**MODSEM**

**Semantic web** Rappresenta contenuti secondo vocabolari dotati di una semantica comprensibile alle macchine e agli umani, sono le ontologie computazionali (scritte in RDF e OWL)

**Linked Data** Usare dei vocabolari condivisi per descrivere e rappresentare dati usando il sistema degli IRI esposti tramite API

**Rappresentazione della conoscenza**

**Limiti della logica classica**

* **Inadeguatezza espressiva** La logica classica è un formalismo piatto, tutte le affermazioni sono sullo stesso piano e esprimono conoscenza personale e immutabile, le procedure di dimostrazione sono diverse dal ragionamento umano, non è adatta a rappresentare il mondo reale
* **Monotòna** Le conoscenze non possono essere cancellate, la conoscenza può solo aumentare, non si può rappresentare il cambiamento
* **Limiti computazionali** Il calcolo dei predicati è semi-dicibile ovvero è possibile dimostrare cosa discende dalla base di conoscenza ma non cosa NON discende dalla base di conoscenza

**Reti semantiche**

Le reti semantiche sono basate su una struttura a GRAFO, i nodi sono i concetti gli archi le relazioni

**Vantaggi** Ogni nodo è collegato alle sue proprietà, le informazioni sono immediatamente raggiungibili

Permette di indicare la nozione di RILEVANZA, le info di un nodo sono vicine al nodo stesso

**Il ragionamento nelle reti semantiche** Nelle reti semantiche il ragionamento consiste nel seguire un percorso tra i nodi

**Reti semantiche proposizionali SNePS**

Le reti semantiche proposizionali includono nodi che rappresentano le proposizioni

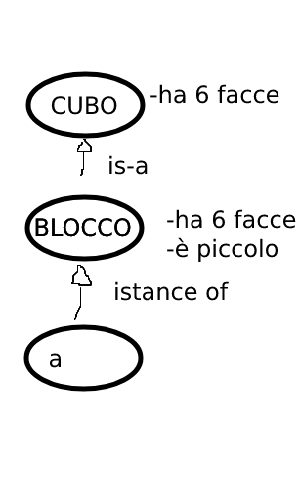
Si introduce una dimensione **EPISTEMICA**: rappresentare conoscenze soggettive di più soggetti senza contraddizioni

Maria crede che il blocco A sia sul blocco B

**Il sistema SNePS** è un software che permette di rappresentare la rete, cercare i nodi con cerche caratteristiche e inferire nuove asserzioni a partire da quelle esistenti

**Relazioni IS-A nelle reti semantiche**

Gli archi is-a hanno un significato diverso se collegano 2 classi o un individuo e una classe:

IS-A classe e sottoclasse

ISTANCE OF individuo fa parte di una classe

**Ereditarietà delle proprietà**

Grazie all’ereditarietà è possibile avere rappresentazioni meno ridondanti

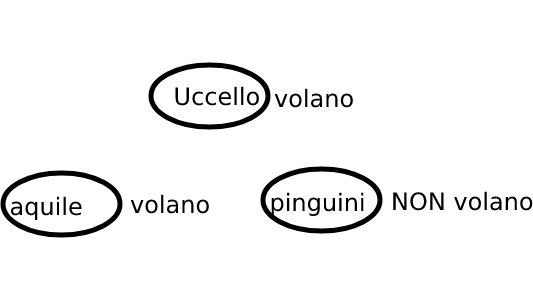
* Una classe eredita le proprietà delle sovraclassi
* Le sottocolassi hanno proprietà piu specifiche delle sovraclassi
* Seguendo il percorso di is-a e istance of è possibile ragionare sulle

Proprietà di una classe

**Eccezioni nell’ereditarietà**

Per gestire le eccezioni è necessario rilasciare la proprietà della monotonicità, ovvero

Il fatto che non si puo aggiungere conoscenza

****INFERENZE BLOCCATE DAL BACKGROUND

D= { Uccello (x) : Vola (x) } //Se x è un uccello vola

-------------------------------

{Vola (x)}

W= { Uccello(Aquila), Uccello(Pinguino), NonVola(Pinguino)}

Anche se uccello pinguino è vera non si puo applicare la regola perché è bloccata dal NonVola(Pinguino)

**Frame Theory**

Serve a rappresentare la conoscenza di tipo stereotipato

**FRAME** E’ una struttura per rappresentare situazioni stereotipate es La festa di compleanno

I livelli + alti sono conoscenze sempre vere (c’è una torta)

I livelli + bassi sono condizioni o dati specifici (che tipo di torta)

