

Cosa sono le ontologie

- **Filosofia/Computer ScienceAI**: area dell'intelligenza artificiale che studia i metodi per rappresentare correttamente l'universo che ci circonda.

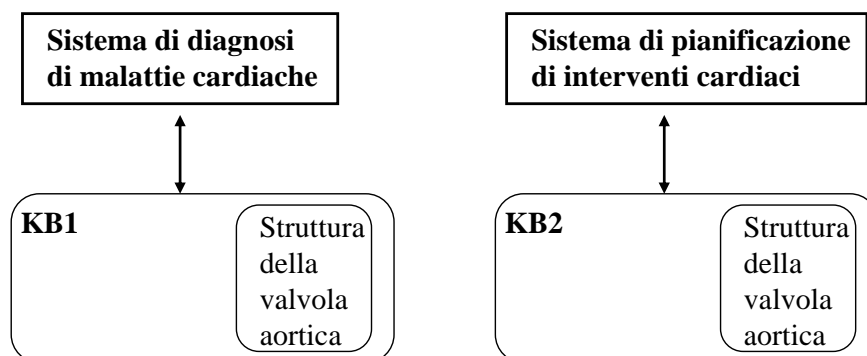
Perchè servono in CS?

- **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!

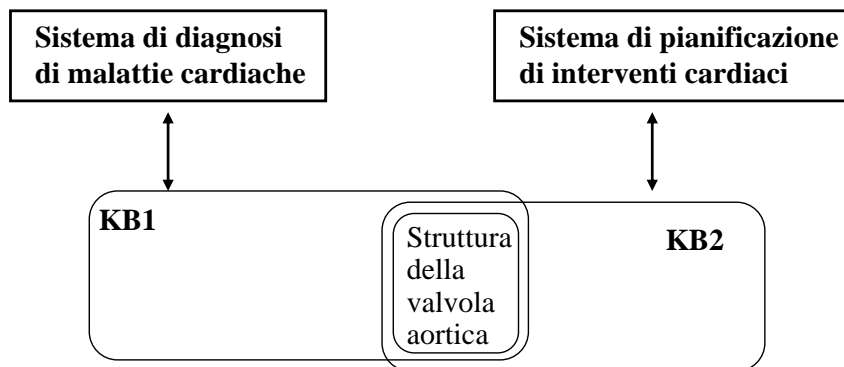
1

Conoscenza non condivisa



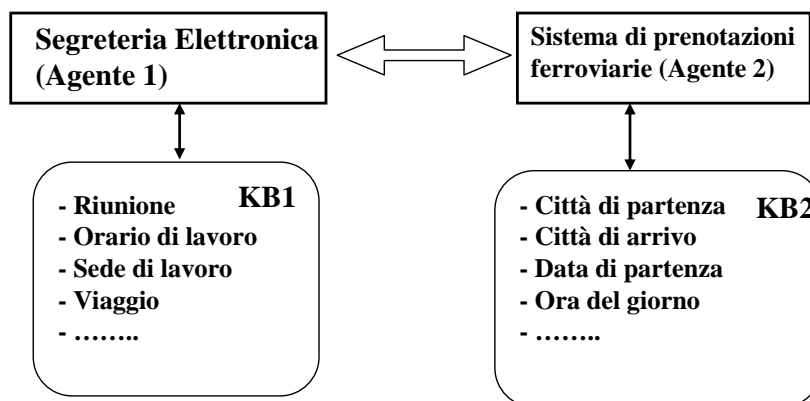
2

Conoscenza condivisa



3

Comunicazione



ma:

cittaDiPartenza := sedeDiLavoro;
cittaDiArrivo := luogoDelMeeting;

E se cittaDiPartenza si chiama CittaPart o DepartureCity?

