccia che altre applicazioni possono utilizzare per accedere alle knowledge bases. Queste applicazioni non necessitano di usare o visualizzare nessuna delle interfacce utente di Protégé.

Esiste una classe che fornisce il metodo *getKnowledgeBase()* il quale permette di accedere al contenuto della base di conoscenza. Tali contenuti possono essere usati ad esempio da motori di inferenza (es. **Drools**, **Mandarax**, **Jess**, ...), i quali possono compiere le loro inferenze ed eventualmente aggiornare la base di conoscenza.

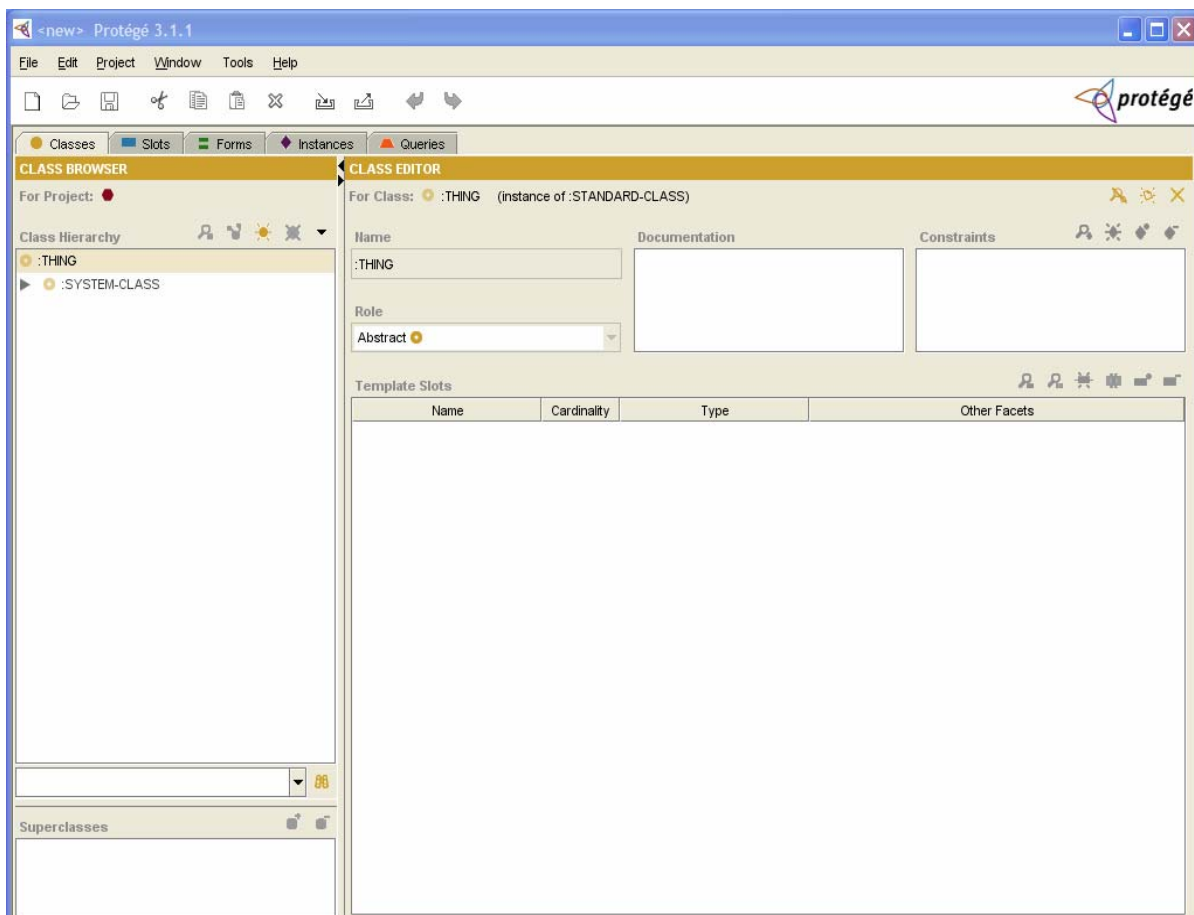
Ulteriori informazioni relative alle API di Protégé possono essere reperite su

<http://protege.stanford.edu/doc/pdk/kb-api.html>

La piattaforma Protégé

Esistono due modalità per creare le ontologie in Protégé:

- Il **Protégé-Frames editor**, consente di costruire e popolare le ontologie che sono basate su “frame”, secondo il protocollo OKBC (Open Knowledge Base Connectivity protocol [www.ai.sri.com/~okbc/]). In questo modello, un’ontologia è costituita da un’insieme di classi organizzate in gerarchia, rappresentanti un insieme di concetti. Le classi sono caratterizzate da slot e relazioni.
- Il **Protégé-OWL editor**, consente di costruire ontologie per il *Semantic Web*, in particolare secondo il linguaggio OWL [www.w3.org/2004/OWL/]. Un’ontologia OWL può includere descrizioni di classi, di proprietà e le loro istanze.