**Slots** Sono i terminali, condizioni che devono essere rispettati

**Il linguaggio RDF** (Resource Description Framework)

// Linguaggi dello stack del web semantico:

**// RDF** E’ un linguaggio che permette di descrivere RISORSE

**// RDFS** Permette di descrivere RELAZIONI tra risorse

**// OWL** Permette di descrivere ONTOLOGIE computazionali

**// OWL2** Permette di descrivere ONTOLOGIE computazionali con un significato DEFINITO FORMALMENTE

RDF è basato sul concetto di grafo, i nodi sono le entità e gli archi le relazioni

**RDF Data model** Triple < Soggetto, Predicato, Oggetto >

IRI identificano univocamente i nodi

Letterali

Blank nodes

**Le triple** SOGGETTO (IRI, blank-node)

PREDICATO (IRI)

OGGETTO (IRI, blank-node, Letterale)

L’insieme delle triple forma un grafo

* Gli IRI sono URL internaizionalizzati
* I letterali sono tipi di dato non IRI, es una stringa o un numero <Bob>
* I blank-node permettono di identificare risorse senza usare un IRI (è tipo una variabile)

**Uso dei vocabolari** Per descrivere una certa entità ci si basa su vocabolari condivisi invece di inventare ogni volta dei nuovi termini, (es love, like,… foaf/topic\_interest), in RDF il vocabolario è identificato da un PREFISSO che corrisponde a un namespace (es foaf: per www….foaf)

**Istance of in RDF <type>**

**Turtle**

Il turtle è uno standard per condivisione documenti RDF molto più leggibile di RDF

IRI racchiusi tra <http..>

**Stesso soggetto più preficati** :spiderman :enemyOf :GreenGoblin; // stesso soggetto

:name :”Spideman”.

:spiderman :name :”Spiderman”. //stesso soggetto e oggetto

:”Peter”.

**Prefissi**

@Prefix foaf: <http…/foaf>

Se si utilizza una BASE si puo rimuovere il prefisso davanti agli IRI

**Blank node in turtle**

[ ] :predicate :object

Si possono anche raggruppare un insieme di triple che hanno come soggetto lo stesso blank node

[

:predicate :object

:preficate :object

]

**Altro esempio**

[ ] :knows [ foaf:name :Bob ] -> [ ] knows [ ] :named :Bob

**Letterali**

:atomicNumber “2” **^^** <…/integer>

//Letterale //datatype

**N TRIPLES** Sotto insieme di Turtle senza abbreviazioni o prefissi

**Come vengono rappresentati i nodi in turtle e cosa succede se vengono caricati su una Linked Data Platform e viene creato il Knowledge Graph?**

Sono rappresentati da una coppia di parentesi quadre e i predicati associati a un blank node sono scritti al suo interno

Quando vengono caricati su una LDP gli viene associato un nome e l’estensione della LDP ma vengono mantenuti come blank nodes

**Cosa si intende per quadrupla**

La quadrupla è una tripla < Soggetto, Predicato, Oggetto > a cui si aggiunge un quarto valore che identifica a quale grafo appartiene la tripla

**RDFS Schemi RDF**

RDFS è l’estensione di RDF che permette di rappresentare le relazioni, è property centric perché ha un focus sulle proprietà

Classi, Proprietà, Type, Subclass, Subproperty, Domain, Range //rdf:type == a

**Class e subclass** :Quadrupede rdf:type rdfs:Class

rdf:type rdfs:subClassOf :Animale

**Proprietà** :hasFourLegs rdf:type rdfs:Property

**Collegare classi tramite le proprietà**

:abita rdfs:domain :Animal

Rdfs:range :Habitat

**Vocabolari RDF**

**Il vocabolario** è la definizione di classi e proprietà, viene pubblicato a un certo IRI perché possa essere riferito da tutti, serve a unificare ontologie diverse per non inventare ogni volta dei termini nuovi

**Knowledge graph** E’ la descrizione di risorse in riferimento a uno o piu vocabolari con le proprietà che collegano tra loro le risorse (sostanzialmente sono le tripe)

Alcuni vocabolari:

**FOAF** (Friend of a friend) serve a descrivere reti sociali

**DUBLIN CORE** Descrive elementi editoriali (titolo, autore ecc)

**SCHEMA.ORG** Descrive pagine web secondo uno schema semantico  
**SKOS** E’ utilizzato per descrivere e allineare terminologie diverse

