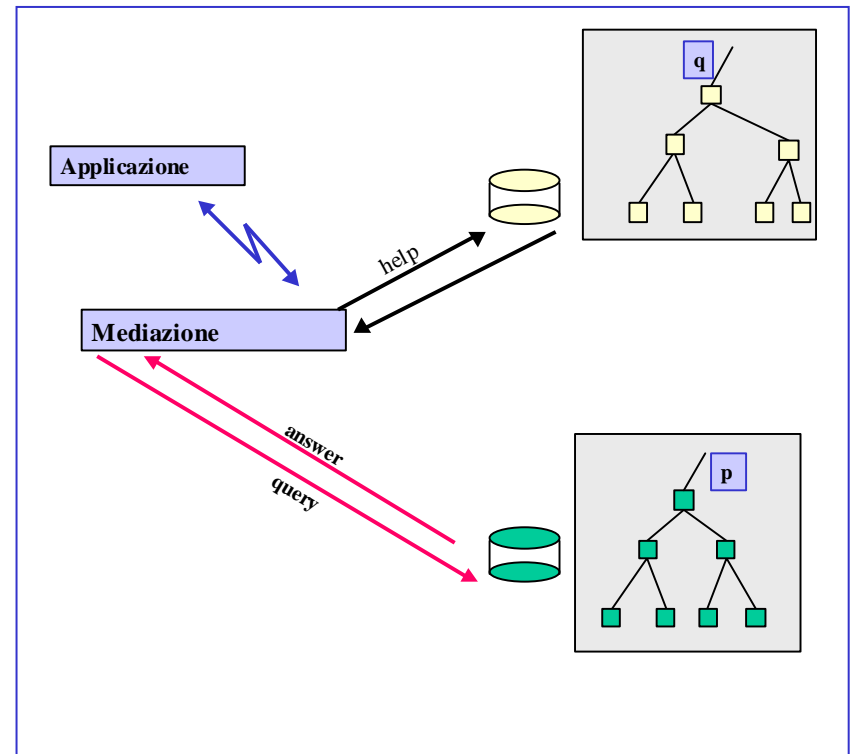
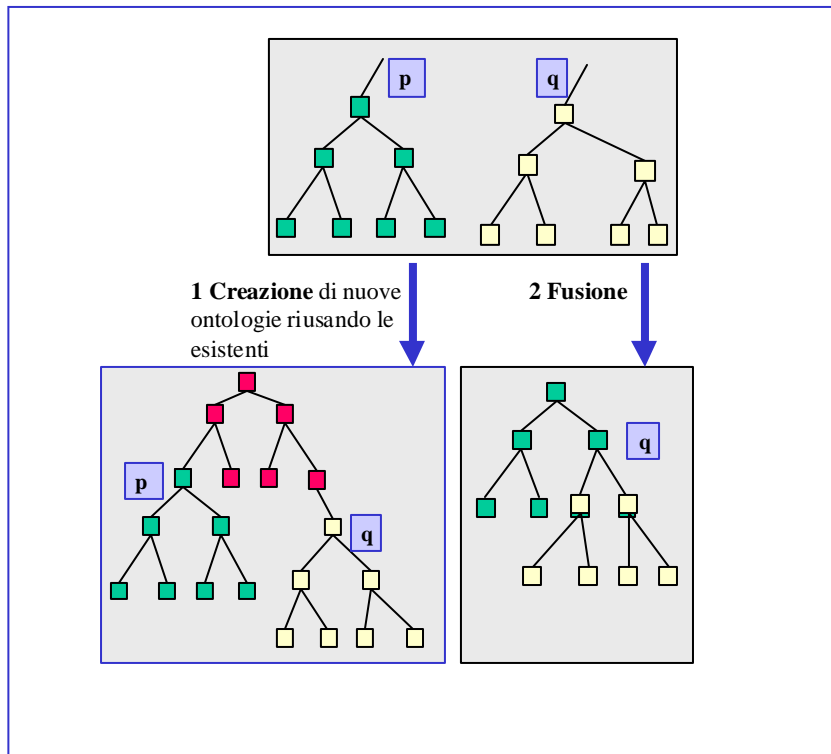
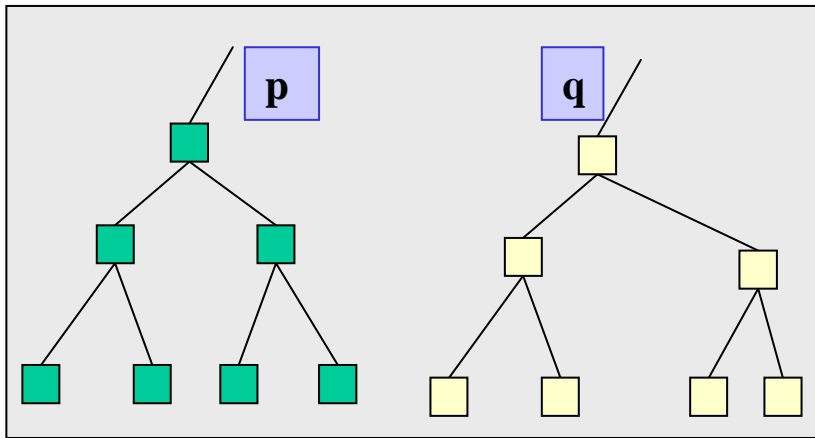


# Ontologie: integrazioni e mediazioni

Lezione del Corso Interazione Uomo Macchina 2, Docente Francesco Mele  
Corso di Laurea in Informatica Università di Napoli Federico II,  
Anno Accademico 2004-2005



# Integrazione di ontologie (S. Pinto)



**1 Creazione di nuove ontologie riusando le esistenti**

**2 Fusione**

**3 Uso**

**Applicazione A**

**Applicazione B**

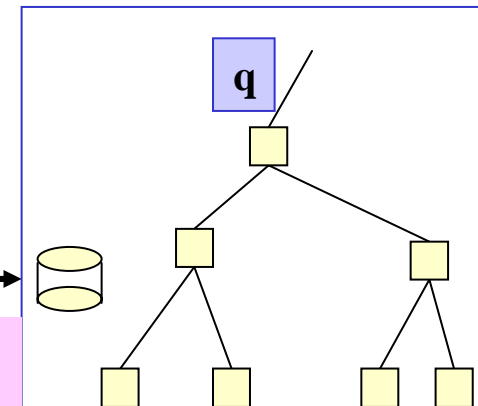
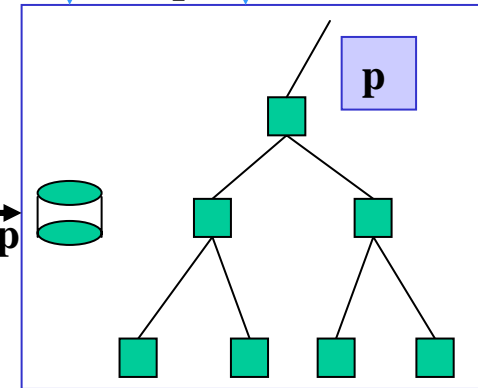
3.1 usa p

3.1 usa p

3.2

usa p

usa q



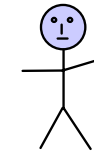
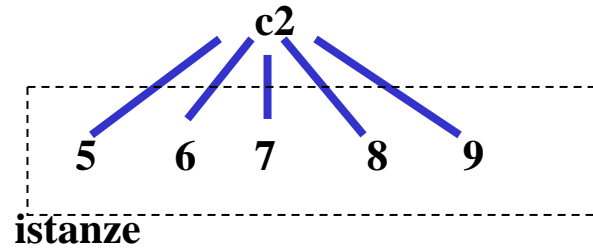
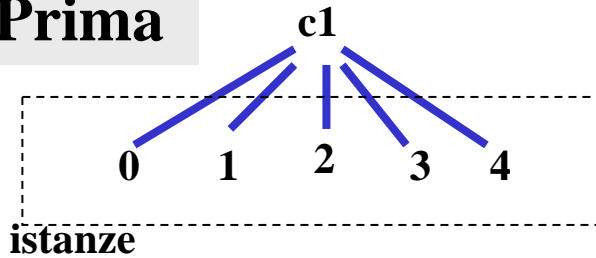
**La precedente definizione e visione dell'integrazione è fornita tenendo presente - dopo l'avvenuta integrazione - del tipo di modifica strutturale che subiscono le ontologie partecipanti al processo.**

**Un classe significativa di integrazione di questo tipo è quella denominata “mutua integrazione”**

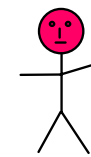
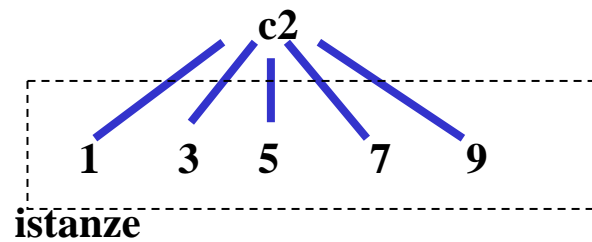
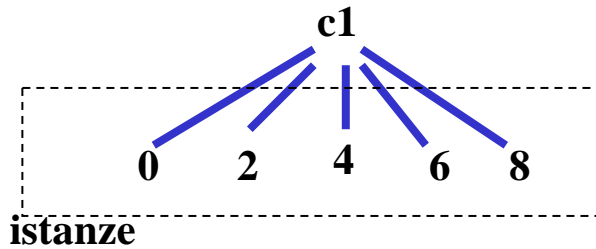