Progettazione di un'ontologia

- **Progettare** un'ontologia significa:
 1. Determinare il **dominio** e lo **scopo** dell'ontologia
 2. Considerare la possibilità di **riusare** ontologie esistenti
 3. Individuare i **concetti** chiave del fenomeno da descrivere
 4. Organizzare i concetti in classi e **gerarchie** tra le classi
 5. Definire le **proprietà** delle classi
 6. Definire **vincoli** (valori leciti) sulle proprietà
 7. Creare le **istanze**
 8. Attribuire i **valori** alle proprietà per tutte le istanze create

Approcci possibili...

Per lo sviluppo di una gerarchia di classi esistono diversi approcci (Uschold and Gruninger, 1996, "Ontologies: Principles, Methods and Applications"):

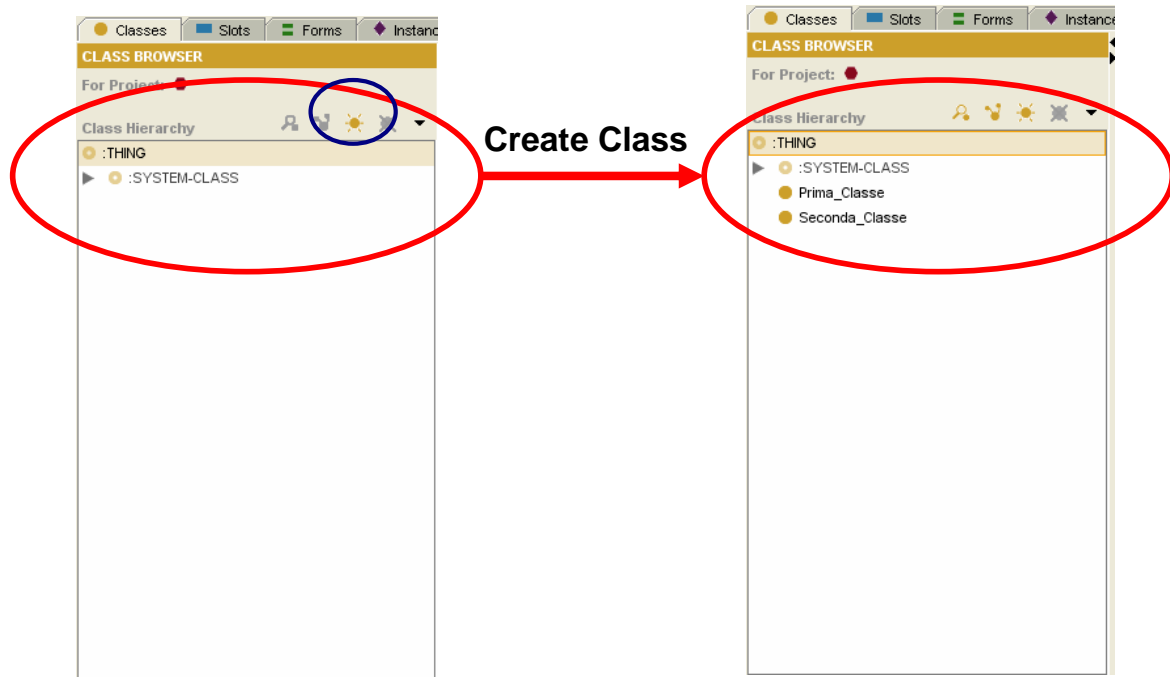
- **Top-down.** Si definiscono i concetti più generali del dominio e la loro specializzazione.
- **Bottom-up.** Si inizia definendo le classi più specifiche della gerarchia, raggruppandole in seguito in concetti più generali.
- **Combinazione.** Si combinano la metodologia bottom-up e quella top-down: si definiscono i concetti di maggior rilievo (importanza?) quindi li si generalizzano e li si specializzano.

Individuate le classi della nostra ontologia, esse potranno essere create in **Protégé Classes Tab.**

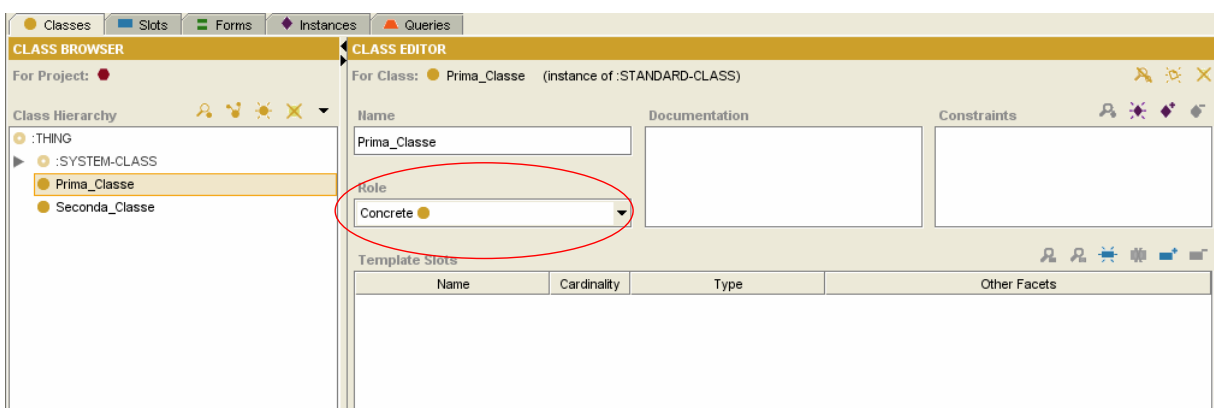
Tutte le classi individuate come capostipiti di una gerarchia, saranno figlie di :THING (sarà la classe più generale della nostra ontologia).