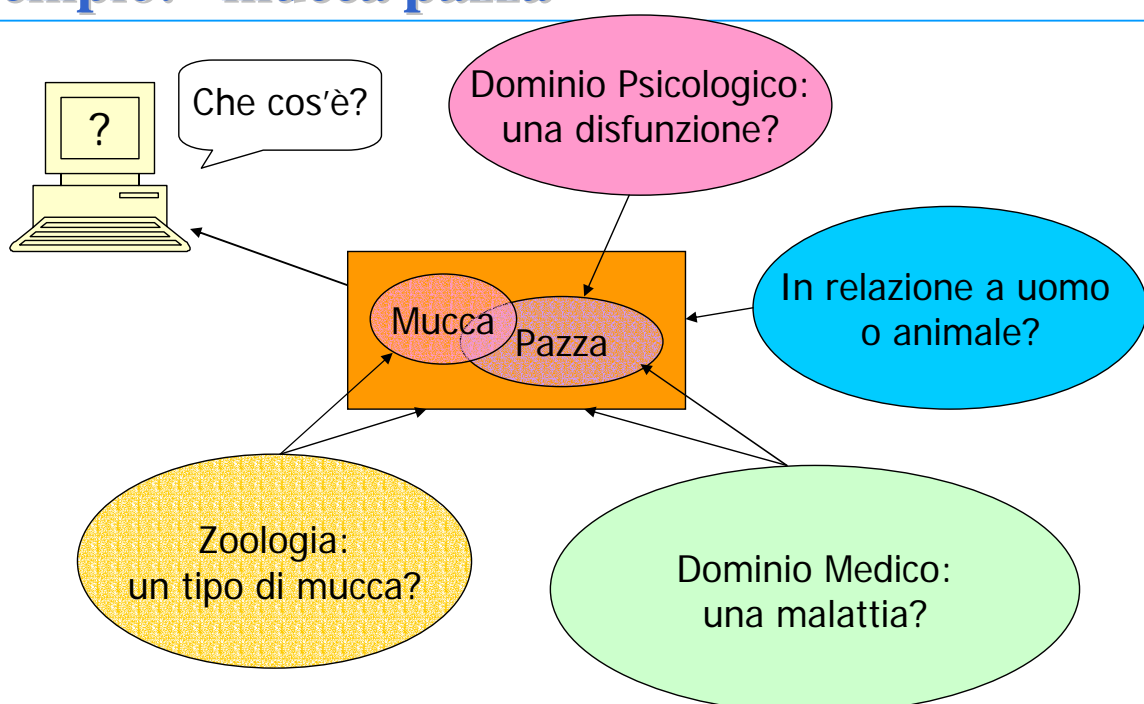
4

Ontologie e Web Semantico

- Possibilità di accesso e acquisizione della conoscenza tramite www
- Costo trascurabile per acquisire basi di conoscenza
- Necessità di organizzare, integrare e interrogare basi di conoscenza
- Necessità di sorgenti di conoscenza facilmente accessibili da macchine e processi automatici
- Necessità di una conoscenza riutilizzabile e condivisibile (in contesti e forme differenti)

5

Esempio: “mucca pazza”



6

Problemi di fondo (II)

1. Occorre eliminare la confusione terminologica e concettuale ed individuare le *entità* cui un pacchetto di conoscenza si riferisce.
2. Organizzare e rendere esplicito il *significato referenziale* permette di *comprendere* l'informazione.
3. Condividere questa *comprensione* facilita il recupero e il riutilizzo della conoscenza tra agenti e in contesti diversi.

↓

ONTOLOGIE

7

Ontologia

Definizione formale di un dominio di conoscenza

↓

Isolare una parte del mondo e i suoi concetti fondamentali

↓

Enumerare e definire (in modo più o meno formale) i concetti e le relazioni che tra essi sussistono:
→ classi, proprietà, assiomi, individui

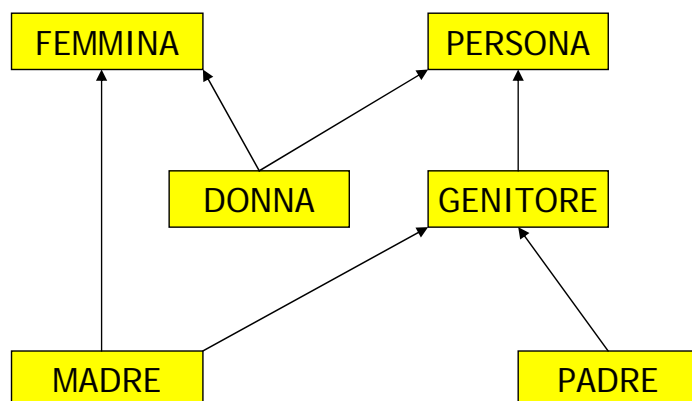
↓

Una descrizione strutturata gerarchicamente dei concetti importanti e delle loro proprietà che trovi il consenso di diversi attori interessati a condividerla e utilizzarla.

