Ontology Engineering

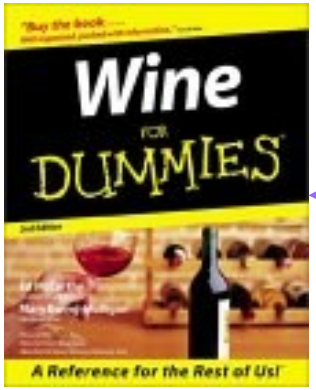
"Ontology Development 101: A Guide to Creating Your First Ontology"
by Natalya F. Noy and Deborah L. McGuinness
http://protege.stanford.edu/publications/ontology_development/ontology101.html

Which wine should I serve with seafood today?

A shared ONTOLOGY of wine and food

French wines and wine regions

California wines and wine regions



Outline

- **Che cos'è un'ontologia?**
- Perché sviluppare un'ontologia?
- Step-By-Step: Sviluppare un'ontologia
- Problemi comuni e soluzioni

Cos'è un'ontologia

- Una ontologia è una descrizione esplicita di un dominio:
  - concetti
  - Proprietà e attributi dei concetti
  - Vincoli su proprietà e attributi
  - Individui *(spesso, ma non sempre)*
- Una ontologia definisce
  - Un vocabolario comune
  - Un intendimento condiviso

Esempi di ontologia

- Tassonomie sul Web
  - Yahoo! categories
- Cataloghi per lo shopping online
  - Catalogo prodotti Amazon.com
- Terminologia standard Domain-specific
  - Unified Medical Language System (UMLS)
  - UNSPSC – terminologia per prodotti e servizi

Cos'è l'"Ontology Engineering"?

Ontology Engineering: Definire i termini all'interno di un dominio e le relazioni tra essi

- Definire i concetti nel dominio (classes)
- Organizzare i concetti in una gerarchia (gerarchia subclass-superclass)
- Definire quali attributi e proprietà possano avere le classi e i vincoli su i loro valori
- Definire gli individui e riempire i valori delle proprietà

Outline

- Che cos'è un'ontologia?
- **Perché sviluppare un'ontologia?**
- Step-By-Step: Sviluppare un'ontologia
- Problemi comuni e soluzioni

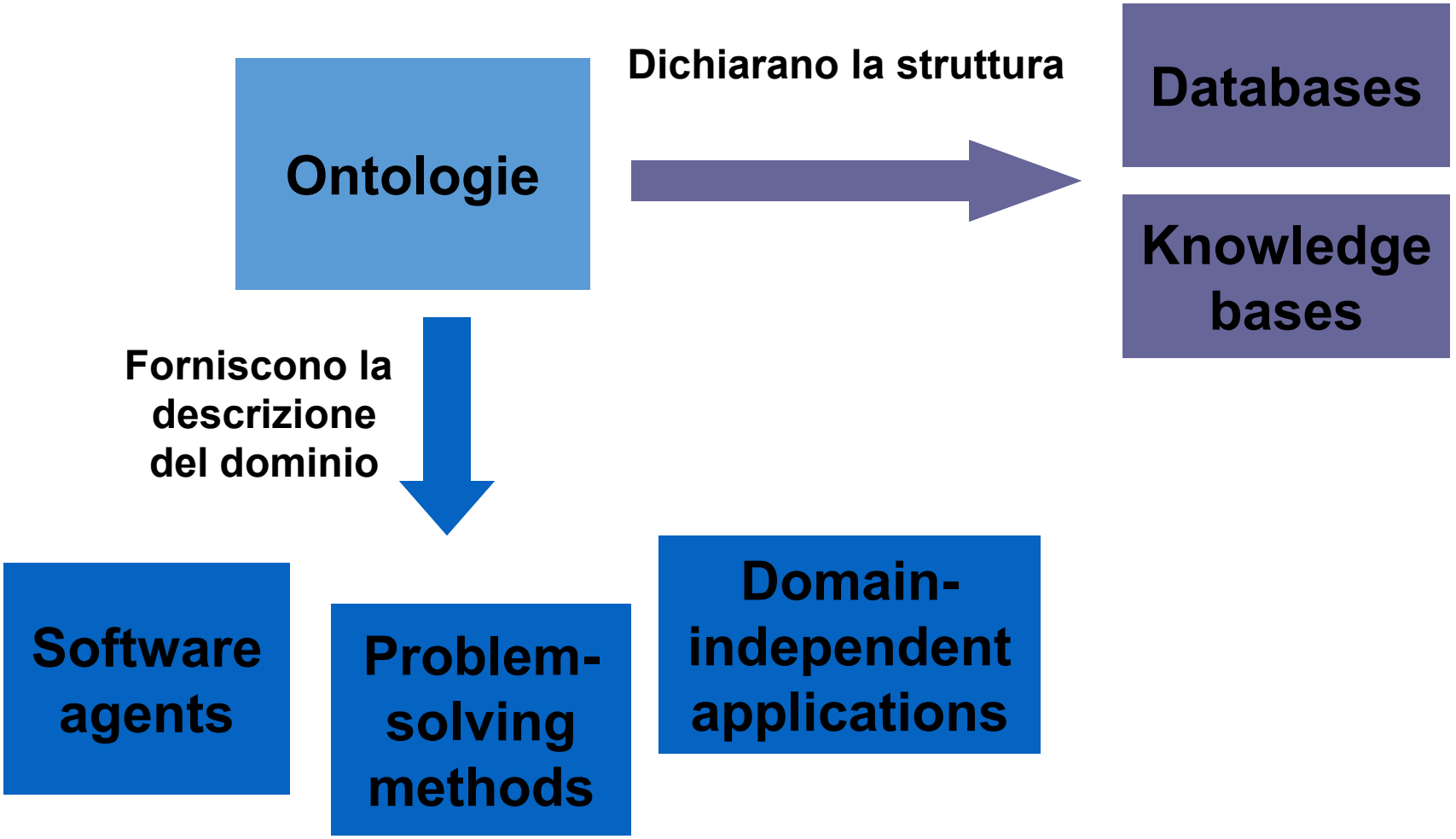
Perché sviluppare un'ontologia?