(Nota: la :SYSTEM_CLASS è utilizzata da Protégé per definire le strutture dei form)

Protégé Classes Tab



Protégé Classes Tab (2)

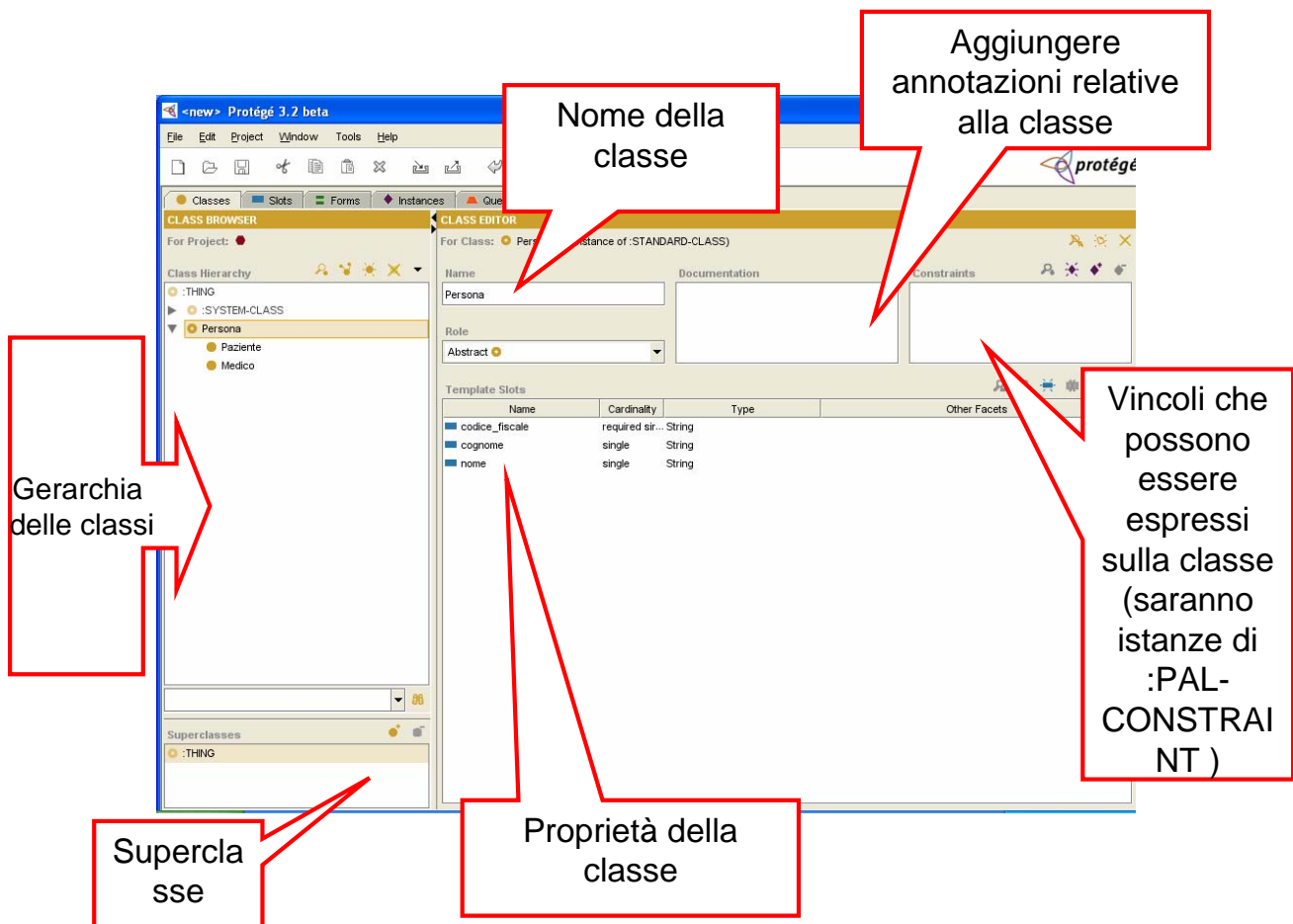


All'interno di Protégé Classes Tab si può definire il **“role”** di una classe:

- Astratta
- Concreta

Una caratteristica importante è che Protégé supporta l'**ereditarietà multipla**.

Inoltre si possono definire gli **slot**, i **vincoli** sui valori degli slot, le **relazioni** fra le classi e le proprietà.



Le proprietà di una classe - slot

Definite le classi, si devono descrivere le proprietà di queste, che nell'ontologia saranno gli slot. Gli slot possono rappresentare:

- proprietà estrinseche (ad esempio il gusto di un vino)
- proprietà intrinseche (ad esempio il nome)
- parti di un oggetto, nel caso in cui questo sia strutturato (possono essere parti sia "astratte" sia "concrete")
- le relazioni con altre classi

Gli slot possono avere dei vincoli, ad esempio il tipo, il numero di valori (la cardinalità)...

In **Protégé** i **tipi** previsti sono: *boolean*, *float*, *integer*, *string*, *symbol* (consente di enumerare i valori possibili), *class*, *instance*, *any*; inoltre la definizione dell'attributo può essere obbligatoria o meno (setando in maniera opportuna il flag "*required*") e uno slot può avere una cardinalità singola o multipla.