# Mutua integrazione: allineamento di ontologie

**Prima**

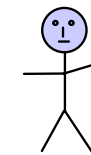
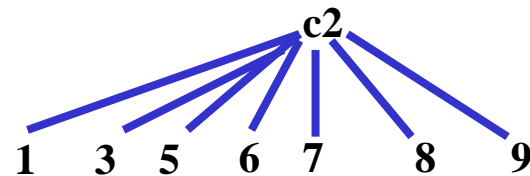
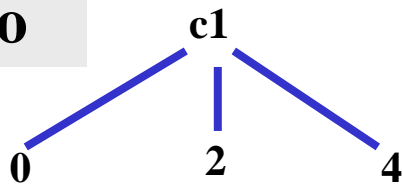


**Agente A**

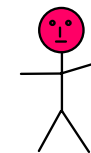
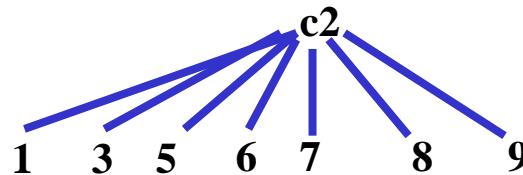
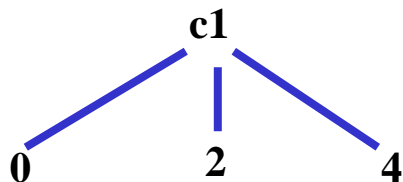


**Agente B**

**Dopo**

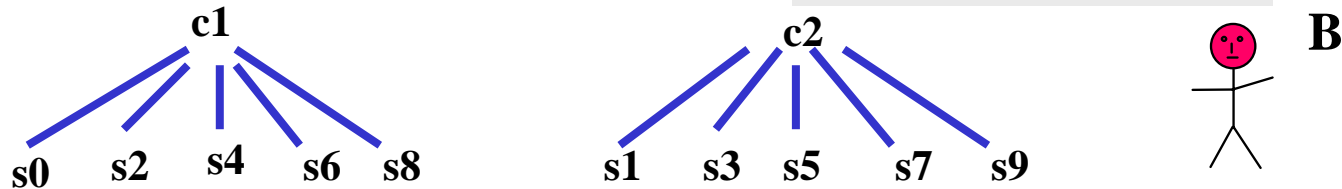
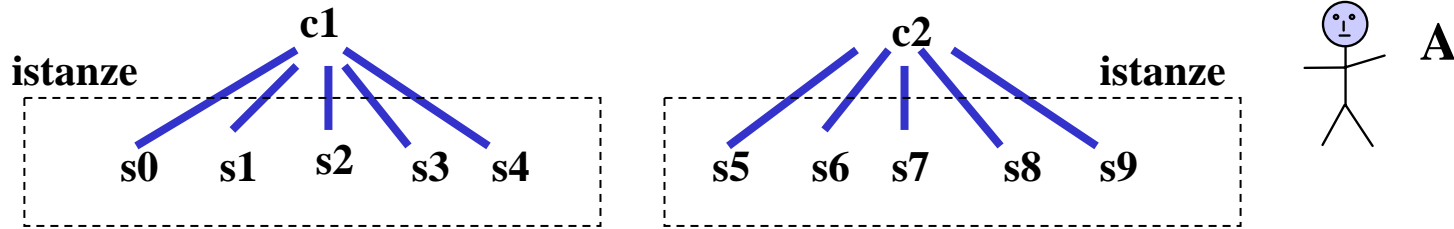


**Agente A**



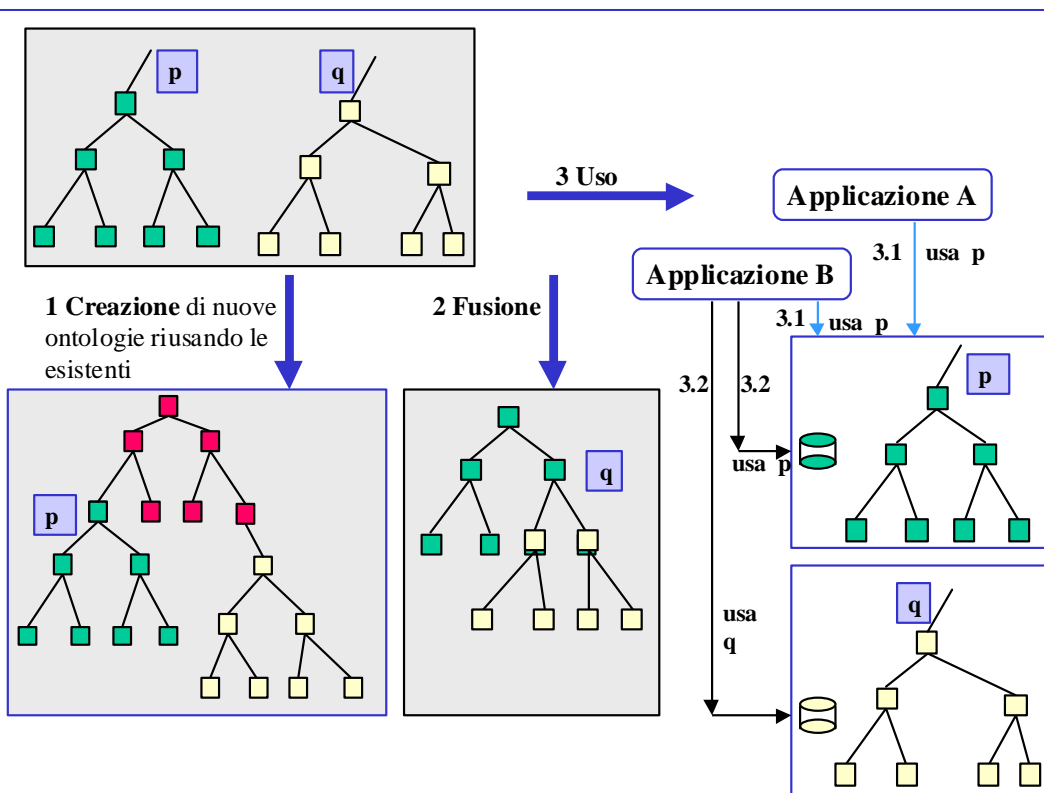
**Agente B**

# Mutua integrazione - processo di allineamento come gioco



## Le tipologie di integrazione considerate da Pinto necessitano di un assestamento teorico

le integrazione 1 e 2 (creazione e fusione) sono due tipi di integrazioni completamente diverse dalla 3 (uso): le prime dopo l'integrazione perdono la propria identità, mentre le seconde la mantengono

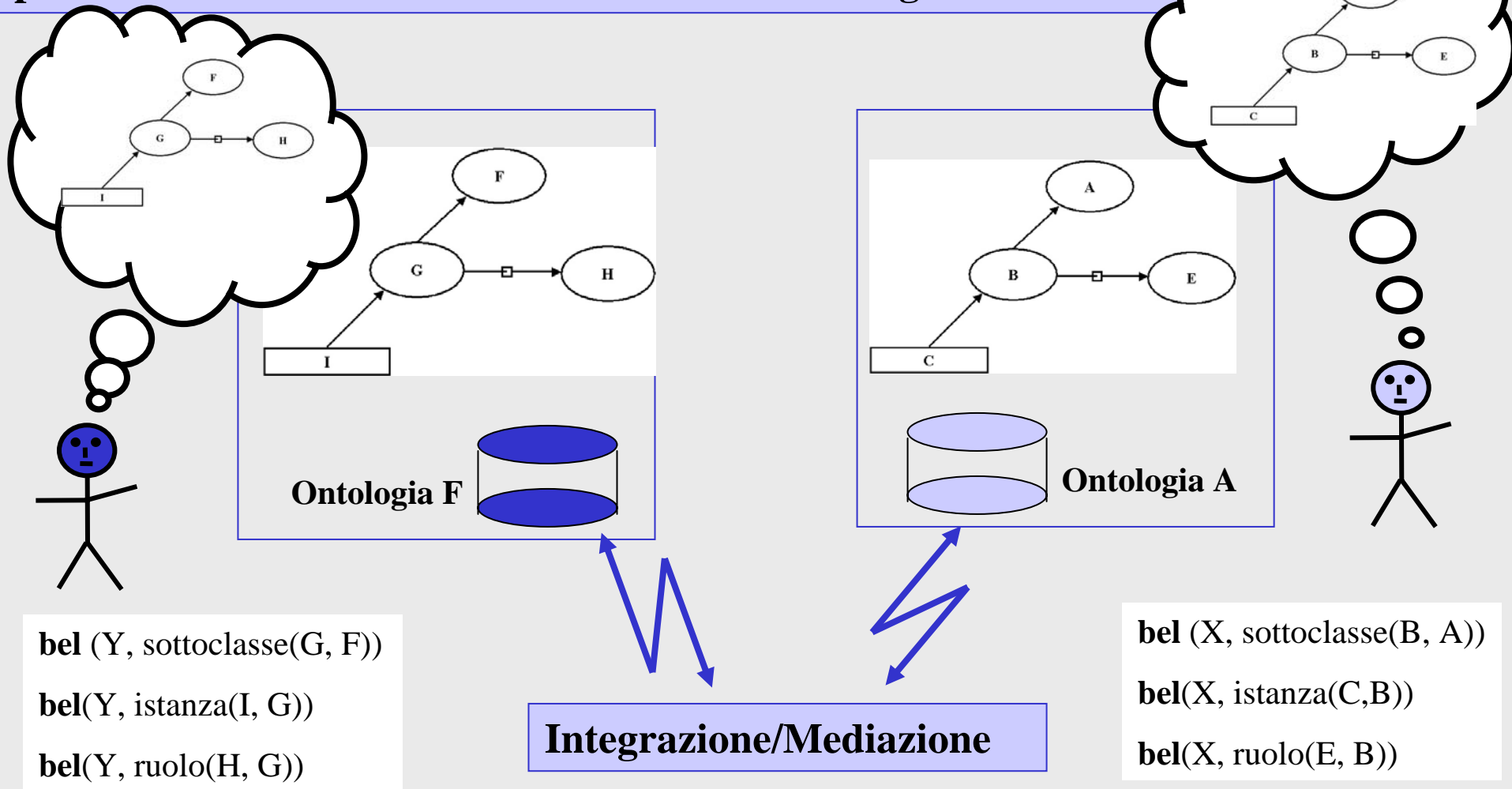


I processi 1 e 2 possono essere considerati più propriamente un'integrazione

Ponendo l'attenzione sulle singole componenti si può fornire la seguente definizione: c'è integrazione quando avviene un processo di aggiornamento o revisione in una o più ontologie che partecipano al processo.

