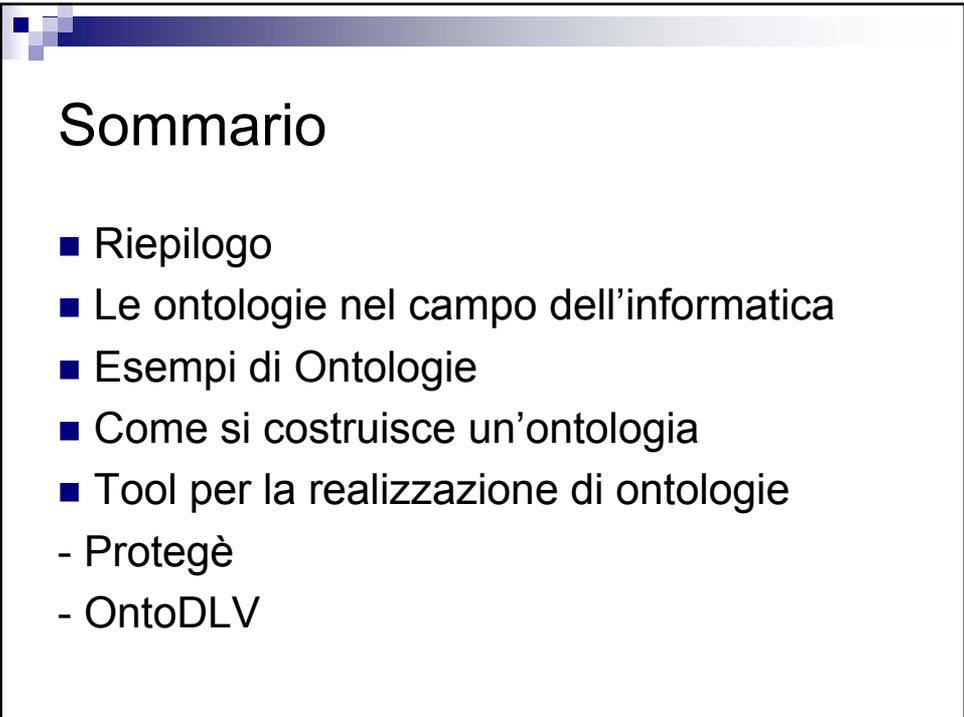


Basi di dati e Sistemi Informativi sul Web

Esercitazione 1

Ing. Giuseppe Pirrò



Sommario

- Riepilogo
- Le ontologie nel campo dell'informatica
- Esempi di Ontologie
- Come si costruisce un'ontologia
- Tool per la realizzazione di ontologie
 - Protegè
 - OntoDLV

Ontologia (una definizione)

- Una ontologia rappresenta il *modello concettuale* di un mondo.
- Il modello concettuale è la struttura formale di (un pezzo di) realtà percepita ed organizzata da chi modella.
- Una *ontologia* è una descrizione formale esplicita dei concetti di un dominio.

Ontologie e Database

- Sia le ontologie che i Database sono *metodologie di rappresentazione della conoscenza*.
- DIFFERENZE PRINCIPALI
 1. Approccio alla realtà da modellare: con un Database si ha una visione più concreta e specifica del mondo; con una ontologia si cerca di modellare concettualmente il mondo (il DB si concentra più sulle istanze, l'ontologia più sulle entità).
 2. Le Ontologie consentono di *ragionare* sul mondo, ovvero di espandere la conoscenza sul mondo, utilizzando *regole di inferenza*, espresse in uno specifico *linguaggio*.