Nome dello slot (si consiglia per nomi composti di utilizzare il carattere “_” per esportabilità)

Note relative allo slot

Scelta del tipo

Valore di default dello slot

Dominio

Slot inverso (vedremo in seguito la sua utilità)

Le proprietà di una classe – slot (2)

Si può definire (per qualunque tipo tranne per il tipo generico Any) il valore di default

Vincoli di cardinalità sugli slot

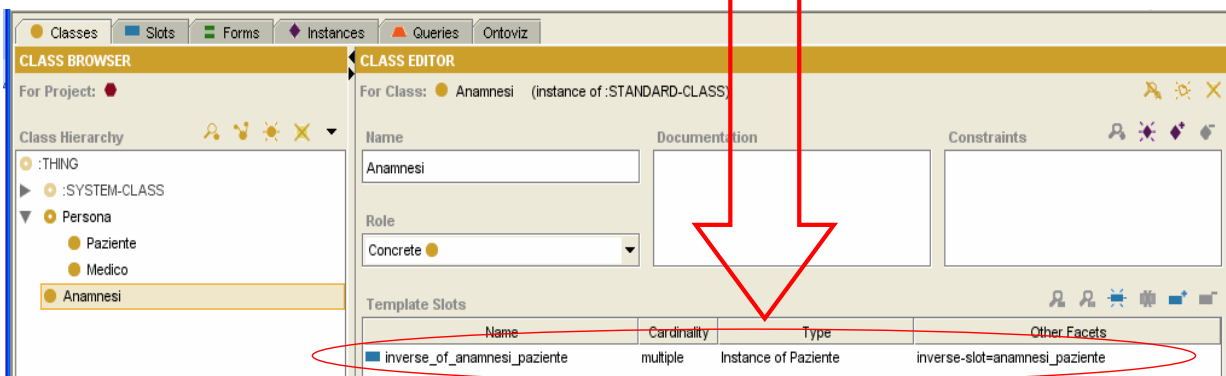
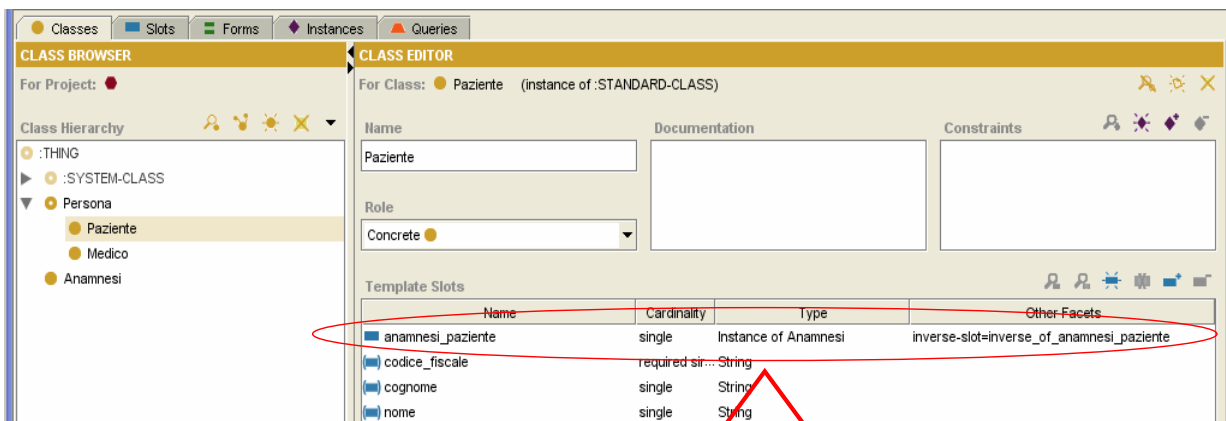
Sugli slot di tipo float e integer si può definire il valore minimo e il valore massimo

Inverse slot

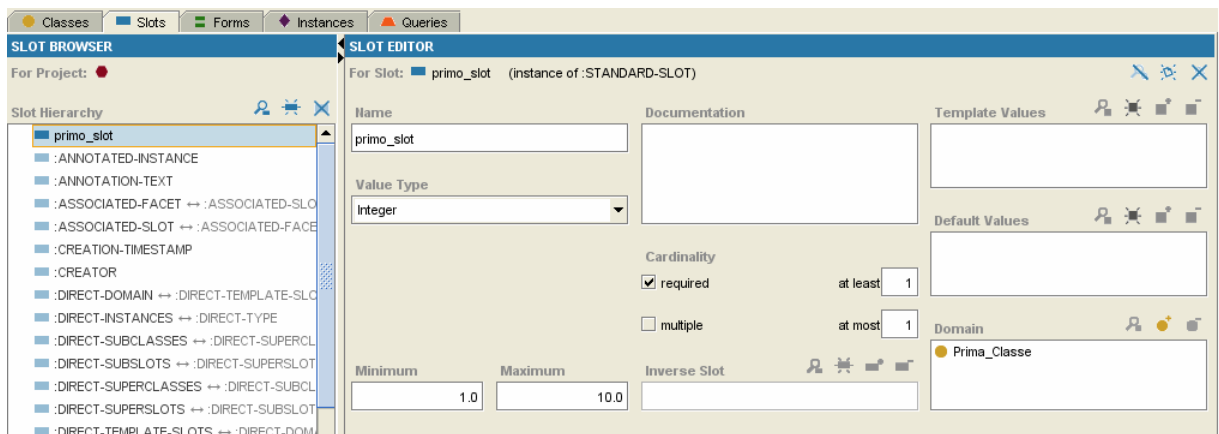
Il valore di uno slot può dipendere dal valore di un altro slot.