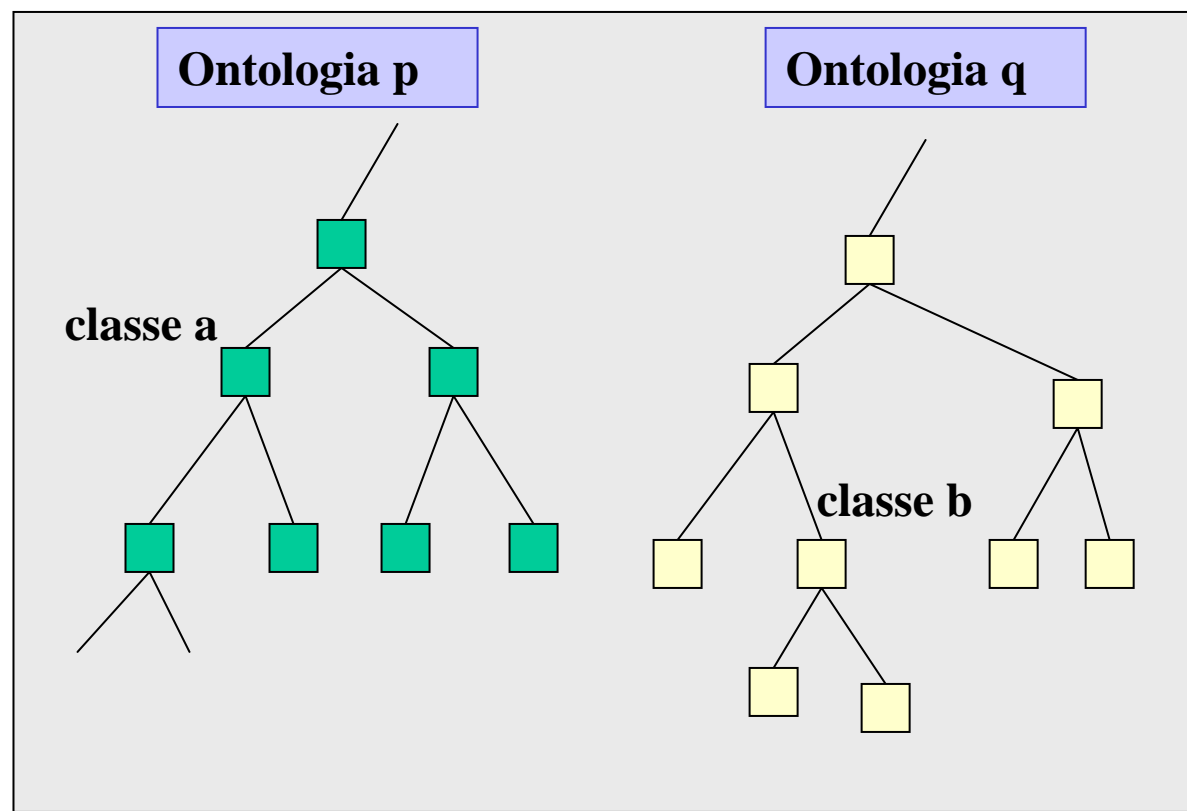
E' il punto 3.1 e 3.2?

**La definizione data in termini di aggiornamento e revisione è giustificata dal fatto che nel particolare scenario dell'integrazione/mediazione - qualunque sia l'operazione che accade o accadrà tra due o più ontologie – queste ultime possono essere considerate come credenze di un agente**



Tre criteri di similarità per classi di diverse ontologie:

- 1- rispetto ai sinonimi;
- 2- rispetto alle proprietà;
- 3- rispetto alle relazione con altre classi dell'ontologia



Una proposta di integrazione (Rodriguez A., Egenhofer Max)

$$S(a^p, b^q) = \omega_w \cdot S_w(a^p, b^q) + \omega_u \cdot S_u(a^p, b^q) + \omega_n \cdot S_n(a^p, b^q).$$

con  $\omega_w, \omega_u, \omega_n > 0$  e

$$\omega_w + \omega_u + \omega_n = 1$$

Una misura di  $S_w(a^p, b^q)$ ,  $S_u(a^p, b^q)$ ,  $S_n(a^p, b^q)$

**a** e **b** sono classi

**A** e **B** sono insiemi di descrizioni delle classi *a* e *b*: insiemi di sinonimi, insieme di proprietà e insieme di classi nell'*intorno* semantico;

$|A \cap B|$  è la cardinalità dell'insieme intersezione delle descrizioni delle due classi;

$|A \setminus B|$  è la cardinalità dell'insieme differenza delle descrizioni delle due classi **a** e **b**

$$S(a,b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a,b) |A \setminus B| + (1 - \alpha(a,b)) |B \setminus A|}$$

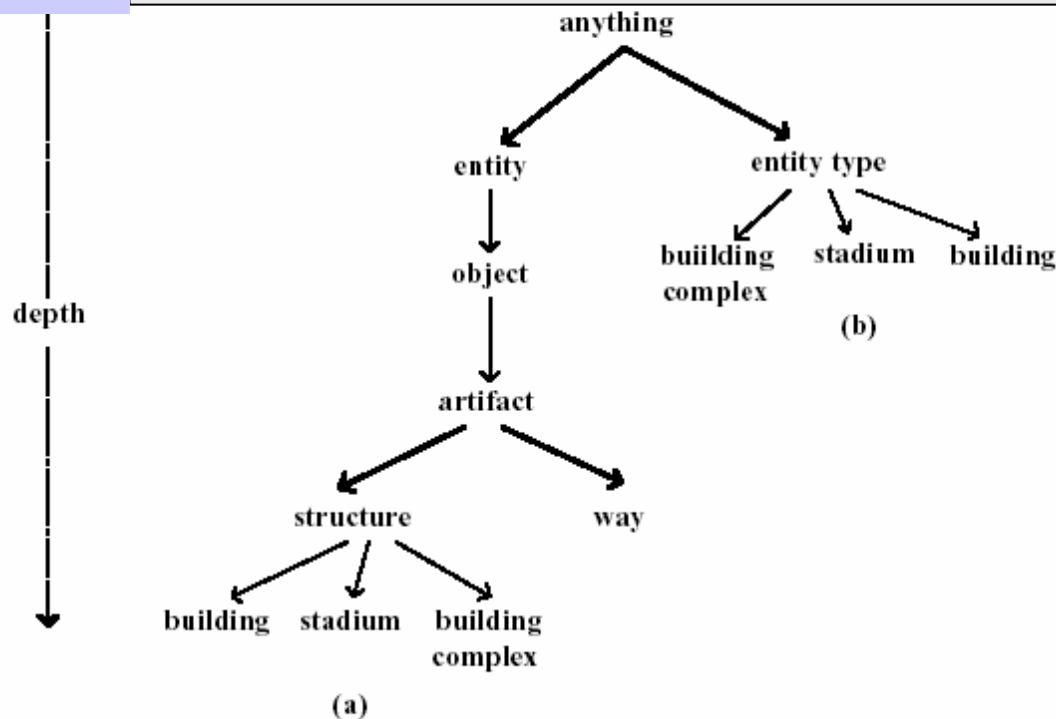
per  $0 \leq \alpha \leq 1$

$$\alpha(a^p, b^q) = \begin{cases} \frac{\text{depth}(a^p)}{\text{depth } a^p + \text{depth } (b^q)} & \text{depth } a^p \leq \text{depth } (b^q) \\ 1 - \frac{\text{depth}(a^p)}{\text{depth } a^p + \text{depth } (b^q)} & \text{depth } a^p > \text{depth } (b^q) \end{cases}$$

Come si misura  $\alpha$

$$0 \leq \alpha \leq 0.5$$

$$\alpha(a^p, b^q) = \begin{cases} \frac{\text{depth}(a^p)}{\text{depth } a^p + \text{depth } (b^q)} & \text{depth } a^p \leq \text{depth } (b^q) \\ 1 - \frac{\text{depth}(a^p)}{\text{depth } a^p + \text{depth } (b^q)} & \text{depth } a^p > \text{depth } (b^q) \end{cases}$$



**Figure 2:** Connecting independent ontologies: (a) partial WorNet ontology and (b) partial SDTS ontology. (*Anything\** denotes an imaginary root)

La misura di  $S_u(a^p, b^q)$  può essere raffinata dividendo le proprietà in parti, funzioni e attributi.