8

Elementi di un'ontologia: classi

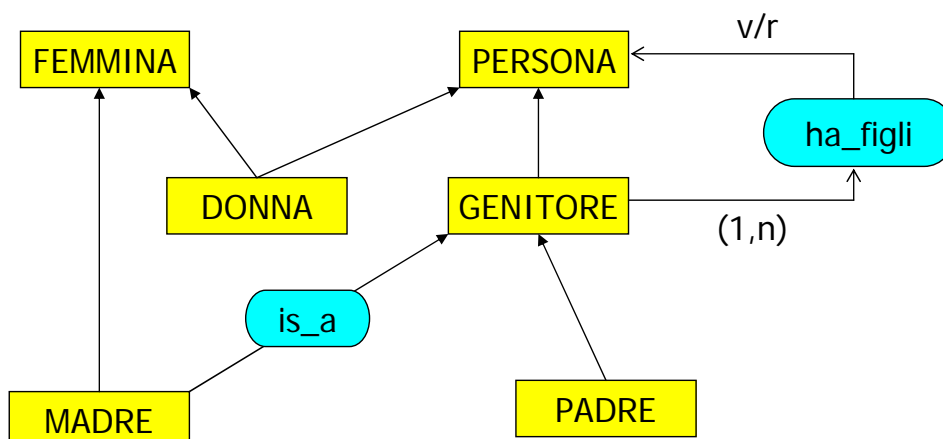
Generalmente organizzati in tassonomie. La "classe" viene utilizzata in senso lato per descrivere i concetti (le classi di individui) che si vuole rappresentare (es. PERSONA, GENITORE, MADRE...)



9

Elementi di un'ontologia: proprietà e attributi (I)

Rappresentano le relazioni esistenti tra le classi del dominio
(es. "le MADRI sono GENITORI", "i GENITORI hanno figli che sono PERSONE"...)



10

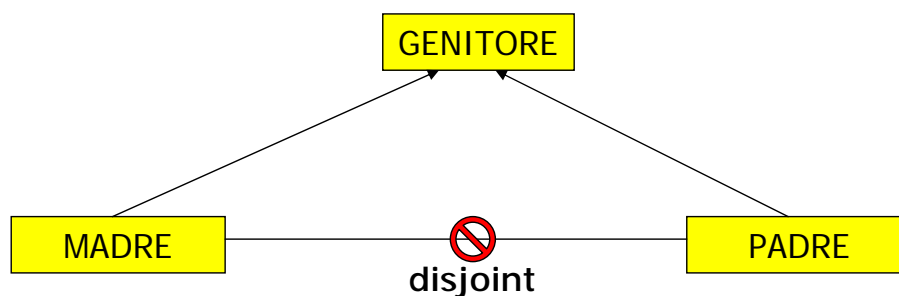
Elementi di un'ontologia: Proprietà e attributi (II)

- Il legame tra MADRE e GENITORE (**is_a**) indica che “le MADRI sono GENITORI” e definisce una *gerarchia* tra concetti, provvedendo una base per l'*eredità* di proprietà: un concetto specifico eredita le proprietà del concetto più generale che lo sussume.
- E' possibile rappresentare anche proprietà più complesse della relazione **is_a**.
- Ad esempio, è possibile definire la proprietà **ha_figli**, che connette le due classi GENITORE e PERSONA, specificando degli *attributi* che ne vincolano l'applicazione: v/r denota una restrizione sulle classi che possono soddisfare la proprietà, mentre (1,n) rappresenta una restrizione di cardinalità.
- L'esempio può essere letto come “un GENITORE è una PERSONA che **ha almeno 1 figlio** e **tutti** i figli che ha sono PERSONE”.