Costruire un'ontologia (1)

■ Ingegneria ontologica

Si occupa della costruzione dell'ontologia:

1. Definisce i concetti del dominio (*classi*);
2. Organizza i concetti in una *gerarchia*;
3. Definisce gli attributi dei concetti, le restrizioni su di essi e le relazioni tra concetti (*proprietà o slot*);
4. Definisce istanze dei concetti, popolando l'ontologia

Costruire un'ontologia (2)

■ Regole Fondamentali

1. Non esiste un'unica metodologia corretta per costruire una ontologia. La soluzione migliore dipende sempre dall'*uso* che si farà dell'ontologia.
2. La costruzione dell'ontologia è un *processo iterativo*.
Una prima versione dell'ontologia viene solitamente valutata, discutendone il contenuto con esperti di dominio o utilizzandola in una applicazione pratica. Successivamente l'ontologia viene raffinata e migliorata.
3. I concetti dell'ontologia sono generalmente oggetti (nomi) e relazioni (verbi) del mondo di interesse

Le ontologie in Informatica

- **Computer Science**: area dell'intelligenza artificiale che studia i metodi per rappresentare correttamente l'universo che ci circonda
- **Perchè servono in CS?**
- **Condivisione di conoscenza**: per non duplicare sforzi nello sviluppo di sistemi software
- **Comunicazione**: sia tra agenti software (tra di loro) che tra agenti software e esseri umani

Semantic Web!

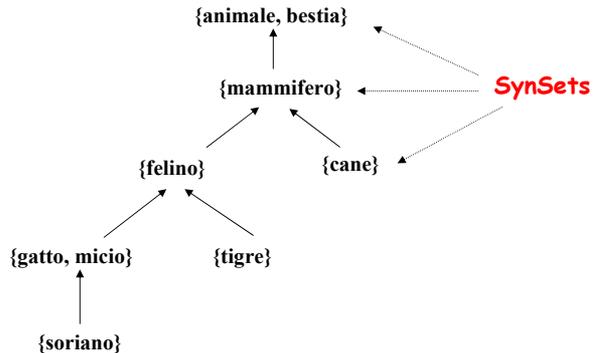
ALCUNE ONTOLOGIE

WordNet

Da un elenco di parole:

<tigre, cane, animale, mammifero, bestia, micio, soriano, gatto, felino>

A un dizionario strutturato:



NB: La stessa parola può appartenere a più SynSets

Alcune versioni

WordNet: Cognitive Science Laboratory dell'Università di Princeton
(inglese) Fine anni '80

EuroWordNet: su Fondi dell'Unione Europea.
(multilingue - ILC-Pisa per l'italiano) Metà anni '90

ItalWordNet: IRST-ICT (Trento). Un progetto nazionale
(italiano) Fine anni '90

Le relazioni di WordNet (originale)

Relazione	POS collegate	Esempi
Sinonimia	nome/nome; verbo/verbo; aggettivo/aggettivo; avverbio/avverbio	book/volume; to eat/to take in; great/outstanding; greatly/dramatically
Antonimia	nome/nome; verbo/verbo; aggettivo/aggettivo	man/woman; to enter/to exit; long/short
Iponimia	nome/nome; verbo/verbo	slicer/knife; to walk/to move
Meronimia	nome/nome	head/nose
Implicazione	verbo/verbo	to buy/to pay
Causa	verbo/verbo	to kill/to die
Somiglianza	aggettivo/aggettivo	wet/humid
Attributo	aggettivo/nome	tall/stature
Relativo al nome	aggettivo/nome	fraternal/brotherhood

Osservazioni

- Alcune delle relazioni adottate sono alquanto vaghe
- POS sta per Part Of Speech (nome, verbo, ecc.)
- POS diverse collegate solo in casi particolari
- Le Relazioni sono strutture del tutto diverse dai Synsets

Cyc

- Il progetto Cyc (da enCYClopedia) nasce nel 1984 ed è ancora in corso (si veda il sito <http://www.opencyc.org/>).
- Attualmente, Cyc include oltre un milione di concetti, mentre la versione pubblica OpenCyc comprende circa 6.000 concetti e 60.000 relazioni tra di essi

Struttura di Cyc

- Cyc è costituita da due componenti

Constraint Language (Logica dei predicati)
CyL (linguaggio basato su frame)

Cycl

Units

I frame di Cyc si chiamano Units. Esse includono, come tutti i frames, degli slots