- Per condividere una comprensione comune della struttura di un'informazione
  - Tra le persone
  - Tra gli agenti software
- Per abilitare il **riuso** della conoscenza di dominio
  - Per evitare di «re-inventare la ruota»
  - Per introdurre standard per consentire l'interoperabilità

Altre ragioni

- Per rendere **esplicite** delle assunzioni all'interno di un dominio
  - Più facile cambiare le assunzioni (considerare una knowledge base in ambito genetico)
  - Più facile capire e aggiornare dati legacy
- Per **separare** la conoscenza di dominio dalla conoscenza operativa
  - Riusare le conoscenze di dominio e operative in modo separato (es., configurazione basata su vincoli)

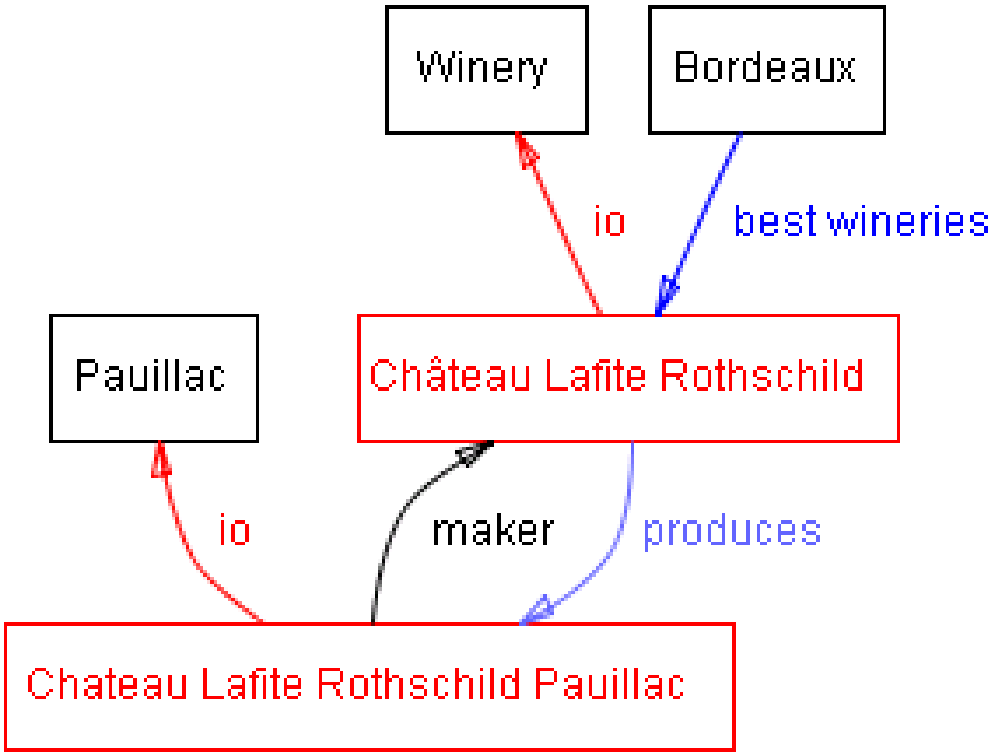
Spesso l'Ontologia è solo l'inizio



Outline

- Che cos'è un'ontologia?
- Perché sviluppare un'ontologia?
- **Step-By-Step: Sviluppare un'ontologia**
- Problemi comuni e soluzioni

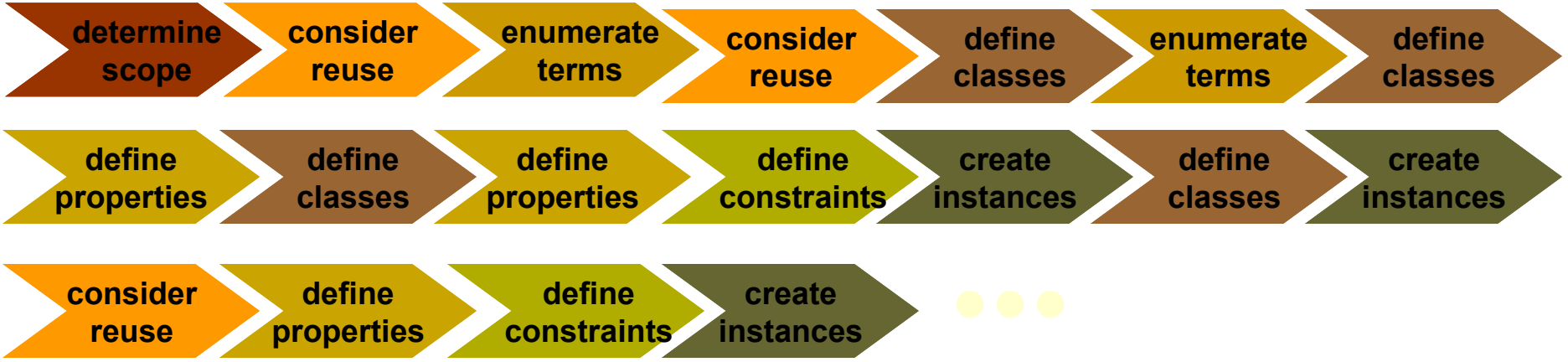
Vini e cantine



Processo di sviluppo di un'Ontologia

In questa presentazione:

Nella realtà – un processo iterativo:



Ontology Engineering versus Modellazione Object-Oriented

Ontologia

- Rispecchia la struttura del **mondo**
- È incentrata sulla **struttura** dei concetti
- **Non** si occupa della rappresentazione fisica delle informazioni

Struttura classi OO

- Rispecchia la struttura dei **dati e del codice**
- È incentrata sul **comportamento** (metodi)
- Descrive la **rappresentazione fisica dei dati** (long int, char, etc.)

Strumenti

- Protégé:
  - Uno strumento di sviluppo di ontologie grafico
  - Supporta un modello di conoscenza ricco
  - È open-source e disponibile gratuitamente (http://protege.stanford.edu)

Determinare il Dominio e l'Ambito (scope)

- Qual è il dominio che l'ontologia andrà a rappresentare?
- In che modo andremo ad utilizzare l'ontologia?
- Per quali tipi di domande le informazioni contenute nell'ontologia dovranno fornire risposte (**domande di competenza**)?

Le risposte a queste domande potrebbero cambiare durante in ciclo di vita dell'ontologia

Domande di competenza

- Che caratteristiche del vino devo prendere in considerazione quando scelgo un vino?
- Il Bordeaux è un vino rosso o bianco?
- Il Cabernet Sauvignon si sposa bene con i frutti di mare?
- Qual è la miglior scelta per una grigliata di carne?
- Quali caratteristiche del vino influenzano la sua appropriatezza per una pietanza?
- Il sapore o la corposità di un vino cambiano a seconda dell'annata?
- Quali sono state le buone annate per lo Zinfandel della California?

Considerare il riuso

- Perché riusare altre ontologie?
  - Per non specare le **forze**
  - Per **interagire** con strumenti che utilizzano altre ontologie
  - Per utilizzare ontologie che **sono state validate** mediante l'uso in altre applicazioni

Cosa riutilizzare?

- General ontologies
  - DMOZ (www.dmoz.org)
  - WordNet (www.cogsci.princeton.edu/~wn/)
- Domain-specific ontologies
  - UMLS Semantic Net
  - GO (Gene Ontology) (www.geneontology.org)

Enumerare i Termini importanti

- Quali sono i termini che dobbiamo trattare?
- Quali sono le proprietà di questi termini?
- Cosa vogliamo dire riguardo questi termini?