11

Elementi di un'ontologia: assiomi

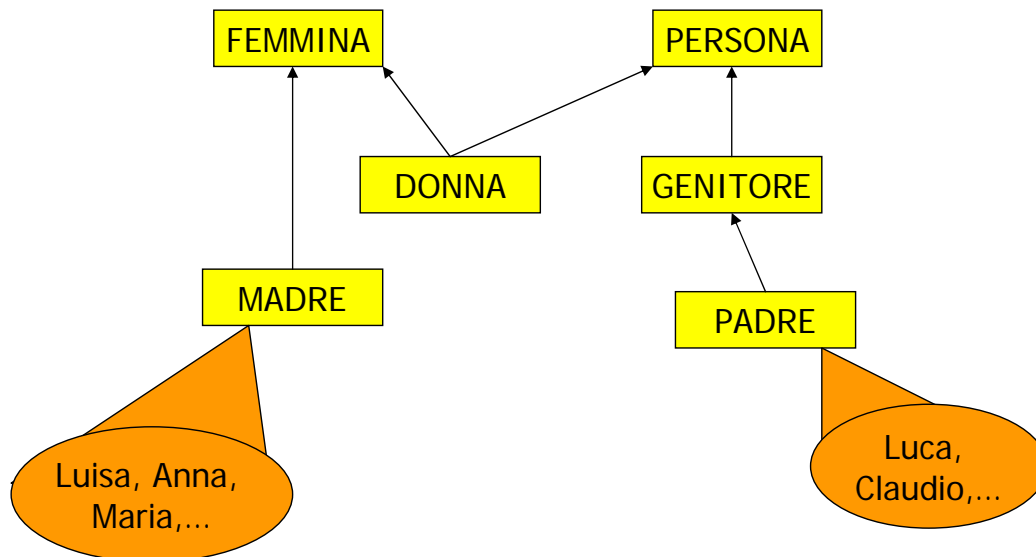
- Vengono utilizzati per modellare in maniera esplicita espressioni in ogni caso vere.
- Possono essere utilizzati per diversi scopi: definire il significato dei vari componenti dell'ontologia, definire relazioni complesse, verificare la correttezza dell'informazione specificata o dedurre nuova informazione.
(es. *disjoint* (MADRE, PADRE) esprime il fatto che un elemento della classe PADRE non può mai essere anche un elemento della classe MADRE)



12

Elementi di un'ontologia: individui

Sono i singoli oggetti contenuti in una classe, a vari livelli di generalità, a seconda dello scopo dell'ontologia



13

Linguaggio

Una volta selezionati i componenti dell'ontologia occorre definirli in maniera esplicita. E' possibile distinguere ontologie diverse sulla base del linguaggio utilizzato per definirne i componenti:

- **Informali:** espresse tramite linguaggio naturale
- **Semi-formali:** espresse tramite linguaggio artificiale definito formalmente
- **Formali:** espresse in un linguaggio dotato di semantica formale, teoremi e proprietà quali validità e completezza.