Ad esempio ad un Paziente si può fare un'Anamnesi e l'Anamnesi è relativa solo ad un paziente.

Il Sistema è in grado di assicurare la consistenza della conoscenza di base in tal senso. Ad esempio cancellando un'istanza di Anamnesi, Protégé provvederà a rimuovere il corrispettivo riferimento dal paziente.



Protégé Slot Tab



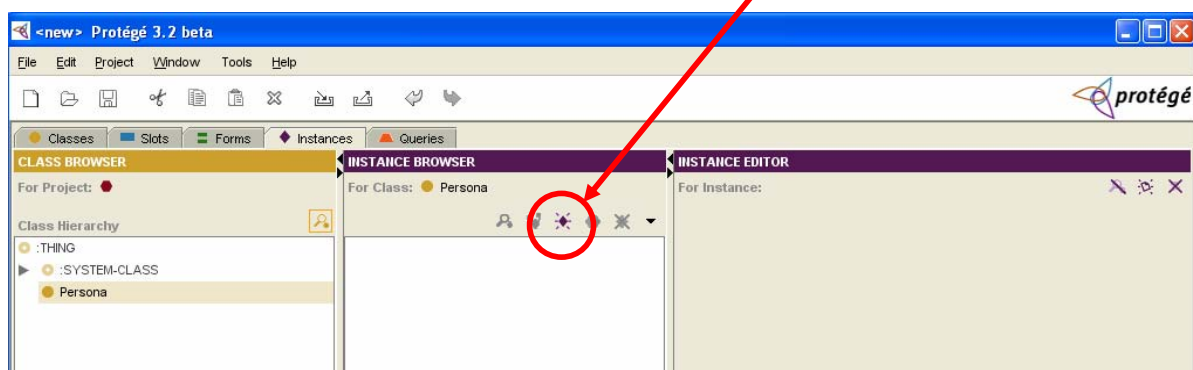
Nota: se dovete eliminare uno slot che avevate precedentemente creato, si consiglia vivamente di eliminarlo sempre da Protégé Slot Tab, in quanto se viene eliminato per mezzo dell'interfaccia Protégé Class Tab, lo slot non verrà definitivamente eliminato ma rimarrà presente nell'ontologia non legato a nessuna classe.

Creazione di un'istanza

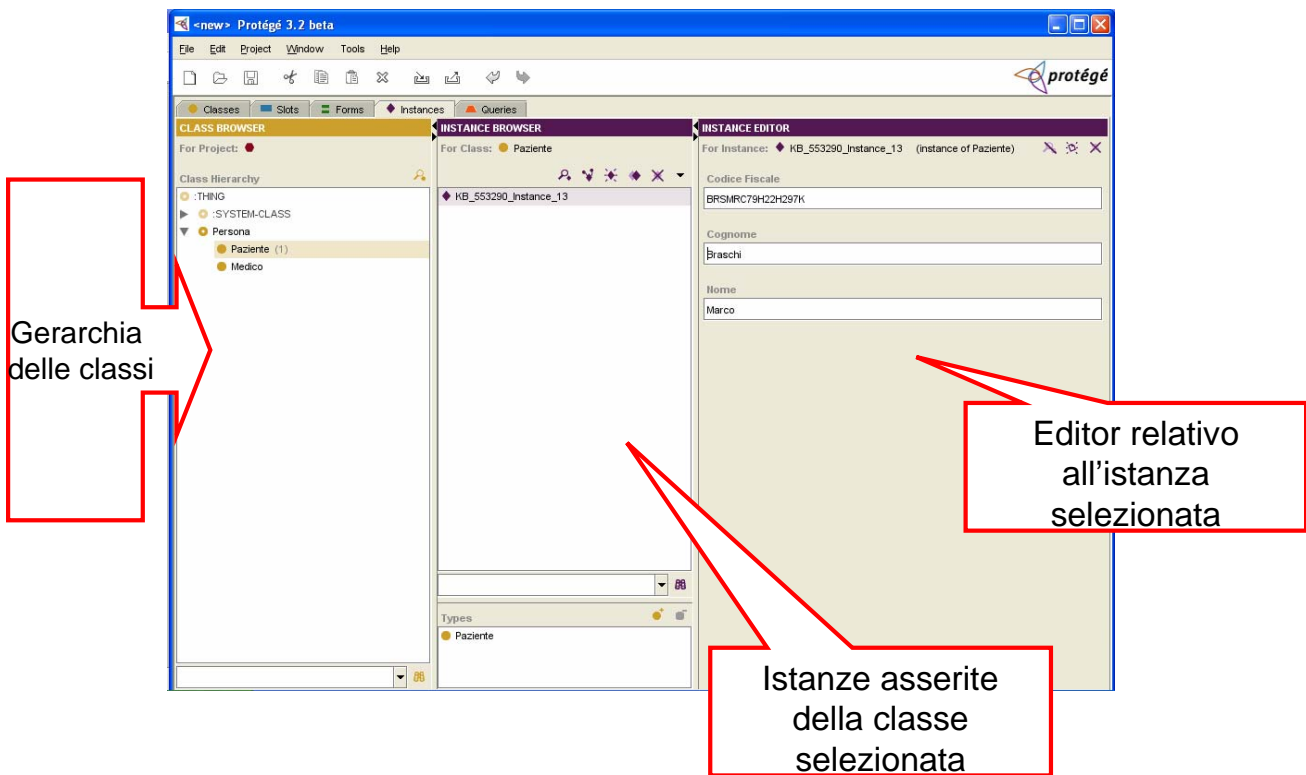
All'interno di **Protégé Instance Tab** si possono andare a definire le istanze per le varie classi.