Elencare i Termini – L'Ontologia dei Vini

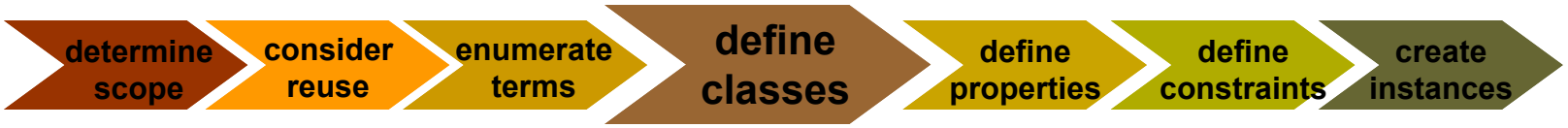
Vino, uva, cantina, posizione

Colore del vino, corpo del vino, sapore del vino, contenuto di zucchero

Vino bianco, vino rosso, vino Bordeaux

Cibo, frutti di mare, pesce, carne, verdura, formaggio

Definire le Classi e la gerarchia delle classi



- Una classe è un **concetto** nel dominio
  - Una classe dei vini
  - Una classe delle cantine
  - Una classe dei vini rossi
- Una classe è una **collezione** di elementi con proprietà simili
- **Istanze** delle classi
  - Un bicchiere del vino della California

Gerarchia di classi

- Le classi di solito costituiscono una gerarchia tassonomica (gerarchia superclasse-sottoclasse)
- Una gerarchia di classi è spesso una gerarchia IS-A:

Una istanza di una sottoclasse è una istanza della superclasse

Se si pensa ad una classe come un **insieme** di elementi, la sottoclasse è un **sottoinsieme**

Ereditarietà - esempi

- Mela è sottoclasse di Frutto

  Ogni mela è un frutto

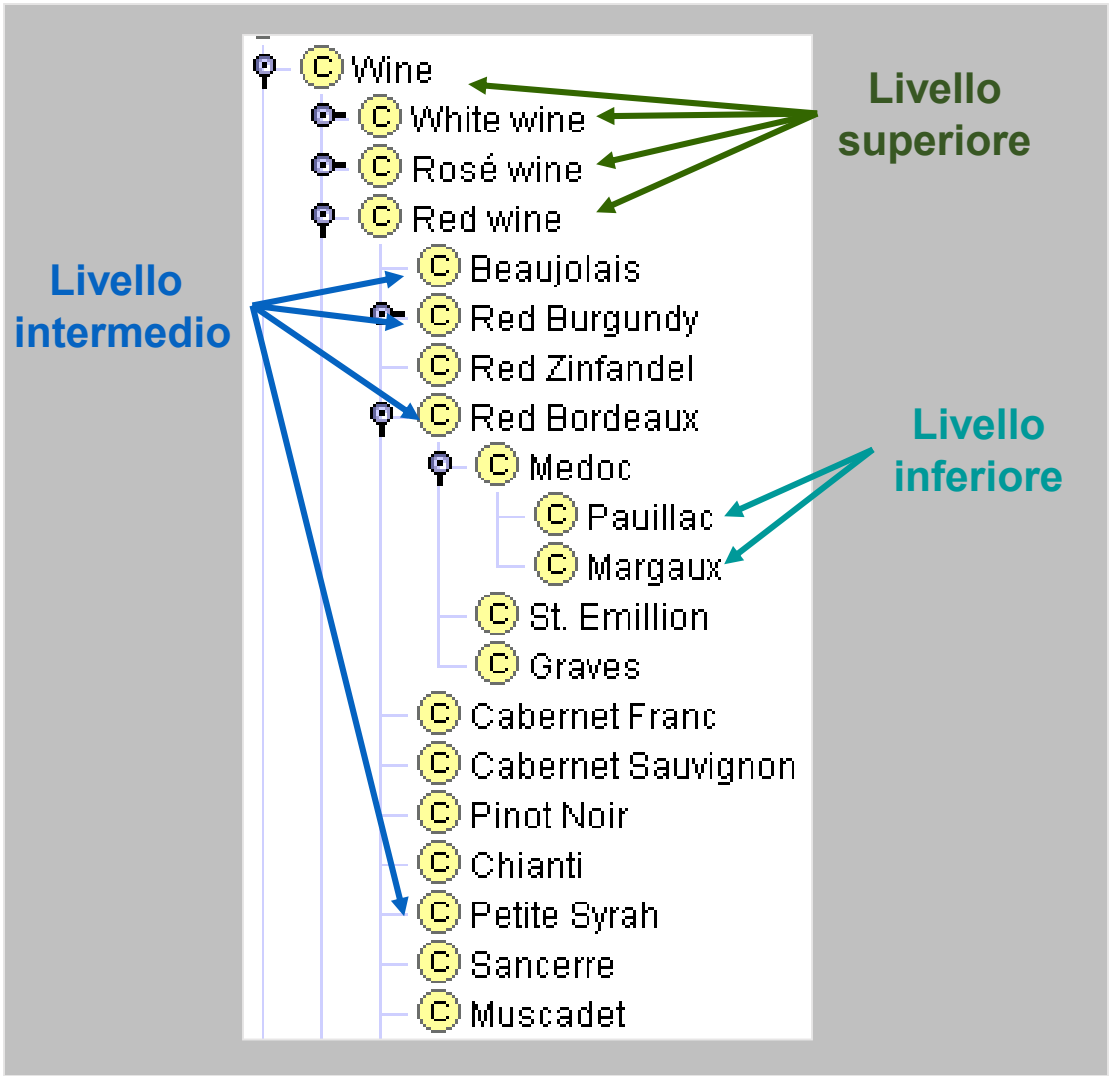
- I Vini Rossi sono sottoclasse di Vino

  Ogni vino rosso è un vino

- Il Chianti è sottoclasse di Vino Rosso

  Ogni Chianti è un vino rosso

Livelli nella gerarchia



Modalità di sviluppo

- top-down – definire prima i concetti più generali e poi specializzarli
- bottom-up – definire prima i concetti più specifici e poi organizzarli in classi più generali
- misto – definire prima i concetti salienti e poi generalizzarli o specializzarli

Documentazione

- Le Classi (e le proprietà) spesso hanno una documentazione
  - Che descrive la classe in linguaggio naturale
  - Che elenca le assunzioni rilevanti per la definizione della classe
  - Che elenca i sinonimi
- Documentare le classi e le proprietà è importante tanto quanto documentare un codice di programmazione!

Definire le Proprietà delle Classi

- Le proprietà descrivono gli attributi delle istanze della classe e le relazioni con altre istanze

  Ciascun vino avrà un colore, un contenuto di zucchero, un produttore, ecc.