14

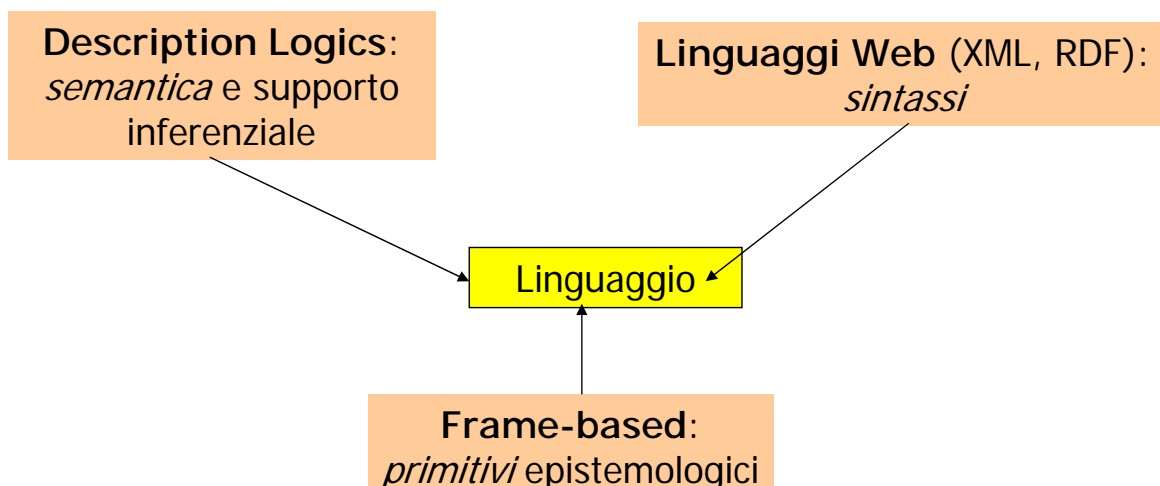
Scelta del linguaggio

- Necessità di un linguaggio comune per lo sviluppo delle ontologie (rende più semplice la condivisione e il riuso di ontologie tra progetti differenti)
- Povertà dei linguaggi di rappresentazione largamente utilizzati, quali XML e RDF.
- Necessità di linguaggi espressivi e rigorosi ma di facile utilizzo

15

Linguaggio di Rappresentazione

Fonde tre caratteristiche ereditate da linguaggi suoi progenitori che definiscono gli aspetti fondamentali di un linguaggio di rappresentazione:



16

Rappresentazione frame-based

- Primitive di rappresentazione basate su classi (frame) dotate di proprietà dette attributi (slot): è possibile definire una classe come una collezione di superclassi o attributi. Es:

Madre italiana con molti figli	
<i>superclasse</i>	madre
ha_figli	≥ 3 figli
vive_in	Italia

- Questo tipo di rappresentazione manca però di una semantica rigorosamente definita: problemi computazionali nell'utilizzo della rappresentazione frame-based.

Una madre italiana con molti figli è una madre con almeno 3 figli che vive in Italia.

17

Description logics

- Permette di definire le classi in termini di descrizioni che specificano le proprietà degli oggetti che appartengono ad una determinata classe.
- Fornisce gli operatori per definire le proprietà: congiunzione, disgiunzione, negazione, quantificazione...
- Semantica formale
- Supporto inferenza

} Frammento decidibile di logica del prim'ordine

Es: Madre italiana con molti figli \equiv madre \wedge ≥ 3 ha_figli \wedge vive_in.Italia