Definire un'istanza significa:

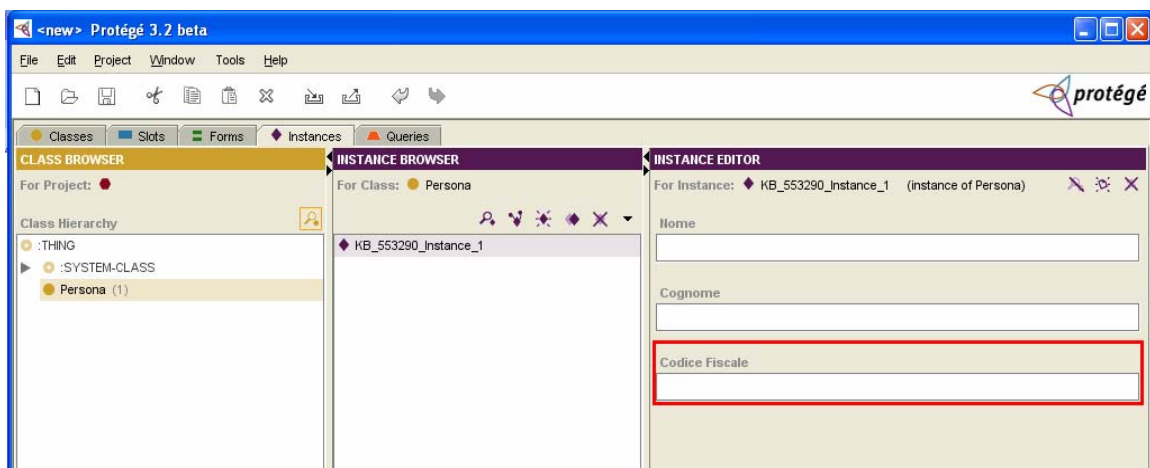
- 1) Scegliere una classe
- 2) Creare un'istanza della classe
- 3) Immettere i valori degli slot



Protégé Instance Tab



Esempio di creazione di un'istanza



Nota: Gli slot required vengono evidenziati da Protégé Instance Tab con un bordo rosso. La **consistenza** deve essere garantita dall'utente, in quanto Protégé consente la memorizzazione di istanze anche aventi campi required non compilati.

Il numero di istanze presenti in una data classe viene visualizzato accanto al nome della classe.

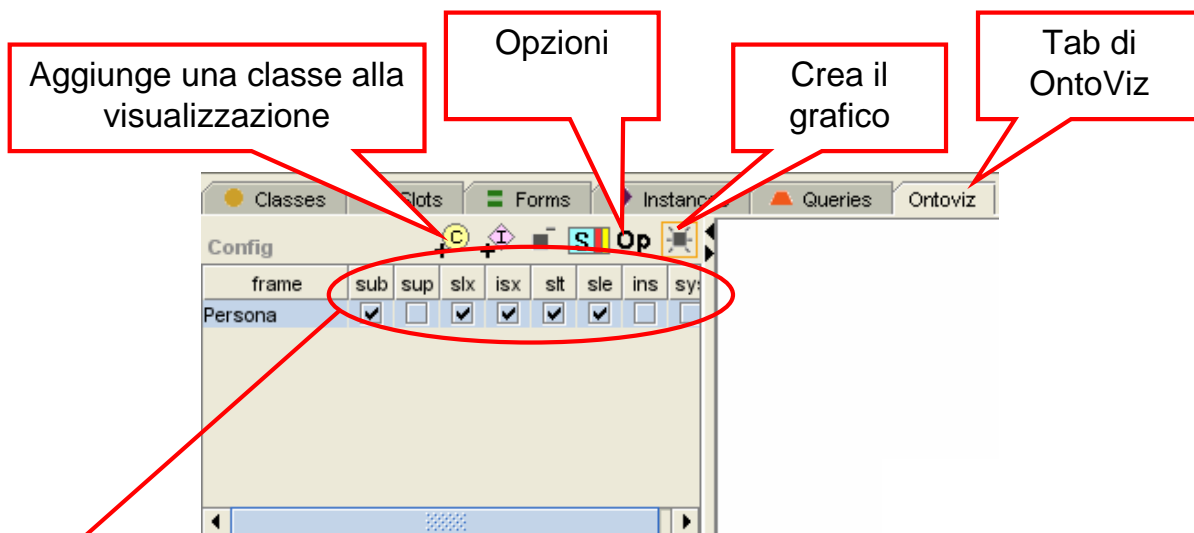
Un esempio di Plug-in: Ontoviz

Proprietà dei Plug-in:

- Estendono le funzionalità di Protégé per adempiere a molteplici necessità (Visualizzazione, Importazione/Esportazione ad altri formati, etc...)
- Ogni plug-in ha un suo tab associato attivabile attraverso il menù Project→Configure

Ontoviz:

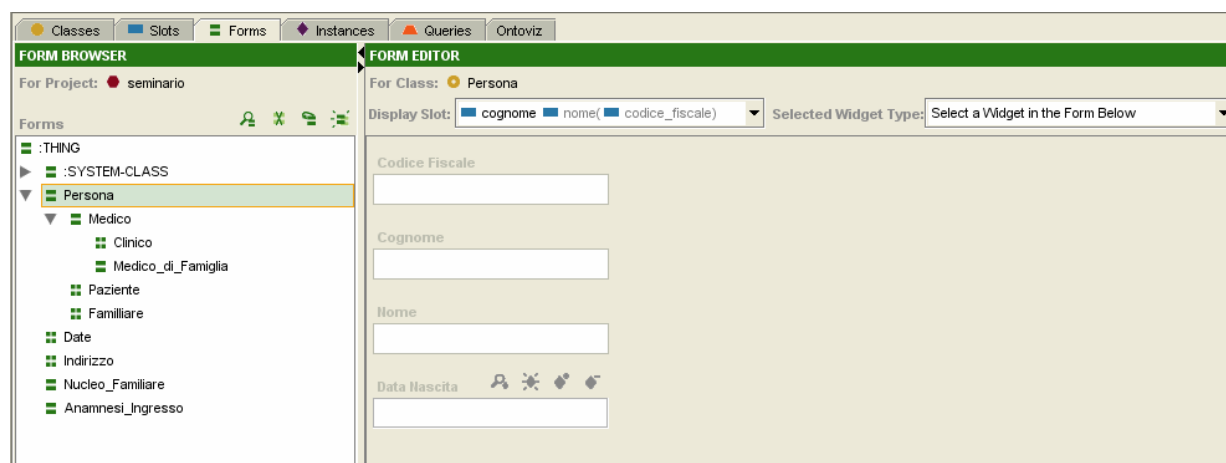
- Plug-in per la visualizzazione grafica dell'ontologia.
- Si basa su GraphViz, il quale deve essere correttamente installato
- Consente di specificare differenti livelli di dettaglio (Classi, Slots, SottoClassi, Istanze, Relazioni, etc...)
- Consente di specificare uno o più soggetti centrali della visualizzazione (Se si desidera visualizzare tutta l'ontologia si può specificare semplicemente THING)



- **sub**: Sottoclassi
- **sup**: Superclassi
- **slx**: Classi correlate
- **isx**: Classi correlate con slot relazione di inverse slot
- **slt**: Slots
- **ins**: Istanze
- **sys**: Classi di sistema

Forms tab

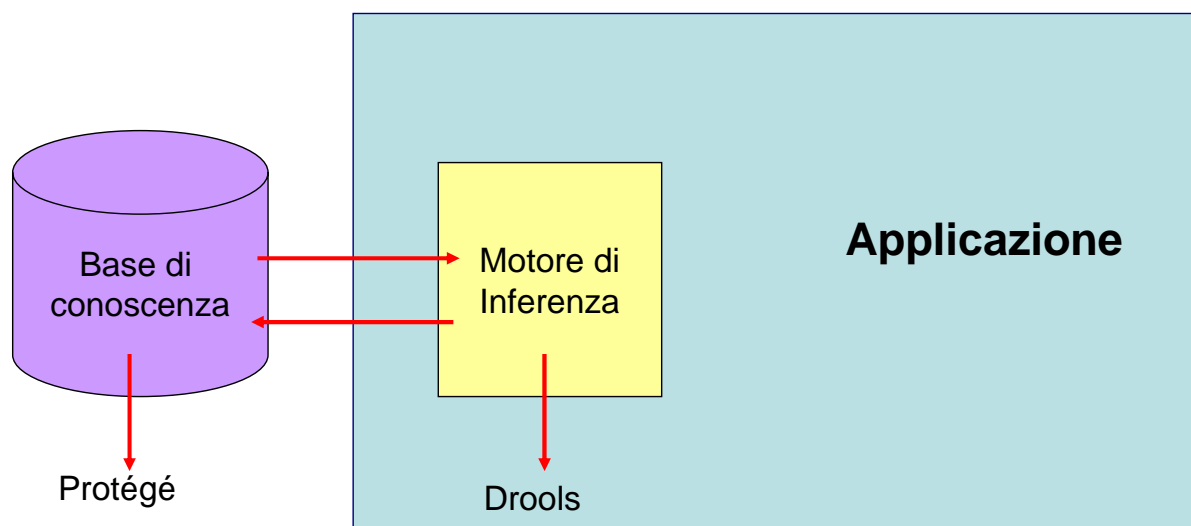
- Funzionalità di protégé che permette di creare interfacce grafiche per l'inserimento di istanze che potranno essere utilizzate dall'utente per popolare la knowledge base.
- Le interfacce sono personalizzabili sia nei contenuti che nella forma. Utile per un uso allo stato prototipale, meno utile per la creazione di un'applicazione in quanto non consente di introdurre componenti aggiuntivi rispetto a quelli supportati di base (ad esempio non esistono bottoni)