L'equazione è la stessa usata per l'integrazione:

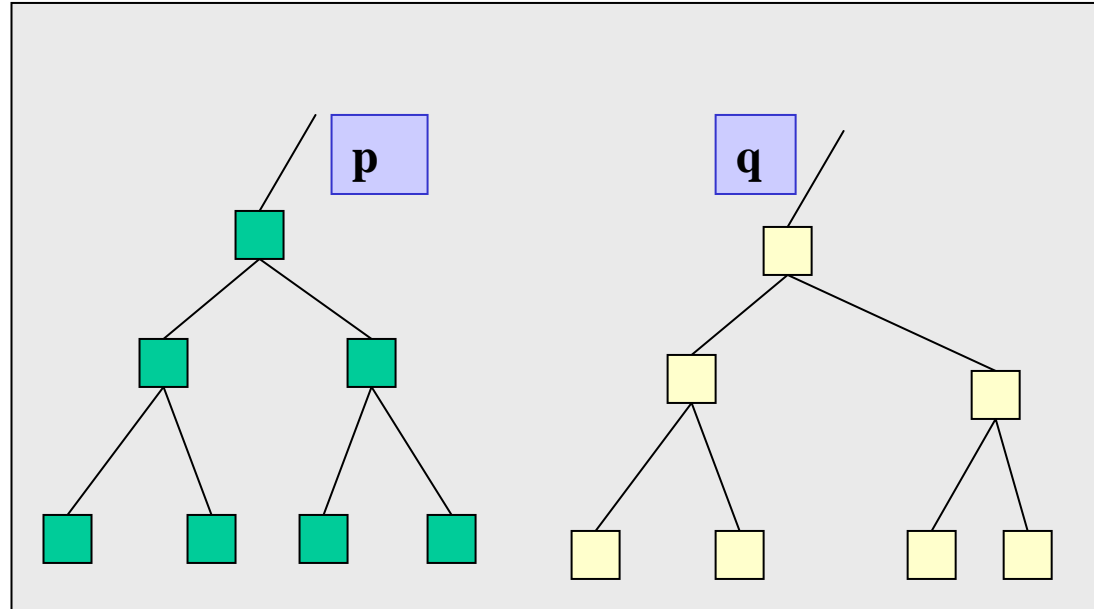
$$S(a^p, b^q) = \omega_p \cdot S_p(a^p, b^q) + \omega_f \cdot S_f(a^p, b^q) + \omega_a \cdot S_a(a^p, b^q).$$

Stadium (WS)	Stadium (WordNet)
<pre>entity_class {   name: {stadium,bowl,arena}   description: large often unroofed structure in                which athletic events are held   is_a: {construction*}   part_of: {}   whole_of: {athletic_field*}   parts: {{athletic_field,sports_field,playing_field},           {dressing_room},{foundation},           {midfield},{spectator_stands,stands},           {ticket_office,box_office,ticket_booth}}   functions: {{play,compete},{play,practise},               {recreate,play}}   attributes: {{architectural_property},                {covered/uncovered}, {name},                {lighted/unlighted},{owner_type},                {sports_type},{user_type}}}</pre>	<pre>entity_class {   name: {stadium,bowl,arena}   description: large often unroofed structure in which                athletic events are held   is_a: {construction*}   part_of: {}   whole_of: {athletic_field*, sports_arena*}   parts: {{athletic_field,sports_field,playing_field},           {foundation},{midfield},{plate},           {sports_arena,field_house},{stands},           {structural_elements},           {standing_room},{tiered_seats}}   functions: {}   attributes: {}}</pre>

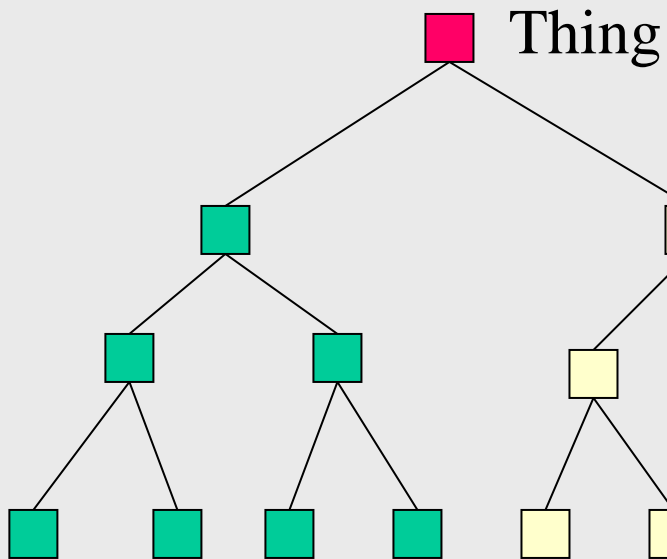
**Table 2:** Entity\_class definition of *stadium* in WS and WordNet. ( $x^*$  denotes a pointer to t

# Come possono essere impiegate le misure di similarità? - ad esempio

Se  $p$  e  $q$  sono ontologie indipendenti



Allora



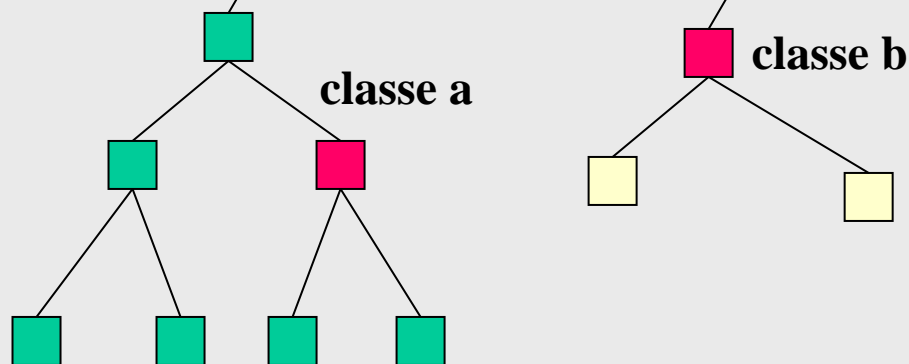
Se  $a$  e  $b$  hanno gli stessi attributi

Caso frequente:

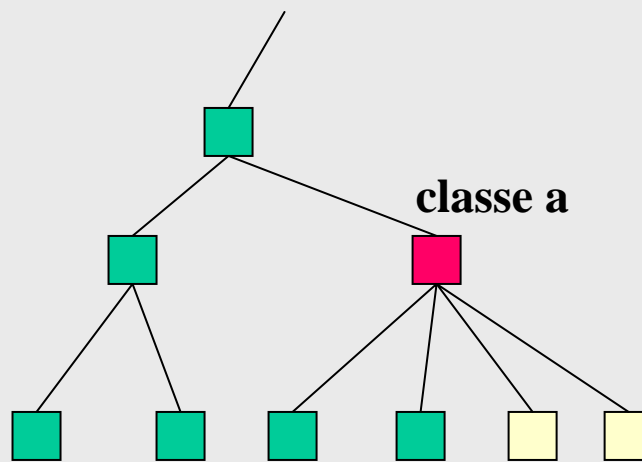
$b$  è una tabella di un data base

Ontologia  $p$

Ontologia  $q$



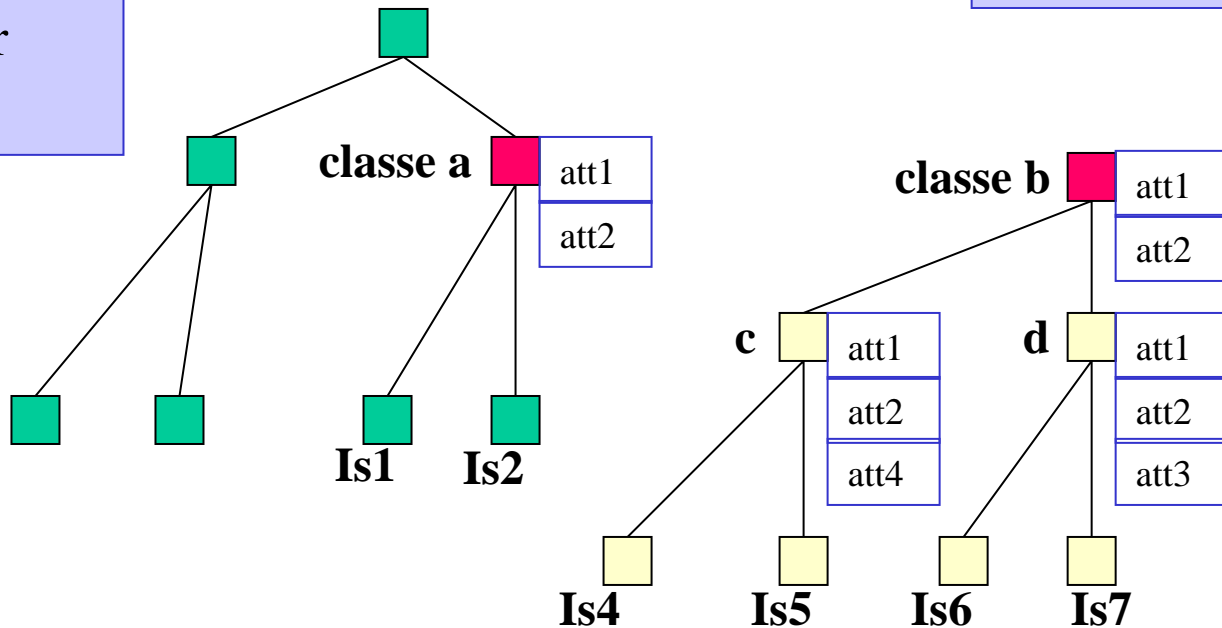
Se  $a$  e  $b$  hanno nomi diversi - se è permesso avere più nomi nella KB come secondo nome  $b$  - altrimenti risulta (ovvio) che  $a$