18

Linguaggi Web

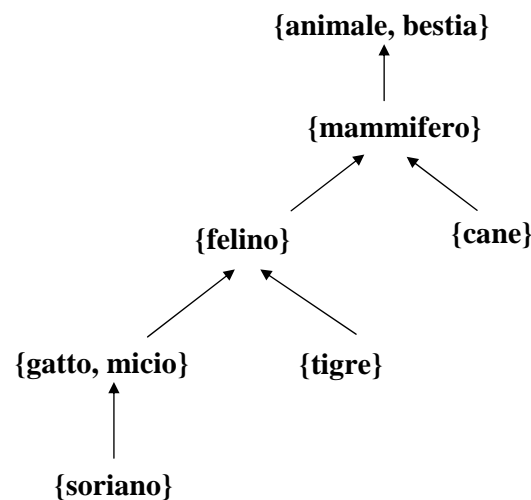
- Un linguaggio per la rappresentazione di ontologie necessita, oltre a primitivi epistemologici e semantica, una sintassi concreta e un formato che ne permetta lo scambio e la lettura da parte di una macchina: sintassi XML e RDF.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-
syntax-ns#
  xmlns:a="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:about="madre#ha_figli" />
  <a:Class rdf:about="madre#madre" />
- <a:Class
rdf:about="madre#madre+italiana+con+molti+figli">
  <a:subClassOf rdf:resource="madre#madre" />
  </a:Class>
  <a:Class rdf:about="madre#persona" />
  <rdf:Property rdf:about="madre#vive_in" />
</rdf:RDF>
```

19

Esempio di Ontologia: WordNet

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



20

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
- **So, the mattress in the road to AI is lack of knowledge, and the anti-mattress is knowledge. But how much does a program need to know to begin with? The annoying, inelegant, but apparently true answer is: a non-trivial fraction of *consensus reality* - the millions of things that we all know and that we assume everyone else knows” (Guha & Lenat 90, p.4)**

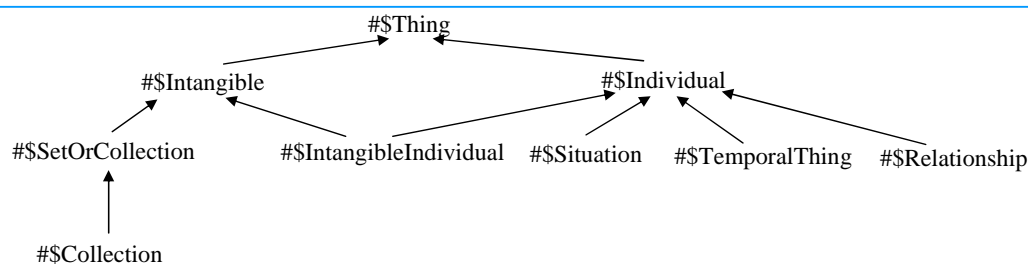
21

2 componenti

Constraint Language (variante della Logica dei predicati)
CycL (linguaggio basato su frame/units)

22

Il top level di Cyc



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`.

23

#\$Intangible:

la collezione di cose che non sono fisiche - non sono fatte di, o codificate nella, materia. Ogni `#$Collection` è `#$Intangible` (anche se le sue istanze sono tangibili) e tali sono anche alcuni

~~`#$Individual`. Attenzione: non si confonda 'tangibilità' con 'percettibilità' - gli esseri umani possono percepire la luce anche se essa è intangibile.~~

#\$Individual:

la collezione di tutte le cose che non sono insiemi o collezioni. Così `#$Individual` include, tra le altre cose, oggetti fisici, sottoastrazioni temporali di oggetti fisici, numeri, relazioni e gruppi. Un elemento di `#$Individual` può avere parti o una struttura (includere parti che sono discontinue); ma NESSUNA istanza di `#$Individual` può avere elementi o sottoinsiemi.

#\$IntangibleIndividual:

la collezione degli individui intangibili. I suoi elementi non hanno massa, volume, colore, ecc. Ad esempio, ore, idee, algoritmi, interi, distanze, e così via. D'altra parte, in quanto sottoinsieme di `#$Individual`, questa collezione ESCLUDE insiemi e collezioni, che sono elementi di `#$Intangible`, ma non di `#$IntangibleIndividual`

#\$TemporalThing:

la collezione delle cose che hanno una particolare estensione temporale, cose delle quali uno potrebbe ragionevolmente chiedere 'Quando?'. Essa include molte cose; come le azioni, gli oggetti tangibili, gli accordi, e porzioni astratte di tempo. Alcune cose NON sono istanze di `#$TemporalThing` perchè sono astratte, atemporali, come un insieme matematico, un intero, ecc.

24

OWL

(Web Ontology Language)

- il legame con il World Wide Web
- OWL è un prodotto del W3C (organizzazione internazionale per lo sviluppo di WWW); un linguaggio di markup semantico per pubblicare e condividere ontologie.
- OWL è derivato dal precedente DAML+OIL
- OWL è basato sui linguaggi di markup (XML) ed è ancorato all'RDF (Resource Description Framework), una notazione uniforme e non ambigua (sintatticamente definita come estensione di HTML tramite XML) per esprimere le risorse presenti sul Web.

25

OWL: Web Ontology Language

- Un aggancio tra strumenti ontologici ed il WWW, nell'ottica di creare un Semantic Web

VANTAGGI:

- E' una layer supplementare di strumenti quasi universalmente accettati e in uso
- Sono disponibili vari livelli di potere espressivo

SVANTAGGI:

- Manca un reasoner incorporato
- La sintassi (derivata da HTML e RDF) è pesante e poco leggibile

26

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 (e, forse, sull'opportunità culturale) di realizzare un'ontologia standard (difficoltà tecniche e salvaguardia di diversità culturali)**
- ma**
- **Si tratta comunque di una strada obbligata**