Esempio di utilizzo di Protégé – Progetto Sere-na

Scopo:

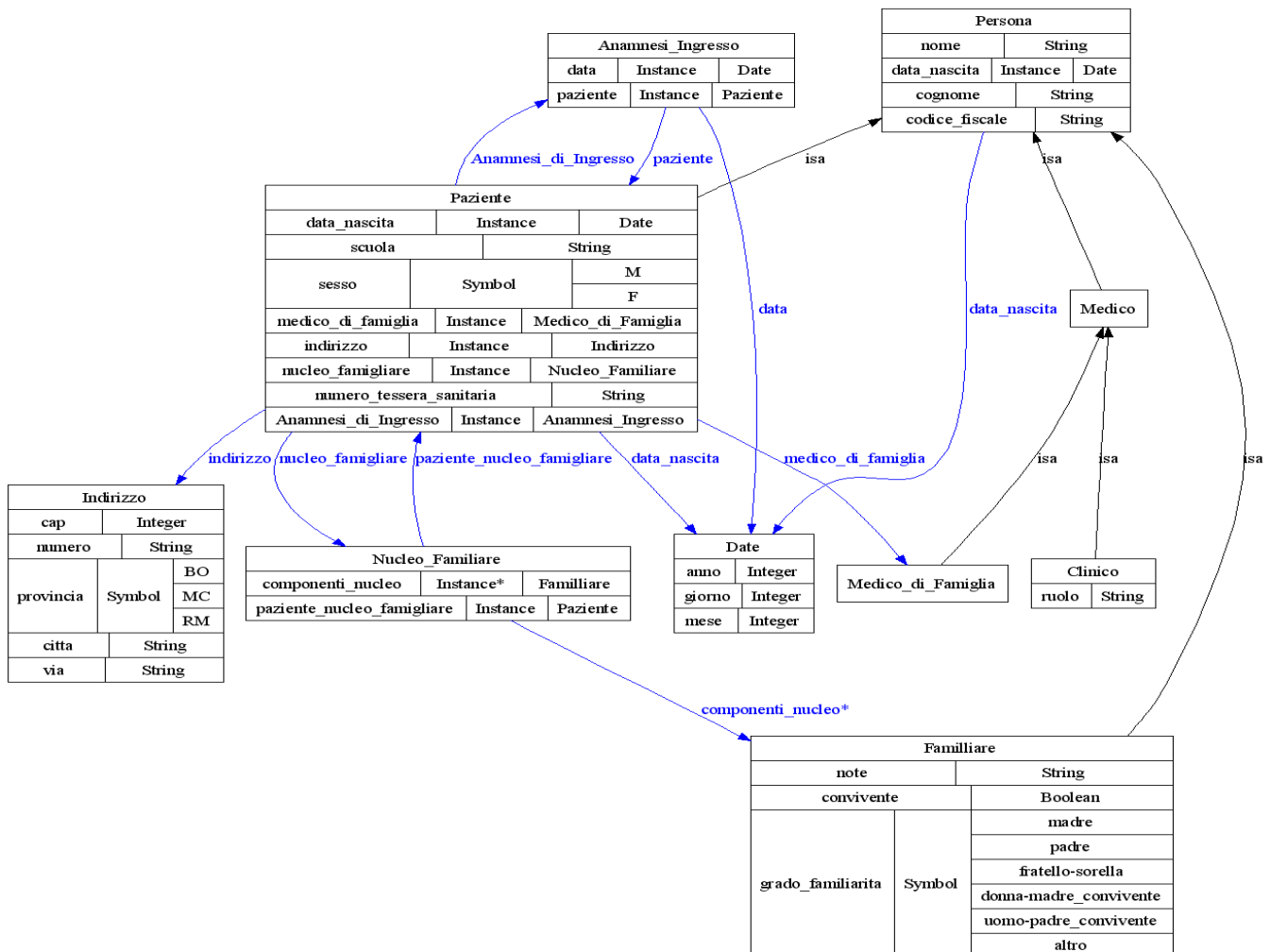
Realizzazione di un Sistema Esperto in campo medico per il supporto e la formazione alla diagnosi della Dislessia.



Esercitazione

- Creare una semplice ontologia cercando di seguire le linee guida espresse in questa breve introduzione
- L'ontologia assegnata è parte dell'ontologia reale del progetto Sere-na opportunamente semplificata
- Il dominio è un ambiente ospedaliero
- Lo scopo è la gestione dei dati del paziente

Ulteriori esempi e tutorials possono essere reperiti dal sito ufficiale di Protégé



Riferimenti

Sito ufficiale:

<http://protege.stanford.edu/>

OWL:

www.w3.org/2004/OWL/

OKBC:

www.ai.sri.com/~okbc/

API Protégé:

<http://protege.stanford.edu/doc/pdk/kb-api.html>

Tutorial:

<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>

<http://www.kde.cs.uni-kassel.de/lehre/ws2004-05/wissensverarbeitung/protegetutorial-part1.pdf>