Proprietà

- Tipi di proprietà
  - “intrinseche”: sapore e colore del vino
  - “estrinseche”: nome e prezzo del vino
  - parti: ingredienti di una pietanza
  - Relazioni con altri oggetti: produttore del vino (cantina)
- Proprietà semplici e composte
  - Semplici (attributi): contengono valori primitivi (stringhe, numeri)
  - Complesse : contengono (o puntano a) altri oggetti (es., un’istanza di cantina)

Ereditarietà delle Classi

- Una sottoclasse **eredita tutte le proprietà** dalla superclasse

  Se un vino ha un nome e un sapore, un vino rosso avrà anch'esso un nome e un sapore

- Se una classe ha più superclassi, erediterà le proprietà di ciascuna di esse

  Il Porto è sia un vino da dessert che un vino rosso. Eredita la proprietà «sugar content: high» dalla prima classe e «color:red» dalla seconda

Vincoli delle prorietà

- I vincoli delle proprietà descrivono o limitano l'insieme dei possibili valori attribuibili ad una proprietà
  - *Il nome di un vino è una stringa*
  - *Il produttore di un vino è un'istanza di Cantina*
  - *Una cantina ha esattamente una Posizione*

Vincoli comuni

- Cardinalità – il numero di valori che può assumere una proprietà
- Value type – il tipo di valori che può assumere una proprietà
- Minimum and maximum – un intervallo di valori per una proprietà numerica
- Default value – il valore predefinito di una proprietà se non ne viene specificato alcuno

Cardinalità

- Cardinality
  - Cardinalità N significa che la proprietà **deve** avere N valori
- Minimum cardinality
  - Cardinalità Minima 1 significa che la proprietà deve avere un valore (*obbligatorio*)
  - Cardinalità Minima 0 significa che il valore della proprietà è *opzionale*
- Maximum cardinality
  - Cardinalità Massima 1 significa che la proprietà deve avere al più un valore (single-valued slot)
  - Cardinalità Massima maggiore di 1 significa che la proprietà può avere più di un valore (multiple-valued slot)

Tipi di dato

- String: una stringa di caratteri (“Château Lafite”)
- Number: un integer o un float (15, 4.5)
- Boolean: true/false
- Enumerated type: una lista di valori consentiti (high, medium, low)
- Complex type: un’istanza di un’altra classe
  - Specificare la classe a cui l’istanza appartiene

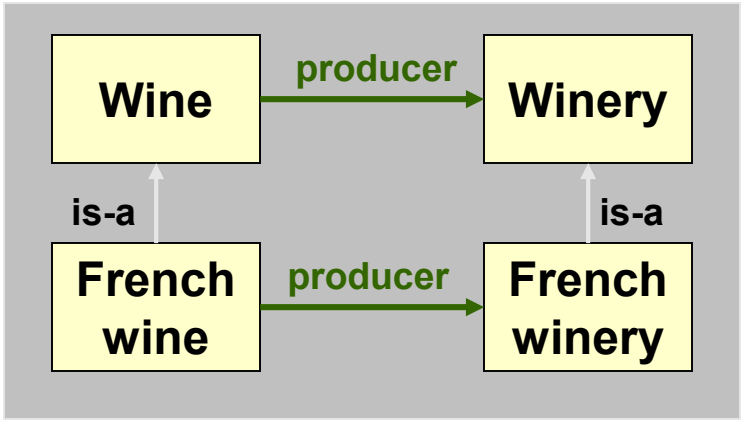
  La classe Vino è un Value Type per la proprietà «produce» della classe Cantina

Dominio e Range di una relazione

- Dominio – la classe (o le classi) che possiedono la relazione
  - Precisamente: la classe (o le classi) le cui istanze possono essere soggetto della relazione
- Range – la classe (o le classi) a cui appartengono i valori della relazione

Ereditarietà delle classi

- Una sottoclasse **eredita** tutte le proprietà della superclasse
- Una sottoclasse può fare **l'override** dei vincoli per «restringere» la lista dei valori ammissibili
  - Rendere la cardinalità più piccola
  - Sostituire una classe nel range con una sottoclasse



Creare le Istanze

- Creare un'istanza di una classe
  - La classe diventa un **tipo diretto** dell'istanza
  - Ogni superclasse del tipo diretto è un **tipo** dell'istanza
- Assegnare i valori delle proprietà per la porzione di istanza
  - I valori delle proprietà devono conformarsi ai vincoli
  - Gli strumenti di Knowledge-acquisition spesso effettuano una validazione in tal senso

Outline

- Che cos'è un'ontologia?
- Perché sviluppare un'ontologia?
- Step-By-Step: Sviluppare un'ontologia
- **Problemi comuni e soluzioni**

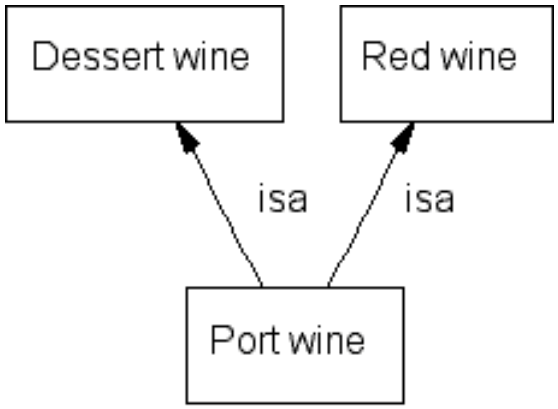
Definizione delle classi e della gerarchia di classi

- Da ricordare:
  - Non vi è una sola gerarchia corretta
  - Vi sono alcune linee guida
- La domanda da porsi:

  “Ciascuna istanza della sottoclasse è istanza della sua superclasse?”

Ereditarietà multipla

- Una classe può avere più di una superclasse
- Una sottoclasse eredita le proprietà e le restrizioni da tutti i suoi genitori
- Sistemi differenti risolvono i conflitti in modo differente