**EUROPEANA**  Archivio europeo di beni culturali

**Ontologia computazionale**

Le ontologie sono la rappresentazione astratta di concetti e le loro relazioni, sono costituite da un vocabolario per descrivere la realtà + un set di assunzioni specifiche, si scrivono in OWL

**Ontologia formale** E’ rappresentata secondo un formalismo di rappresentazione

Gli elementi delle ontologie sono Classi, Proprietà, Assiomi e individui

**Tipi di ontologie**

* Ontologia TOP-LEVEL Indicano concetti comuni a tutti i domini (spazio, tempo ecc)
* Ontologia MID-LEVEL Utilizzano il livello fondazionale per descrivere concetti generali (stati fisici..)
* Domain ontologies Rappresentano concetti e le loro relazioni per un dominio specifico

**CYC** Conoscenza di senso comune

**Ontologia SUMO** E’ una mid level ontology

**Ontologia YAGOSUMO** E’ una integrazione che incorpora le entità di YAGO all’ontologia SUMO

**DBPedia** progetto per estrarre i contenuti strutturati da wikipedia

**Ontologie light weight** sono ontologie “leggere”, semplici tassionomie con poche relazioni, ad esempio Word net (ontologia strutturata in sinonimi, ipernomi, iponomi) e tesauri

**Ontologie large scale** Ontologie di grosse dimensioni tipo YAGO o DBpedia, possono essere ottenute tramite l’estrazione automatica di concetti dai testi o tramite comunità di utenti

Data la grande dimensione sono necessari strumenti di indicizzazione

**Interfaccia tra ontologie e linguaggio naturale** Per l’accesso ai concetti dell’ontologia è importante l’integrazione tra di essi e il linguaggio naturale, questa avviene tramite la DOCUMENTAZIONE, ovvero associando ogni concetto a una descrizione informale in linguaggio naturale, oppure associando ai concetti un equivalente lessicale in una risorsa linguistica esterna.

**LINKED OPEN DATA** Sono dati pubblicamente disponibili pubblicati secondo il paradigma dei linked data, compaiono in rete essendo cosi linked open data

**LOGICHE DESCRITTIVE** Sono un sottoinsieme della logica dei predicati, sono ORIENTATE ALLA CLASSIFICAZIONE e sono basate sulla relazione di SOTTOCLASSE (SUSSUNZIONE) permettono di costruire classi da classi

**TBOX E A BOX**

* TBOX E’ la definizione dei termini, la parte concettuale (Squadra ha componenti > 2)
* ABOX Sono le asserzioni sui singoli individui (Giovanni is-a uomo)

**IL LINGUAGGIO OWL**

Owl2 è il linguaggio per creare ontologie per il web semantico con un significato definito formalmente

Le ontologie scritte in owl contengono CLASSI, PROPRIETA’, INDIVIDUI, LETTERALI

**Ragionamento automatico** In owl la conoscenza espressa dall’ontologia può essere oggetto di ragionamento automatico tramite REASONER che rendono esplicita la conoscenza implicita dell’ontologia

**Struttura di un documento OWL**  owl2 non fornisce strumenti che descrivono in maniera prescrittiva la struttura di un documento, NON C’E modo di spedificare che una specifica informazione deve essere necessariamente presente (es il codice fiscale)

**Terminologia (tbox) e asserzioni (abox)** tipicamente nell’ontologia troviamo la terminologia (tbox e vocabolario) che costituisce la conoscenza generale del dominio dato. Inoltre troviamo le asserzioni (abox) che descrivono entità concrete o specifiche

**Elementi dell ontologia**

* **Entità** sono gli elementi che si riferiscono al mondo reale (Persona)
* **Assiomi** Sono le assunzioni generali contenute nell’ontologia
* **Espressioni** Sono combinazioni di entità che vanno a formare entità più complesse

**Tipi di entità**

* Oggetti -> individui
* Categorie -> Classi
* Realzioni -> Proprietà
  + - Object property: collegano individui a individui
    - Data property: assegnano un dato ad un individuo
    - Annotation property: contengono commenti e descrizioni di entità

**Assiomi** subclassOf, equivalent class ecc…

**Espressioni** Le entità possono essere combinate per formare espressioni usando dei costrutti di classi

La classe è la forma più semplice di class expression

**Restrizioni** le restrizioni sono uno dei meccanismi principali per definire nuove classi a partire da quelle esistenti, un esempio è una restrizione mediante gli operatori insiemistici (and or) o restrizioni su proprieta (esistenziale o universale o sulla cardinalità)

**ONTOLOGY ENGENEERING**

Sono dei criteri per creare ontologie interoperabili e ben fondate tramite l’utilizzo di design patter e metodologie