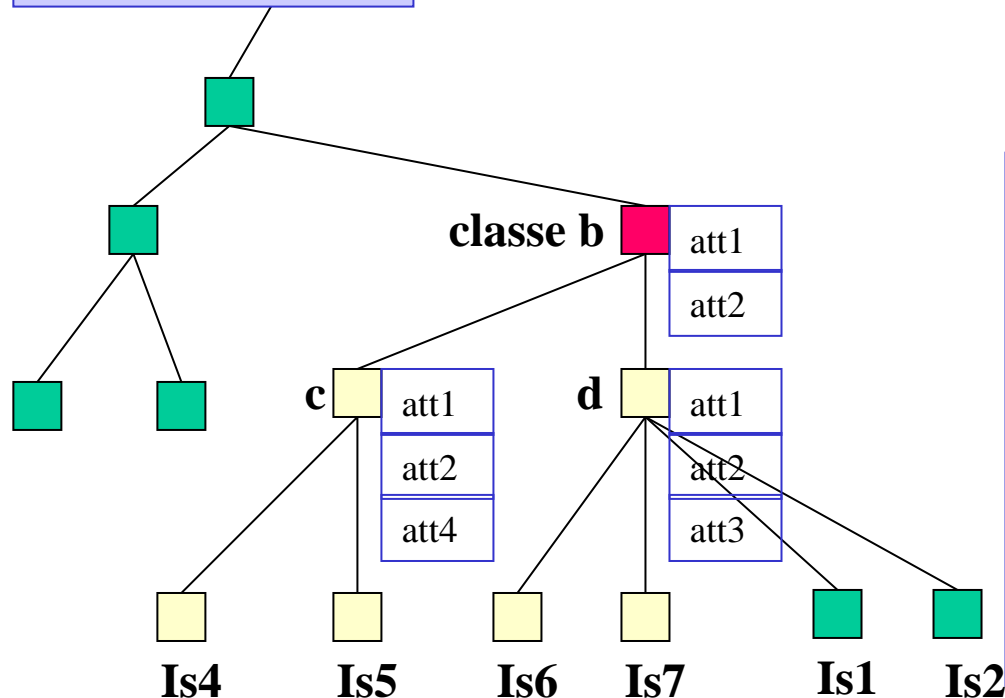
Se  $a$  e  $b$  hanno gli stessi attributi ma  $b$  ha un livello astratto intermedio in più (caso di raffinamento per specificità)

Ontologia p

Ontologia q



Un'ipotesi di nuova ontologia s



Le potenziali istanze  $Is1$  e  $Is2$  necessitano di un completamento di metadati e di valori ( $att3.valore\_att3$  o  $att4.valore\_att4$  a seconda) e possono essere o istanze di  $c$  o di  $d$ . Se i nomi delle classi sono differenti allora prenderà il nome classe  $b$  poiché è più specifica.

Nella costruzione dell'algoritmo precedente ci sono tre figure o tre punti di vista impliciti relativi a: Op, Oq e Om



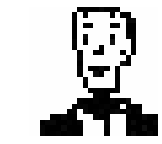
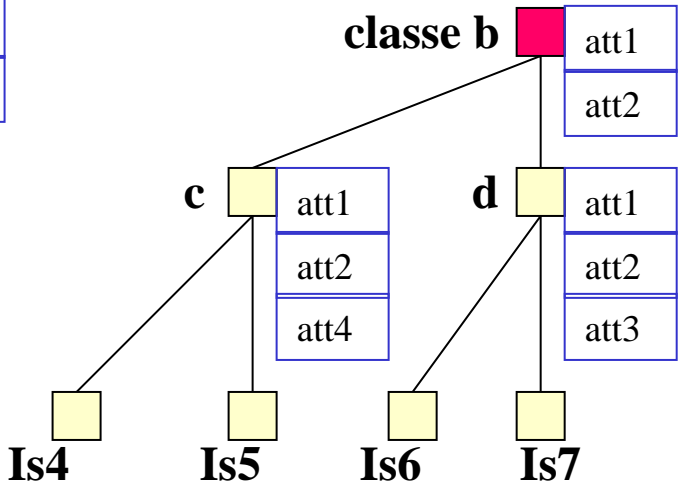
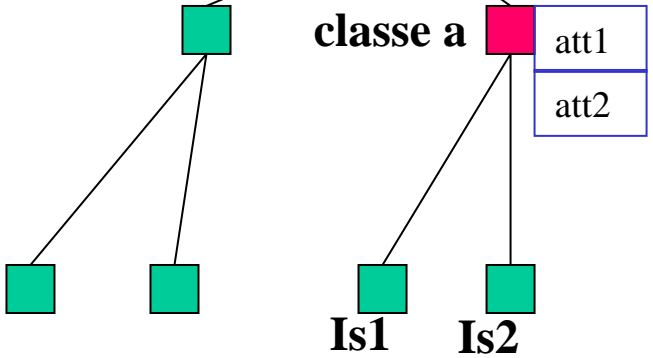
**Op: costruttore dell'ontologia p**



**Oq: costruttore dell'ontologia q**

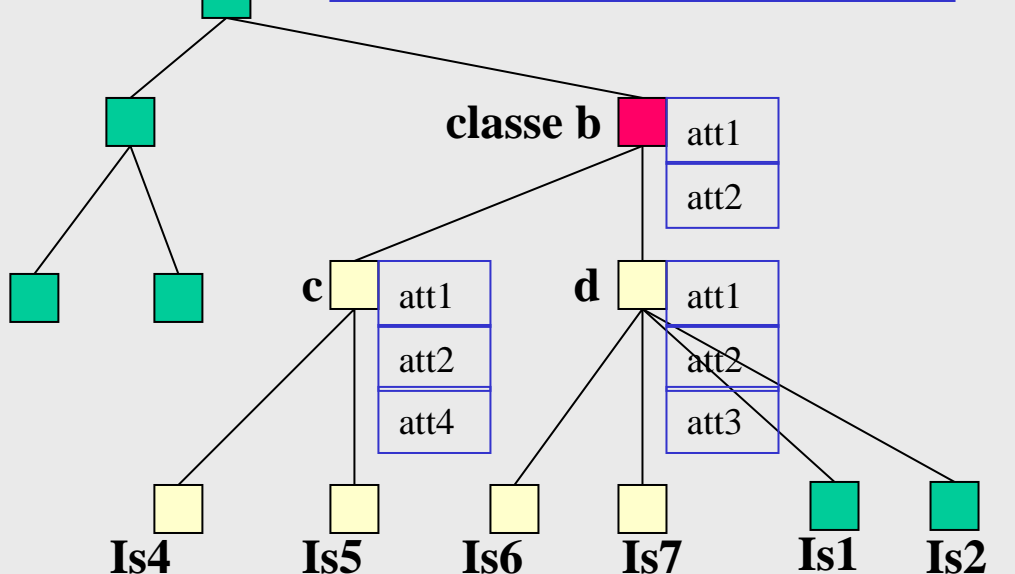
**Ontologia p**

**Ontologia q**



**Om: integratore**

**Ipotesi di nuova ontologia s**

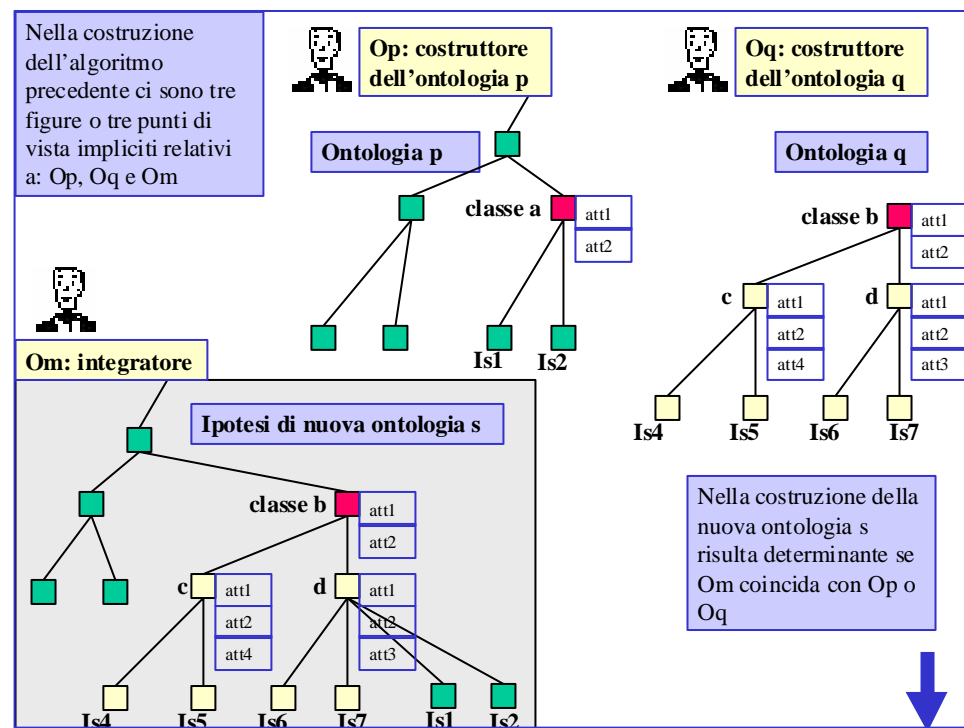


Nella costruzione della nuova ontologia s risulta determinante se Om coincida con Op o Oq



## Primo percorso:

la nuova ontologia s può essere costruita (interattivamente) da un mediatore umano che di volta decide quali entità delle due ontologie debbano essere integrate



Il mediatore può essere aiutato, nella sua attività di mediazione da un programma che individua punti delle ontologie di **possibili integrazioni** lasciando al mediatore Om la decisione