Esempio di Unit relativa ad un'istanza

#\$Texas

#\$capital: (#\$Austin)

#\$residents: (#\$Doug Guha Mary)

#\$stateOf: (#\$UnitedStatesOfAmerica)

Tutti i simboli che hanno il prefisso # \$ sono Units. Una caratteristica fondamentale di Cyc è che, come si vede dall'esempio, anche gli slots sono Units (SlotUnits)

Cycl

Esempio di Unit relativa ad uno slot

#\$residents

#\$instanceOf: (#\$Slot)

#\$inverse: (#\$residentOf)

#\$makesSenseFor: (#\$GeopoliticalRegion)

#\$entryIsA: (#\$Person)

#\$specSlots: (#\$lifelongResidents # \$illegalAliens # \$registeredVoters)

- **Da un punto di vista formale, gli slots sono relazioni binarie**
- **Quindi, è necessario definire il dominio (*#\$makesSenseFor*) e il range (*#\$entryIsA*)**
- **E' anche possibile definire relazioni tra relazioni (*#\$inverse* e *#\$specSlots*)**

Il Constraint Language

Una variante della logica dei predicati basata su 'restricted quantification'

Es. Tutte le persone hanno una madre e la loro differenza di età è maggiore di 16

Versione in logica dei predicati standard

$$(\forall x) \text{person}(x) \rightarrow [(\exists y) \text{person}(y) \wedge \text{mother}(x,y) \wedge \text{greaterThan}(\text{diff}(\text{age}(y), \text{age}(x)), 16)]$$

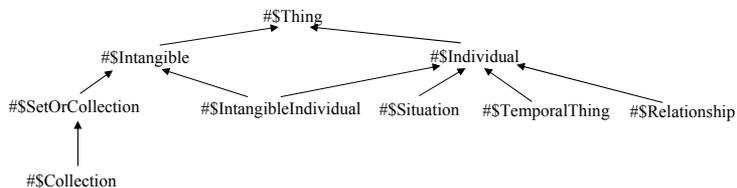
Versione con quantificazione ristretta

$$(\forall x_{\text{person}}) (\exists y_{\text{person}}) \text{mother}(x,y) \wedge \text{greaterThan}(\text{diff}(\text{age}(y), \text{age}(x)), 16)]$$

Versione Cyc

$$\begin{aligned} &(\#\$\text{ForAll } x (\#\$\text{Person}) \\ &\quad (\#\$\text{Exists } y (\#\$\text{Person}) \\ &\quad\quad (\text{LogAnd } (\#\$\text{mother } x y) \\ &\quad\quad\quad (\#\$\text{GreaterThan } (\#\$\text{Diff } (y \#\$\text{Age}) (x \#\$\text{Age}) 16)))) \end{aligned}$$

Il top level di Cyc



Una lista dei concetti di Cyc si può trovare al sito
<http://www.cyc.com/cyc-2-1/toc.html> (aggiornato al 1997)

Alcune descrizioni di concetti (dalla documentazione Cyc)

#Thing:

è l'insieme universale: la collezione di ogni cosa! Ogni costante Cyc nella Base di Conoscenza è membro di questa collezione. Inoltre, ogni collezione della Base di Conoscenza è membro della collezione #Thing.

Conclusioni su Cyc

- Un sistema enormemente complesso, che include sia una parte di rappresentazione di conoscenza e inferenza, sia una ontologia vera e propria

VANTAGGI:

- Dimensione
- Potenza inferenziale
- Ottimizzazione dei ragionamenti

SVANTAGGI:

- Troppo complesso
- Non chiare le scelte ontologiche
- Alcuni insuccessi (es. Legami col linguaggio naturale)