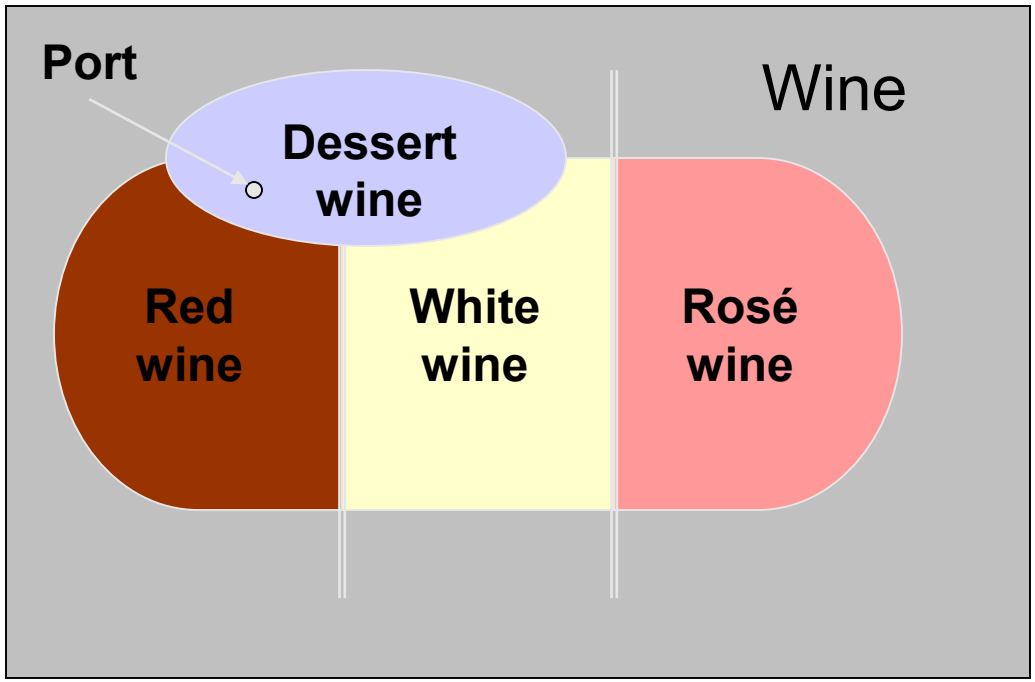


Classi Disgiunte

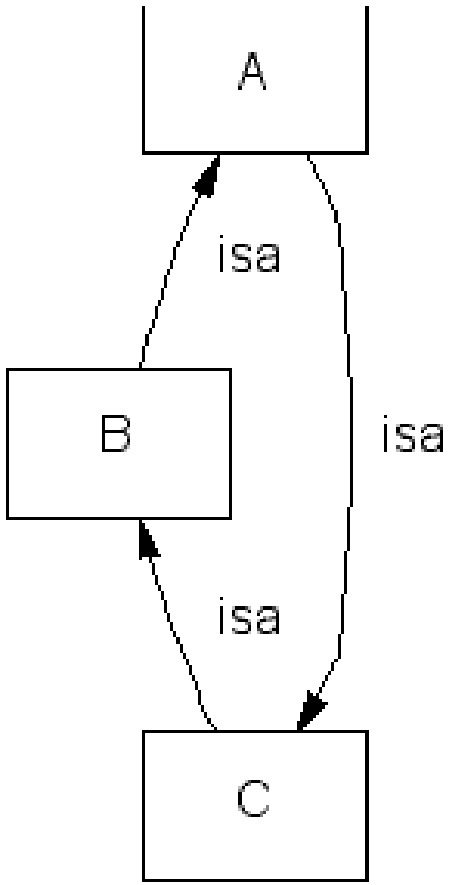
- Le classi sono **disgiunte** se non possono avere istanze in comune
- Classi disgiunte non possono altresì avere **sottoclassi** in comune

Red wine, White wine, Rosé wine sono disgiunte

Dessert wine e Red wine non sono disgiunte

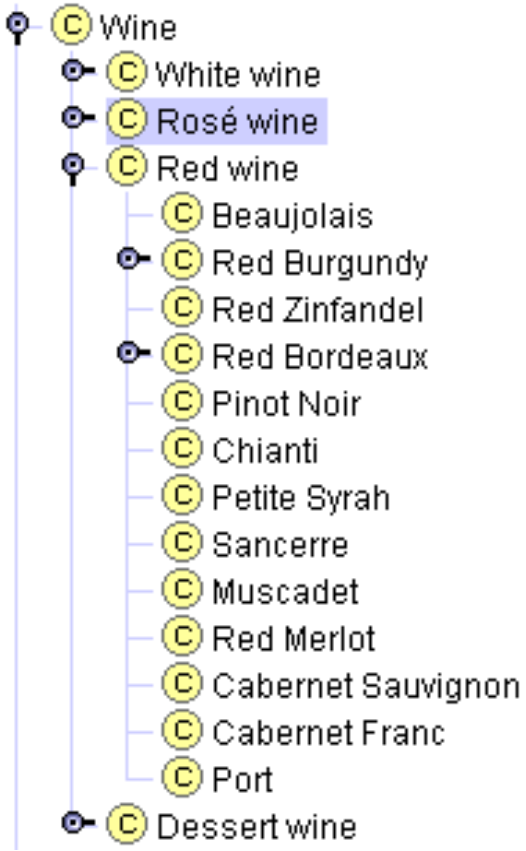


Evitare i cicli tra le classi



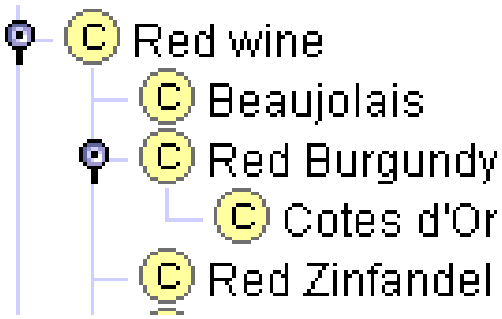
- Pericolo di ereditarietà multipla: cicli nella gerarchia delle classi
- Le classi A, B, e C hanno insiemi di istanze equivalenti
  - A, B, e C sono quindi equivalenti

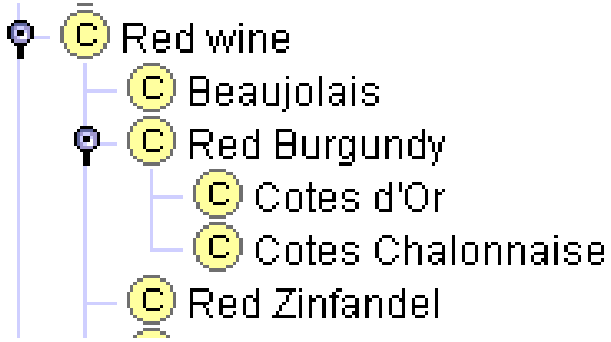
Fratelli (Sibling) in una gerarchia di classi



- Tutti i fratelli in una gerarchia di classi devono essere allo stesso livello di generalità
- Pensare ai capitoli, paragrafi e sottoparagrafi di un libro

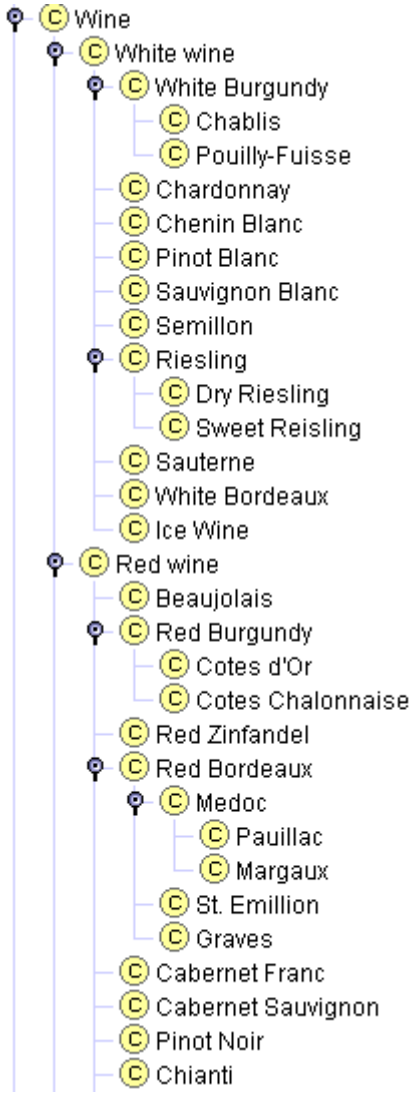
Dimensione perfetta delle classi





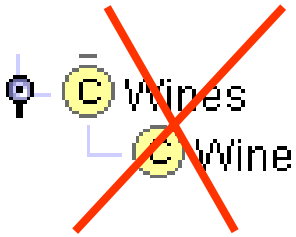
- Se una classe ha solo un figlio, potrebbe esserci un problema di modellazione
- Perché, ad es. introdurre una gerarchia se l'unico vino Burgundy rosso che abbiamo a disposizione è il «Côtes d'Or»?
- Pensare ai punti elenco di una lista puntata