**METODOLOGIE**

**Ontoclean** E’ una metodologia che si basa su nozioni come essenza, identità, unità che vengono usate per caratterizzare gli aspetti dell’ontologia

Funzionamento: Si basa sulla rappresentazione delle meta proprietà, cioè le caratteristiche delle proprietà delle classi. Ogni classe ha un insieme di proprietà, e questa metodologia analizza le proprietà delle proprietà per fornire indicazioni sulle classi

Le proprietà di ontoclean sono IDENTITA (I) Proprieta che identifica il tipo di oggetto (es per il triangolo il num di lati)

UNITA’ (U) Proprietà di un tipo di oggetto di essere unitario (persona oppure

Pattumiera)

RIGIDITA’ (R) Proprietà di un tipo di oggetto che non è soggetta ai

Cambiamenti, es persona ha la proprietà studente,

studente è anti-rigida perché puo cambiare

DIPENDENZA (D) E’ la proprietà di un tipo di oggetto di dipendere da un altro

Esempio studente dipende da scuola

Meta proprietà e sussunzione, vengono tutte ereditate tranne R

**Neon** E’ una metodologia orientata agli aspetti collaborativi nello sviluppo di ontologie, ha 9 scenari

**PATTERN**

Sono dei pattern per creare ontologie (es. List)

**Dolce** E’ una ontologia basata tra distinzione di ENDURANTI e PERDURANTI

Gli enduranti sono sempre presenti in ogni momento in cui sono presenti (partecipante) //no temporale

I perduranti si estendono nel tempo, solo una parte di loro è presente (evento) // hanno natura temporale

**Provenance** E’ una ontologia per scambio di informazioni sulla provenienza delle entità, descrive l’origine delle entità

**SKOS**

Serve ad organizzare le risorse nei linked data, è una sorta di concorrente di OWL, serve ad uniformare la pubblicazione di vocabolari (è alternativo a owl) E’ basato sulla nozione di “Concetto”, si identifica un concetto con un IRI, e si linka ad altri concetti

**SWRL**

Sono delle regole in formato if-then, NON POSSONO CREARE NUOVI INDIVIDUI O CLASSI, solo asserire nuove proprietà

REGOLE DICHIARATIVE: Il conseguente è una nuova asserzione

REGOLE DI PRODUZIONE: Il conseguente è una azione (hai il raffreddore -> prendi una aspirina)

NB non è una inferenza quindi le modifiche vanno cancellate a mano

**SPARQL interrogazione dei grafi rdf**

Sparql permette di restituire delle triple che rispondono a una query

La struttura è FROM da che dataset vogliamo estrarre SELECT dei dati che ci interessano WHERE le condizioni sotto forma di triple e infine i MODIFICATORI tipo group by o altri

**DB PEDIA E WIKIDATA**

**DBPEDIA**

DBPedia è stata creata tramite community che estraggono i dati da wikipedia, si mappa l’infobox verso una unica ontologia tramite degli extractor

Mapping: i template di wikipedia vengono mappati su una ontologia

**WIKIDATA**

E’ un database collaborativo basato sui contenuti di wikipedia, le informazioni sono etichettati con l’origine da cui provengono

**I LINKED DATA**

Linked data è un modo per pubblicare dati strutturati in modo che possano essere collegati e effettuate query si usa un URI per identificare l’entità, ci sono chiamate http per trovare le entità, si usano standard come rdf e sparql.

Le risorse devono essere dereferenziate = all’uri di una risorsa si trova la risorsa stessa

**Linked data platform** E’ una piattaforma per l’accesso ai linked data che usa il paradigma rest

**Perché puo capitare che la gerarchia delle classi inferita sia diversa da quella asserita?**

Perché puo succedere che l’insieme degli individui di una classe è totalmente incluso anche in quello di un’altra classe

**Cos’è una classe definita**

Una classe definita è un tipo di classe che viene definito come equivalente a un insieme di restrizioni, le restrizioni sono le condizioni di appartenenza alla classe

**Cosa è una class expression**

Una class expression identifica l’insieme degli individui grazie alla definizione di condizioni che devono essere soddisfatte dalle proprietà degli indivizui (si creano usando connettivi booleani, quantificatori..)

**Cosa è un named graph**

E’ un grafo di una rete semantica che ha un nome, in turtle si usano i prefissi cosi si possono usare reti proventienti da piu fonti

**Cosa è il default graph**

E’ un frafo a cui non è stato assegnato un nome