E' uno dei servizi presenti nel tool di integrazione **PROMPT**

# Ontologie con difetti: classi nascoste

Classe\_r1r2r3

Classe r1	
	att1
	att2
	att3
	att4

Classe r2	
	att1
	att2
	att5
	att6
	att7
	att8

Classe r3	
	att1
	att2
	att5
	att6
	att9
	att10

# La struttura corretta

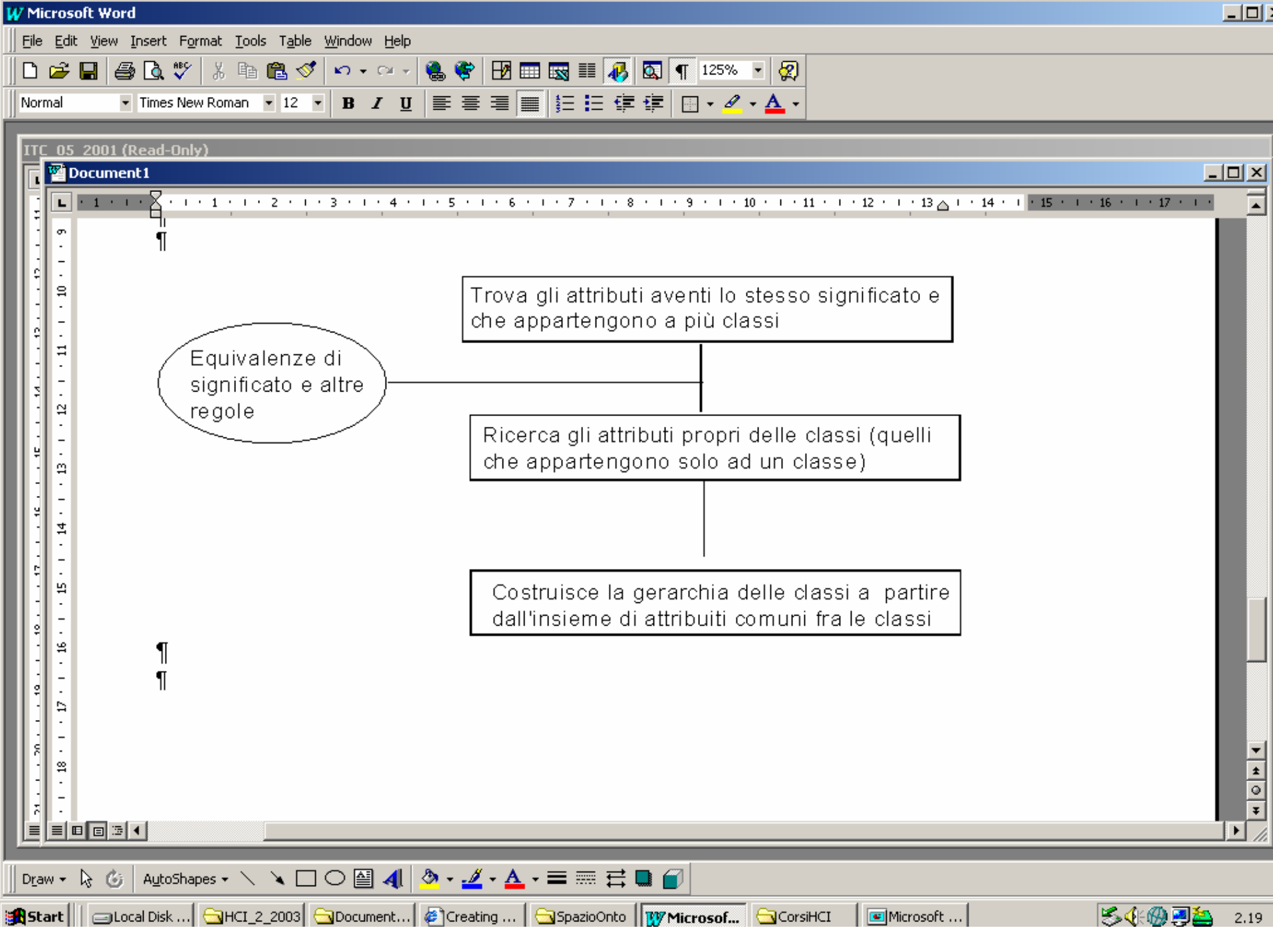
Sottoclasse r1r2r3	
	att1
	att2

Sottoclasse r1	
	att3
	att4

Sottoclasse r2r3	
	att5
	att6

Sottoclasse r2	
	att7
	att8

Sottoclasse r3	
	att9
	att10



Equivalenze di  
significato e altre  
regole

Trova gli attributi aventi lo stesso significato e  
che appartengono a più classi

Ricerca gli attributi propri delle classi (quelli  
che appartengono solo ad un classe)

Costruisce la gerarchia delle classi a partire  
dall'insieme di attributi comuni fra le classi

```
/* Rappresentazione relazioni ed attributi */
```

```
att_type(rel1,attr1,type).  
att_type(rel1,attr2,type).  
att_type(rel1,attr3,type).  
att_type(rel1,attr4,type).
```

```
att_type(rel2,attr1,type).  
att_type(rel2,attr2,type).  
att_type(rel2,attr5,type).  
att_type(rel2,attr6,type).  
att_type(rel2,attr7,type).  
att_type(rel2,attr8,type).
```

```
att_type(rel3,attr1,type).  
att_type(rel3,attr2,type).  
att_type(rel3,attr5,type).  
att_type(rel3,attr6,type).  
att_type(rel3,attr9,type).  
att_type(rel3,attr10,type).
```

```
/* Algoritmo */
```

```
means_attr(Attr,Attr).
```

```
find_eq_attr:-
```

```
    att_type(Rel1,Attr1,_),  
    att_type(Rel2,Attr2,_),  
    not(Rel1 = Rel2),  
    means_attr(Attr1,Attr2),  
    in_eq_means(Rel1,Rel2,Attr2),  
    fail.
```

```
find_eq_attr.
```

## Un possibile algoritmo per la trasformazione

```
in_eq_means(Rel1,Rel2,Attr):-  
    eq_means(Attr,List),  
    member(Rel2,List),!.
```

```
in_eq_means(Rel1,Rel2,Attr):-  
    eq_means(Attr,List),  
    append(List,[Rel2],List1),  
    asserta(eq_means(Attr,List1)),  
    retract(eq_means(Attr,List)),!.
```

```
in_eq_means(Rel1,Rel2,Attr):-  
    asserta(eq_means(Attr,[Rel1,Rel2])).
```

```
st_eq_means:-  
    eq_means(Attr,List),  
    nl, write(Attr),tab(1),write(List),fail.
```

```
st_eq_means:-nl.
```

```
st_num_class:-  
    num_class(N,Attr,List),  
    nl, write(N),tab(1),write(Attr),tab(1),write(List),fail.
```

```
st_num_class:-nl.
```

```
n_attr_class:-  
    eq_means(Attr,List),  
    num_list(List,0,N),
```

```
    asserta(num_class(N,List,Attr)),fail.
```

```
n_attr_class.
```

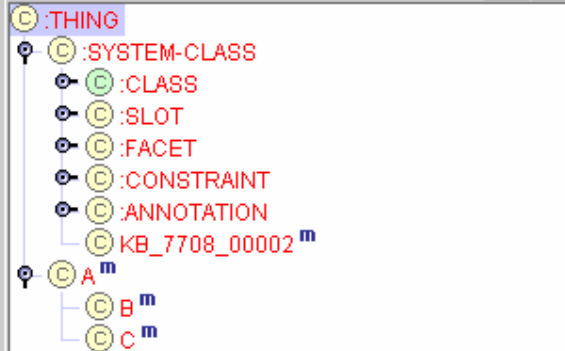
```
num_list([],N,N):-!.  
num_list([_|R],N,M):-  
    N1 is N + 1,  
    num_list(R,N1,M).
```

```
start:-  
    asserta(num_class(n1,list1,a1)),
```

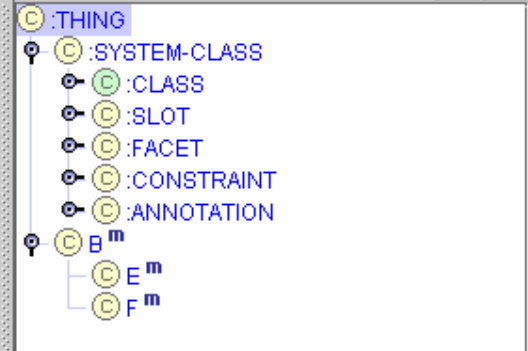
```
    retractall(num_class(N1,List1,A1)),  
    asserta(eq_means(r,type)),  
    retractall(eq_means(R2,A2)).
```



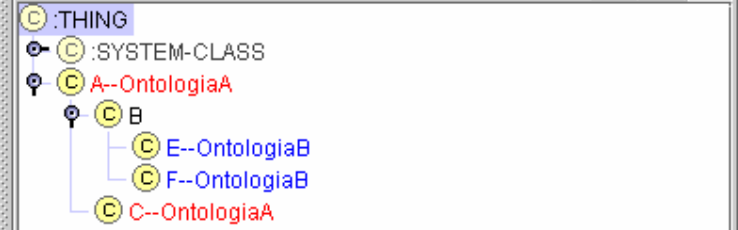
OntologiaA



OntologiaB



merged



merge classes

OntologiaA OntologiaB

Choose class Choose class

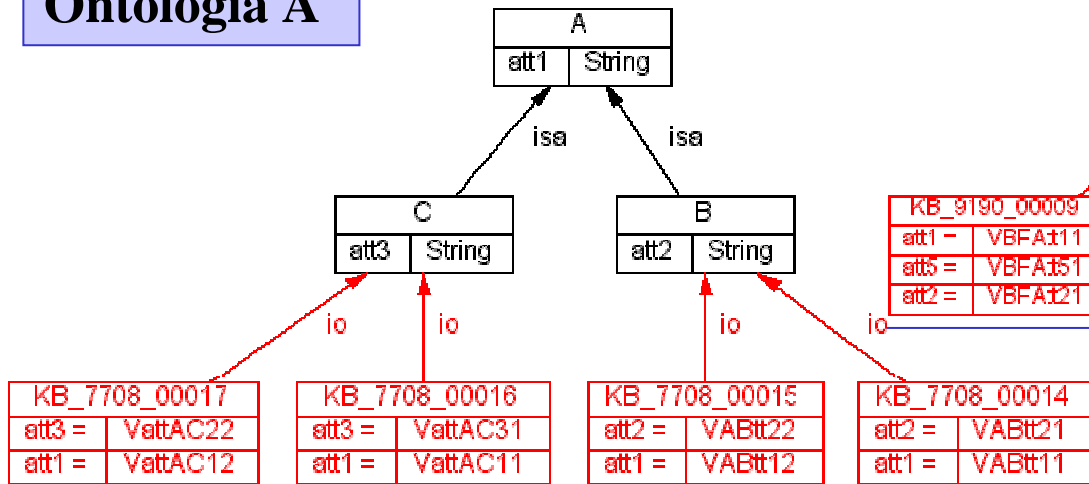
parents (up to the root) subclasses instances