Dimensione perfetta delle classi (II)

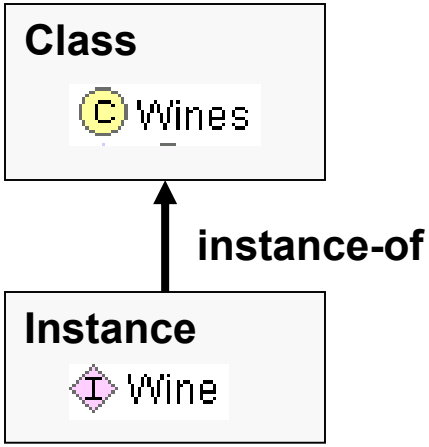


- Se una classe ha più di una dozzina di figli, potrebbero essere necessarie delle sottocategorie
- Tuttavia, se non esistono classificazioni naturali, la lista lunga piò essere più naturale

Nomi delle classi: Singolare o Plurale



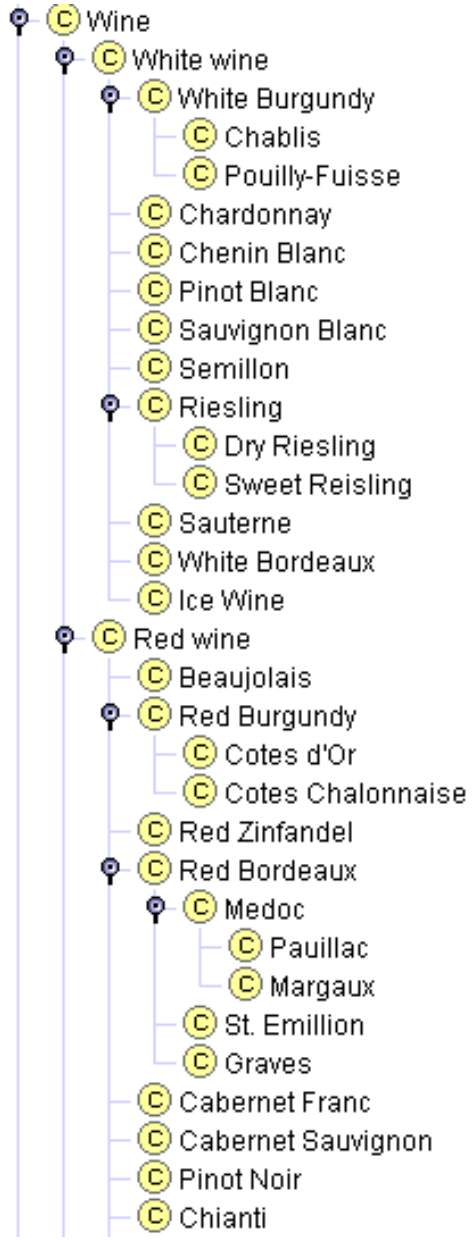
- Un «vino» non è un tipo di «vini»
- Un «vino» è **un'istanza** della classe Vini
- I nomi delle classi dovrebbero essere o
  - Tutti singolari
  - Tutti plurale



Classi e nomi delle Classi

- Le classi rappresentano i **concetti** nel dominio, **non i loro nomi**
- Il nome della classe può cambiare, ma farà riferimento sempre allo stesso concetto
- **Nomi sinomini** per lo stesso concetto non sono classi differenti
  - Molti sistemi consentono l'elenco dei sinonimi come parte della definizione della classe

Gerarchia dei vini completa



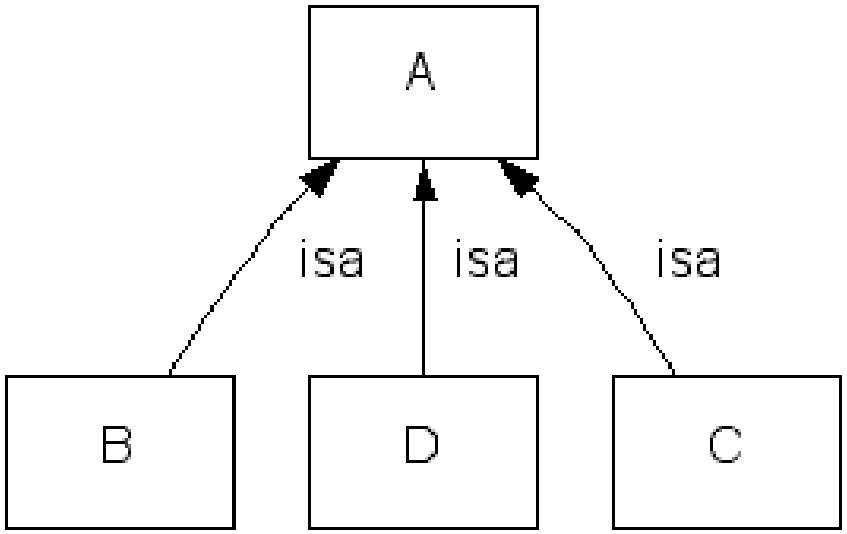
Domain e Range

- Per definire il dominio o il range di una relazione, trovare **la classe più generale** o le classi
- Considerare la relazione **sapore**
  - Domain: Red wine, White wine, Rosé wine
  - Domain: Wine
- Considerare la relazione **produce** per una **Cantina**:
  - Range: Red wine, White wine, Rosé wine
  - Range: Wine

Dominio e Range delle proprietà

- Quando si definisce dominio e range di una proprietà, fare riferimento alla classe (o alle classi) più
- Considerando la proprietà gusto
  - Domain: Red wine, White wine, Rosé wine
  - Domain: Wine
- Considerando la proprietà produce della classe Vineria:
  - Range: Red wine, White wine, Rosé wine
  - Range: Wine