Conclusioni sulle Ontologie

- Le ontologie sono uno strumento essenziale per l'interoperabilità e la comunicazione tra agenti
- Negli ultimi 10 anni una quantità impressionante di finanziamenti e attività di ricerca e sviluppo
- Molte proposte e sistemi, ma ancora nessuno standard; poichè un'ontologia è utile se è ampiamente condivisa, siamo ancora a metà strada
- Alcuni studiosi hanno forti dubbi sulla possibilità di realizzare un'ontologia standard (difficoltà tecniche e salvaguardia di diversità culturali)
ma
- Si tratta comunque di una strada obbligata



STRUMENTI PER A SCRITTURA DI ONTOLOGIE

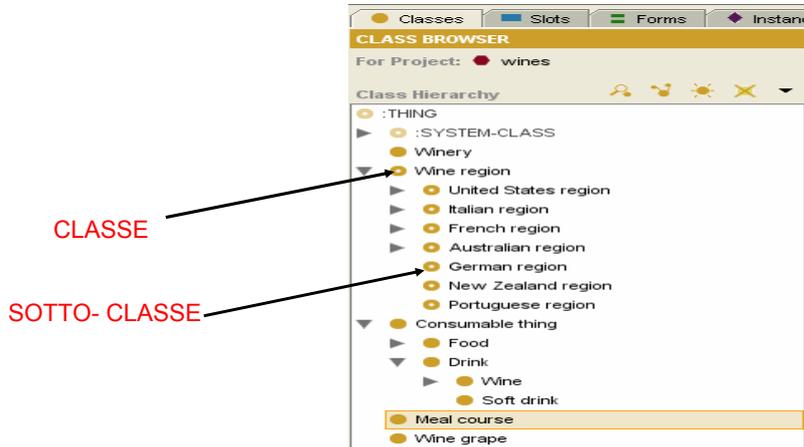


Protege

- Università di Stanford
- protege.stanford.edu/
- Circa 42.000 utenti registrati
- Ultima versione 3.1.1

Protege

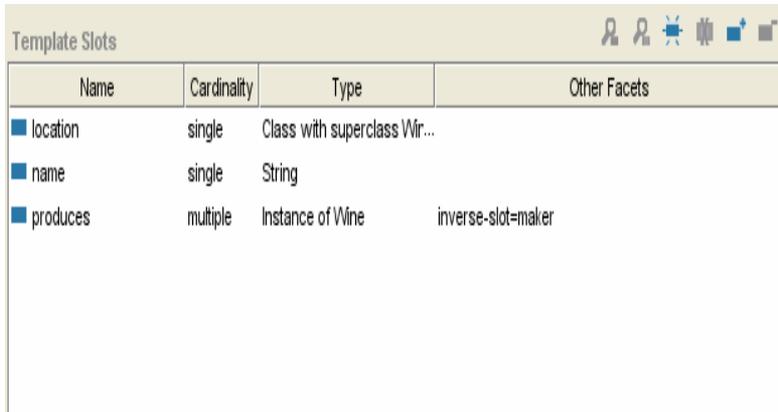
■ Definizione di Classi e Sottoclassi



Definire proprietà

- Le proprietà di una classe (*slot*) descrivono:
 - Gli *attributi* della classe;
 - Le *relazioni* che essa instaura con altre classi.
- Le proprietà possono essere *semplici* (stringhe, numeri, ecc...), oppure *complesse* (“puntano” ad altri oggetti dell’ontologia).
- In una gerarchia una sottoclasse eredita tutti gli slot della relativa superclasse

Definire proprietà



Name	Cardinality	Type	Other Facets
location	single	Class with superclass Wir...	
name	single	String	
produces	multiple	Instance of Wine	inverse-slot=maker

Definire restrizioni

- Le restrizioni sulle proprietà (dette anche *facets*) descrivono o limitano i possibili valori che uno slot può assumere.
- Le restrizioni più comuni sono:
 - Tipo: tipo di uno slot (stringa, numero, istanza ecc...)
 - Cardinalità: numero di valori assegnabili a uno slot
 - Minimo/Massimo: valori di soglia che può assumere uno slot numerico
 - Default: valore di default di uno slot



FINE