Do It ToDo list

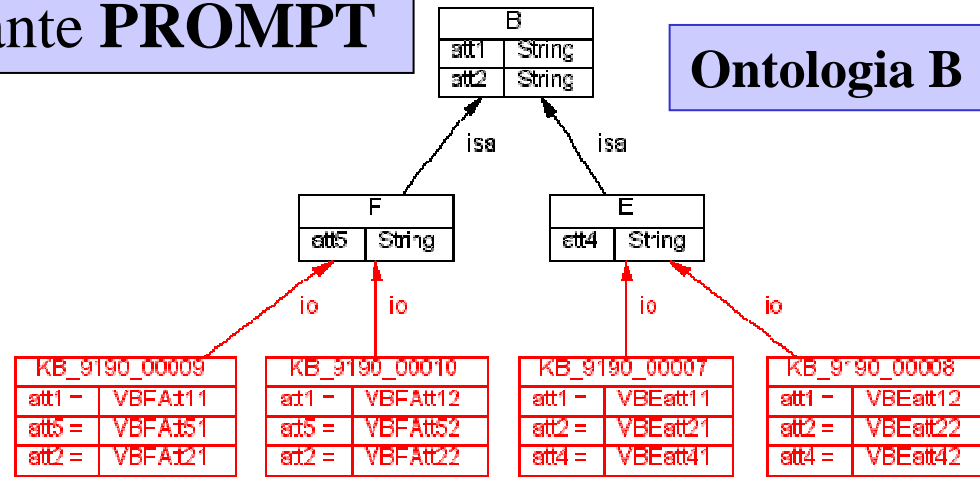


# Risultato dell'integrazione mediante PROMPT

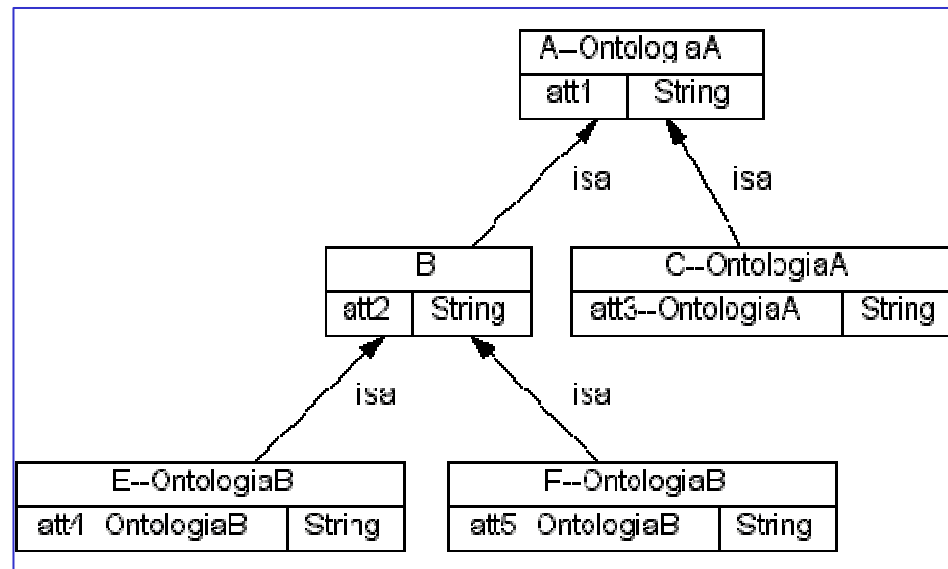
## Ontologia A



## Ontologia B



## Ontologia risultante

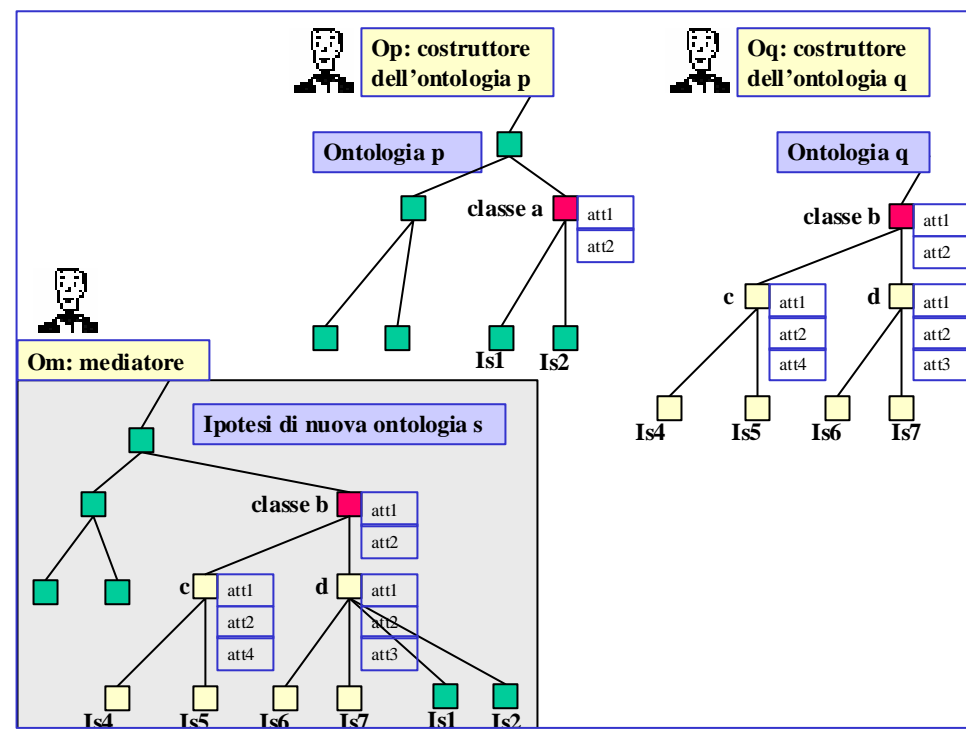


Il plugin **PROMPT** permette di gestire multiple ontologie in Protegé 2000. Usando PROMPT è possibile:

- **Unire due ontologie in una (merge mode);**
  - **Suggerimenti**
    - suggerisce il merge di classi con similarità linguistica
    - suggerire la copia di una frame
    - suggerisce la rimozione di superclassi di classi
  - **Conflitti**
    - indica l'esistenza di cicli
  - **Nuove operazioni**
- **Estrarre una parte di un'ontologia;**
- **Trasportare frame fra progetti;**
- **Comparare due versione della stessa ontologia e creare una nuova versione**

## Secondo percorso:

alla nuova ontologia  $s$  si può pervenire con una procedura programma che integri o cerchi di integrare le due ontologie (scelta del punto di vista e problem solving relativo)



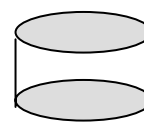
## Le strategie possibili per la costruzione del mapping:

a- si assume che uno dei costruttori di  $p$  o  $q$  abbia lo stesso punto di vista dell'integratore Om;

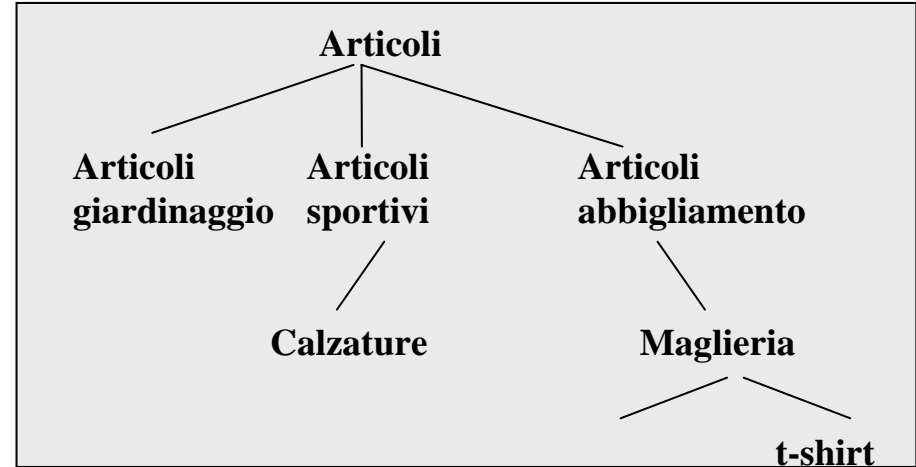
b- si prende una terza ontologia di riferimento che fa da arbitro;

c - si decide di volta in volta il punto di vista usando regole *contestuali* (ad esempio: fra le classe a e b si sceglie b perché propone un raffinamento maggior di a)