Definire Dominio e Range



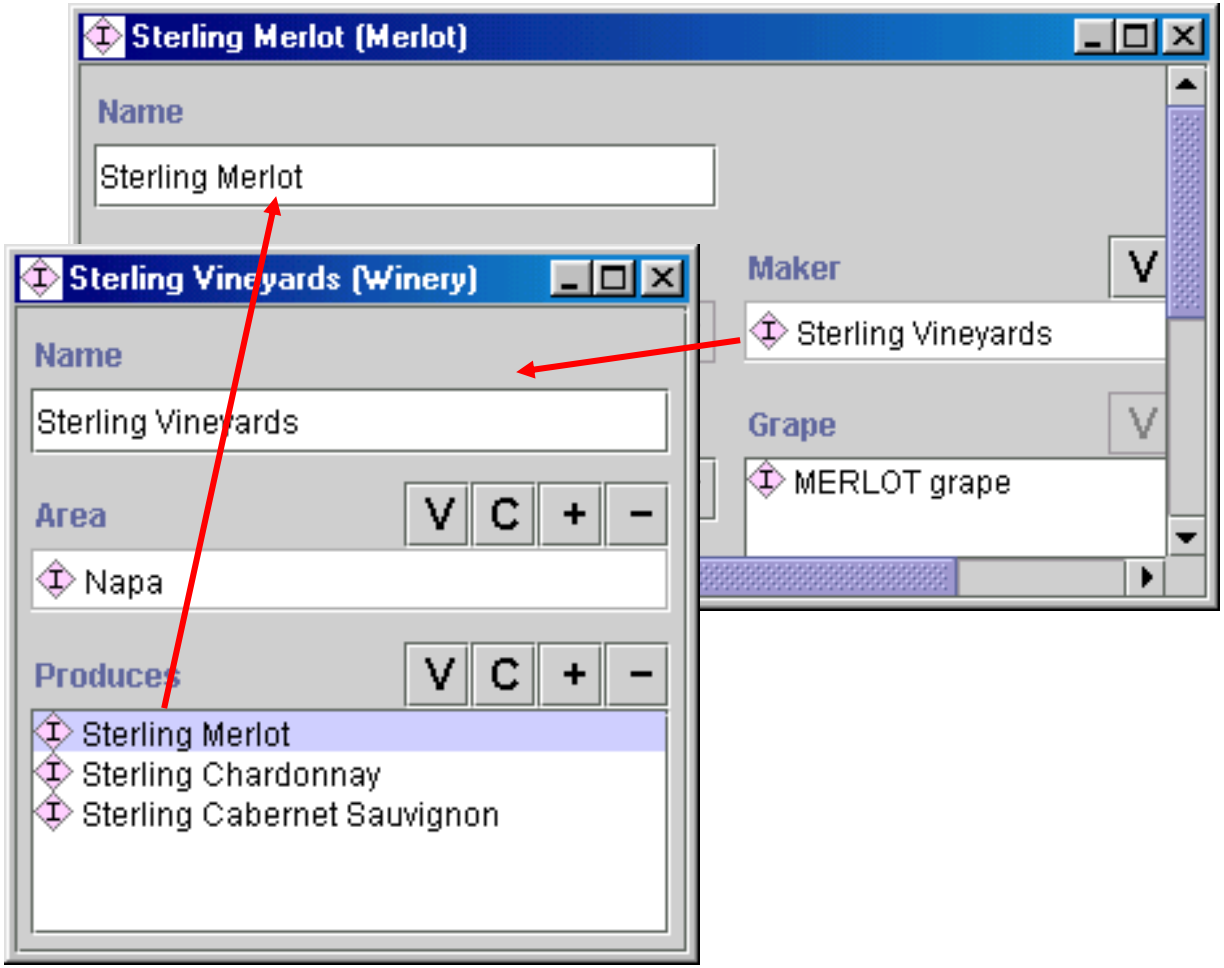
- Una classe e una superclasse –> sostituire con la superclasse
- Tutte le sottoclassi di una classe –> Sostituire con la superclasse
- La maggior parte delle sottoclassi di una classe -> valutare di sostituire con la superclasse

Proprietà inverse

Maker e

Producer

sono proprietà inverse



Relazioni inverse

Maker e

Producer

Sono relazioni inverse

- Le relazioni inverse contengono informazioni ridondanti, ma
  - Consentono l'acquisizione dell'informazione in entrambe le direzioni
  - Abilitano verifiche aggiuntive
  - Consentono la presentazione delle informazioni in entrambe le direzioni
- L'**implementazione** effettiva differisce da sistema a sistema
  - Vengono memorizzati entrambi i valori?
  - Quando vengono inseriti i valori inversi?
  - Cosa accade se cambiamo il link di una proprietà inversa?

Default Values

- Default value – è il valore predefinito che assume la proprietà quando viene creata un'istanza (un individuo)
- Un default value può essere cabiato
- È un valore comune per la proprietà, ma non è un valore obbligatorio
- Per esempio, il default value il corpo (la struttura) di un vino può essere PIENO

Limitare lo Scope

- Un'ontologia non deve contenere tutte le possibili informazioni correlate ad un dominio
  - Non vi è necessità di specializzare o generalizzare più di quanto richiesto dall'applicazione
  - Non vi è necessità di includere tutte le possibili proprietà di una classe
    - Solo le proprietà salienti
    - Solo le proprietà richieste dall'applicazione
- L'ontologia dei vini, dei cibi, etc. probabilmente non includerà concetti tipo
  - Dimensione della bottiglia
  - Colore dell'etichetta
  - Il mio cibo e vino preferito

Outline

- What is an ontology?
- Why develop an ontology?
- Step-By-Step: Developing an ontology
- Going deeper: Common problems and solutions
- Ontologie nei linguaggi del Semantic Web
- Current research issues in ontology engineering

Ontologie e linguaggi del SW

- La maggior parte dei linguaggi del Semantic Web sono progettati esplicitamente per rappresentare le ontologie
  - RDF Schema
  - DAML+OIL
  - SHOE
  - XOL
  - XML Schema

SW Languages

- I linguaggi differiscono per
  - sintassi
  - terminologia
    - Class-concept
    - Instance-object
    - Property
  - espressività
    - Ciò che possiamo esprimere in alcuni linguaggi, non possiamo farlo in altri
  - semantica
    - La stessa istruzione potrebbe significare cose diverse in linguaggi diversi

RDF

- RDF: Resource Description Framework
- RDF è un framework per la descrizione di risorse sul WEB
- RDF è progettato per essere letto e interpretato da computer
- RDF NON è progettato per essere facilmente interpretabile dalle persone
- RDF è scritto in XML

RDF Resource, Property, e Property Value

- RDF identifica le risorse tramite identificatori (URIs), descrive le risorse tramite proprietà e valori.
- Una **Resource** è qualsiasi cosa possa avere un URI, ad esempio "http://www.w3schools.com/rdf"
- Una **Property** è una risorsa che ha un nome, ad es "author" o "homepage"
- Un **Property value** è il valore di una Property, ad es. "Jan Egil Refsnes" o "http://www.w3schools.com" (un valore può a sua volta essere un'altra risorsa)
- Esempio: descrizione della risorsa "http://www.w3schools.com/rdf":

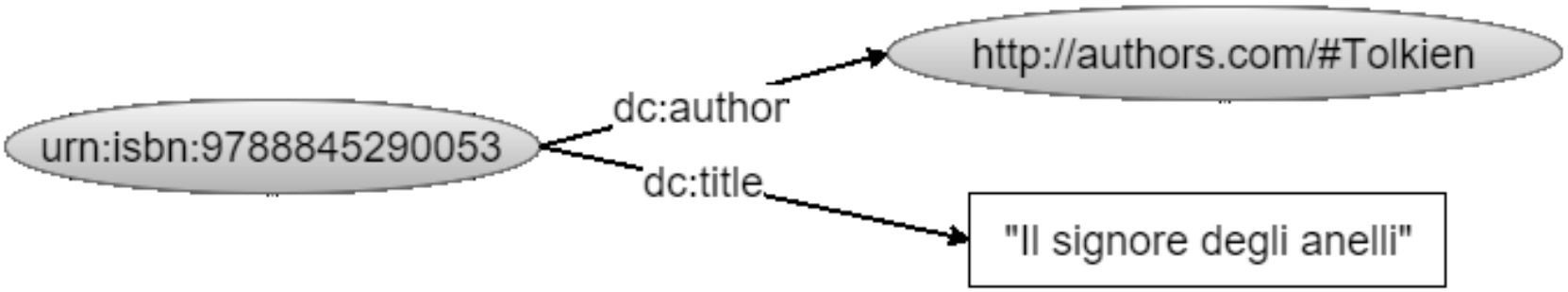
```
<?xml version="1.0"?>
<RDF>
  <Description about="http://www.w3schools.com/rdf">
    <author>Jan Egil Refsnes</author>
    <homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

RDF Statement

- La combinazione di una Risorsa, una Proprietà e un Valore (detti anche **soggetto, predicato e oggetto** dello Statement) costituisce uno **Statement**.
- Esempio: “L’autore di http://www.w3schools.com/rdf è Jan Egil Refsnes".
  - Il soggetto dello statement precedente è: http://www.w3schools.com/rdf
  - Il predicato è: author
  - L’oggetto è: Jan Egil Refsnes

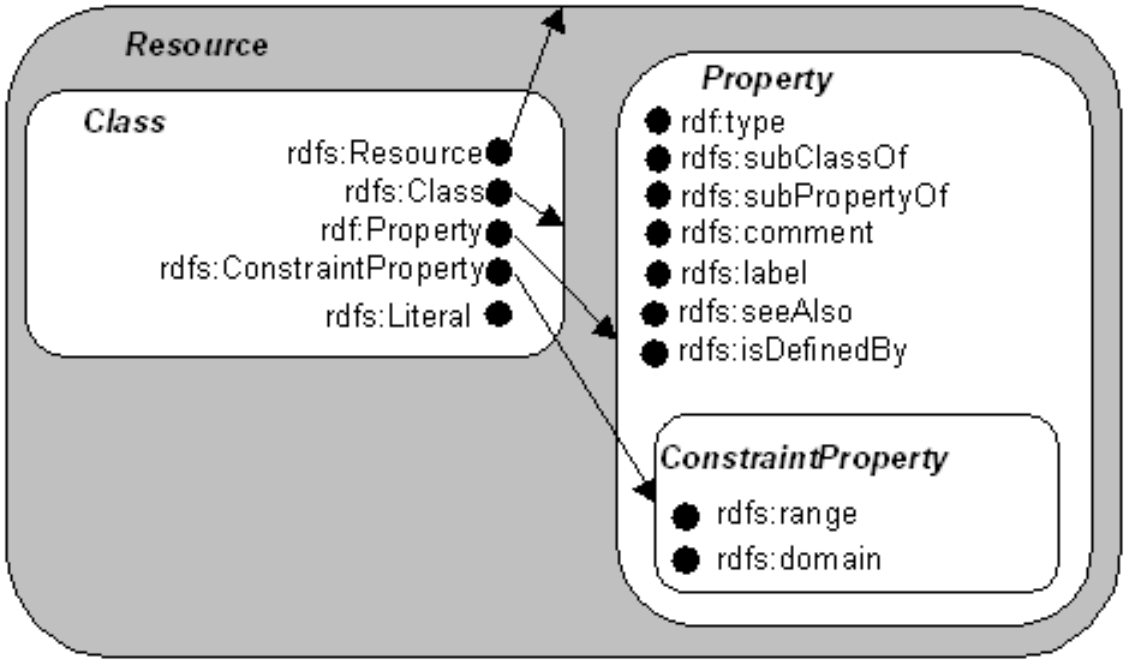
- L’elemento **<rdf:Description>** contiene la descrizione della risorsa identificata dall’attributo **rdf:about**.

ESEMPIO




```
<rdf:Description rdf:about="urn:isbn:9788845290053">
      <dc:title> Il signore degli Anelli </dc:title>
      <dc:auhor> http://authors.com/#Tolkien</dc:author>
</rdf:Description>
```

RDF and RDF Schema Classes



RDF Schema Specification 1.0 (http://www.w3.org/TR/2000/CR-rdf-schema-20000327/)

RDF(S) Terminology and Semantics

- Classes and a class hierarchy
  - Tutte le classi sono istanze di rdfs:Class
  - Una gerarchia di classi è definita da rdfs:subClassOf
- Istanze di una classe
  - Definite da rdf:type
- Proprietà
  - Le proprietà sono globali:

    Un proprietà nome corrisponde alla stessa proprietà nome utilizzata da qualche altra parte (assumendo lo stesso namespace)
  - Anche le proprietà costituiscono ina gerarchia (rdfs:subPropertyOf)

Property Constraints in RDF(S)

- Vincoli di cardinalità
  - Nessun vincolo esplicito di cardinalità
  - Ciascuna proprietà può avere più valori
- Range di una property
  - Una proprietà può avere solo un range
- Domain of a property
  - Una proprietà può avere più di un dominio (puà essere agganciata a una o più classi)
- Nessun valore di default

SPARQL

- Linguaggio di interrogazione per RDF
- **S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage
- Sintassi SQL-LIKE
- Utilizza i concetti di triple e grafo
- Una query è la ricerca del sottografo rdf corrispondente alle triple richieste dall'utente
- Esempio: Trovare il titolo del libro "urn:isbn:9788845290053"

```
PREFIX: dc=
"http://dublincore.org/documents/dcmi-namespaces"

SELECT ?title
WHERE <urn:isbn:9788845290053> dc:title
```

ELEMENTI DI SPARQL

- Graph Pattern
  - Grouping
  - Optional
  - Union
  - Filter
- Tipi di query
  - Select
  - Construct
  - Describe
  - Ask

ANTICIPO: TIPICA QUERY SPARQL

```
prefix
SELECT

FROM

WHERE
```

- Prefisso per gli uri
- Variabili che si vogliono visualizzare
- File rdf o grafo
- Condizioni: graph pattern, filtri, ordinamento