# Mediazioni fra ontologie



Ontologia di riferimento

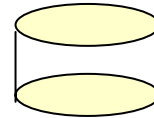


Applicazione

select nome of t-shirt  
where prezzo < 100

Sistema di mediazione

help 2  
sub\_classe(t-shirt, maglieria) 2

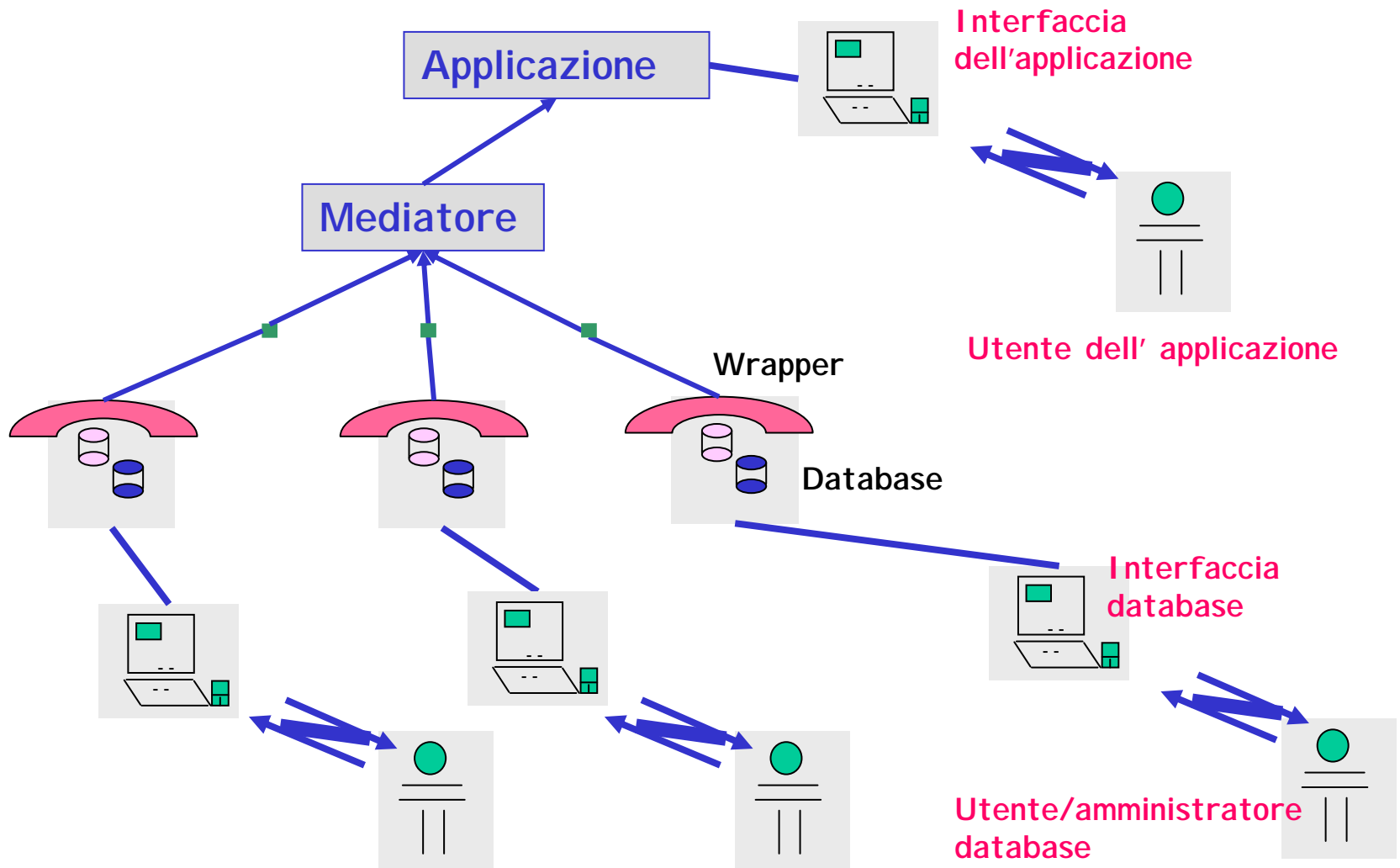


La tabella maglieria

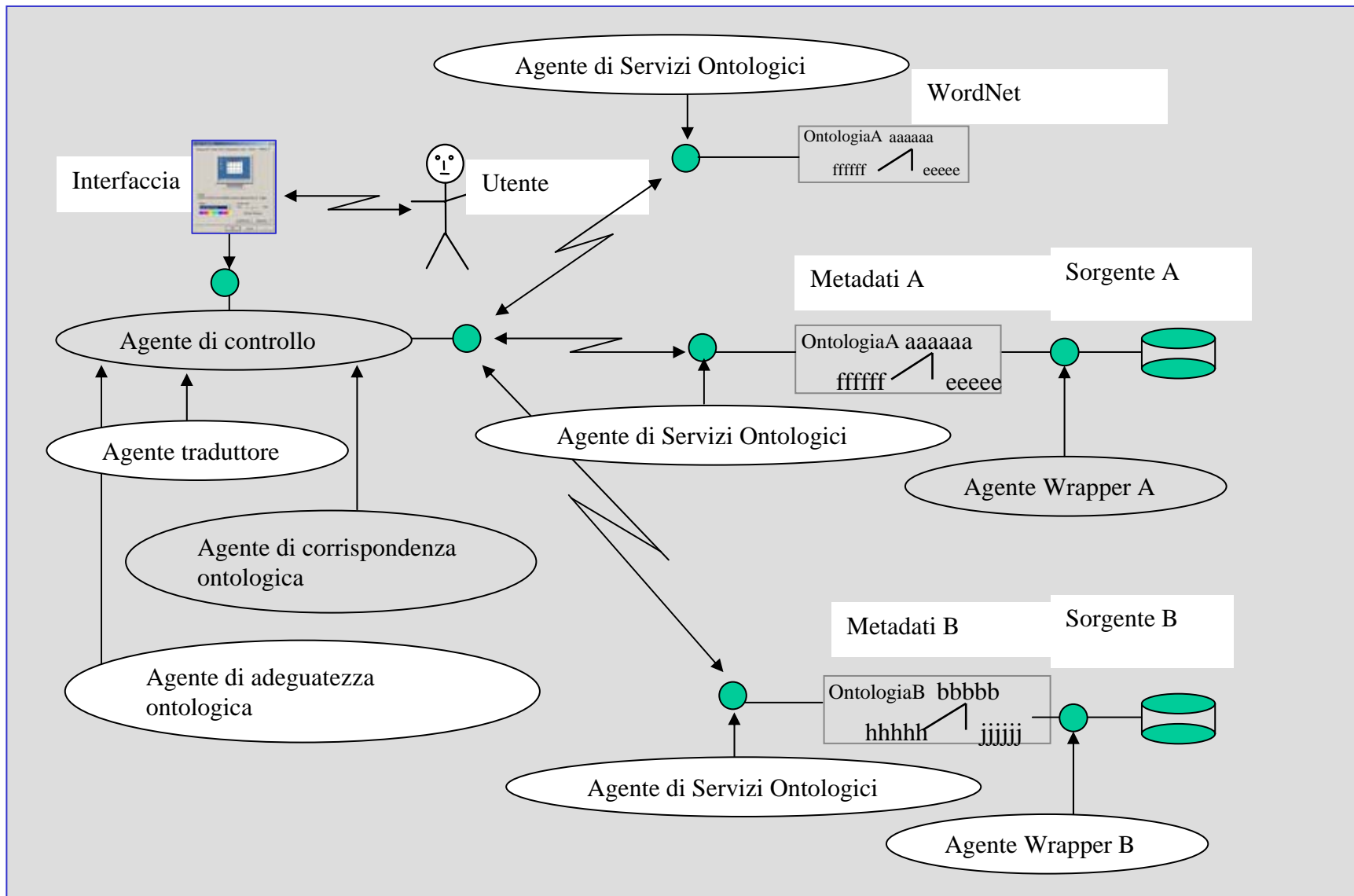
Maglieria				
Cod	Nome	Prezzo	Tipo	Xxx
--	--	--	--	--
--	--	--	--	--
--	--	--	--	--

1 silenzio  
1 select nome of t-shirt where prezzo < 100  
2 select nome of maglieria where ...

# Architettura per l'integrazione/mediazione di risorse

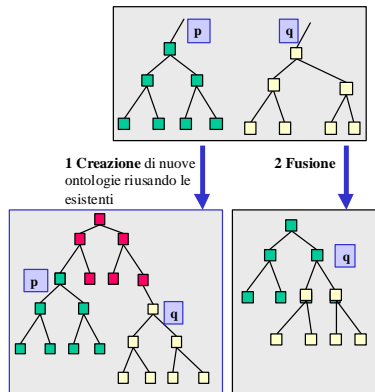


# Un sistema di mediazione basato su agenti

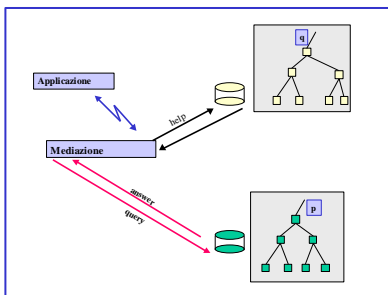


# Commenti e domande finale

## Integrazione



## Mediazione



**Integrazione e mediazione (spesso sono usate come sinonimo) rappresentano due filosofie per che portano a diversi conseguenze “sociali” entrambe interessanti.**

**Sembrerebbe che:**

**l’integrazione tenda ad una totale standardizzazione - una sorta di globalizzazione delle basi di conoscenze;**

**la mediazione tenda ad una separazione delle conoscenze secondo punti di vista, culture differenti**

**Sembrerebbe che:**

**l’integrazione riduce il numero delle ontologie proponendone poche di grandi dimensioni;**

**la mediazione porta ad aumento delle ontologie (private) richiedendo più capacità dialogica fra i sistemi**

**Sembrerebbe che: integrazione e mediazione siano due processi contrapposti la vittoria di uno sull’altro può portare ad una diversa topologia di distribuzione in rete**

# Due processi contrapposti che portano (sembrerebbe che) ad una quasi prevedibile conseguenza: la standardizzazione delle ontologie (delle conoscenze?)

Immaginiamo uno scenario dove c'è una abbondanza di mediazione

