

# Ingegneria della Conoscenza: Introduzione

Nicola Fanizzi

# Ingegneria della Conoscenza

CdL in Informatica • Dipartimento di Informatica

Università degli studi di Bari Aldo Moro

**indice**

## 1 Introduzione

## 1.1 Definizioni Preliminari

## 1.2 Scienza e Ingegneria

### 1.3 La Piramide del Sapere

### 1.3.1 Dato

### 1.3.2 Informazione

### 1.3.3 Conoscenza

## 2 Rappresentazione e Ragionamento

### 2.1 Rappresentazione della Conoscenza

#### 2.1.1 Conoscenza

#### 2.1.2 Soluzioni

#### 2.1.3 Rappresentazioni del Problema

#### 2.1.4 Ragionamento

### 2.2 Sistemi RR

#### 2.2.1 Linguaggio

#### 2.2.2 Esempio | DB come RSS

#### 2.2.3 Ontologie e Concettualizzazioni

## 3 Dimensioni della Complessità

### 3.1 Modularità

### 3.2 Schema di Rappresentazione

### 3.3 Orizzonte

### 3.4 Incertezza

#### 3.4.1 Osservazione Incerta

#### 3.4.2 Effetto Incerto

### 3.5 Preferenza

### 3.6 Numero di Agenti

### 3.7 Apprendimento

### 3.8 Limiti sulle Risorse Computazionali

### 3.9 Interazione delle Dimensioni

## 4 Applicazioni

### 4.1 Infobot

### 4.2 Caratteristiche Comuni

# 1 Introduzione

---

## 1.1 Definizioni Preliminari

---

*Contesto*

### Intelligenza Artificiale (AI)

disciplina tesa a studiare e comprendere i principi che rendono possibile un comportamento intelligente in sistemi artificiali

- Ipotesi:

*ragionamento  $\approx$  computazione*

- collegata alla (ipo)tesi di *Church-Turing*
  - c'è un *livello di astrazione* nel quale si può interpretare il ragionamento come manipolazione di simboli
  - tale livello può spiegare le azioni di un sistema in termini dei suoi input
- Intelligenza artificiale (o computazionale): si propone di fornire metodi per la progettazione di *artefatti SW intelligenti*, utili a scopi precisi

L'*intelligenza* dei sistemi coinvolti non è necessariamente quella umana:

- si consideri la classe delle *organizzazioni*: individui in sé anche non particolarmente intelligenti ma la comunità può esibire un comportamento intelligente
  - prototipo: *ant colony*, ma anche gli sciami (*swarm*)
    - ricerca del cibo, adattamento ai cambiamenti
  - le aziende lavorano su prodotti per i quali la somma delle competenze richieste è molto maggiore di quella di ogni singolo addetto

## 1.2 Scienza e Ingegneria

---

“ È possibile creare macchine in grado di volare? ”

analogia a

“ È possibile creare macchine in grado di pensare? ”

Analogia, che spiega la tensione tra la visione *scientifica* e quella *tecnologica*:

- AI come *scienza* che mira a comprendere i principi del ragionamento
  - secondo il metodo scientifico, si dovrebbero creare e verificare teorie (confutabili)
    - teorie sulla soluzione algoritmica di problemi d'interesse supportate empiricamente attraverso le implementazioni

“ *Knowledge engineering (KE) refers to all technical, scientific and social aspects involved in building, maintaining and using knowledge-based systems.* ”

- AI come disciplina *ingegneristica*, tesa a costruire tecnologie/sistemi che risolvano specifici problemi
  - creare e testare *sistemi SW intelligenti basati su conoscenza* (KBS)
    - la cui qualità può essere valutata attraverso gli standard dell'informatica

La verifica sperimentale delle teorie è essenziale

- *Rasoio di Occam*: preferire sempre teorie e implementazioni più semplici

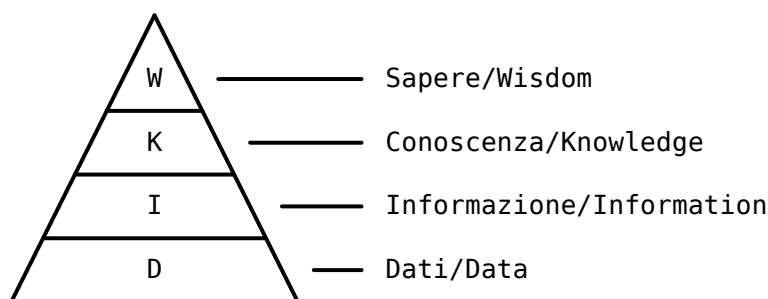
## 1.3 La Piramide del Sapere

---

### Piramide del Sapere<sup>1</sup>

(detta anche *gerarchia della conoscenza* o *DIKW*) classe di modelli atti a rappresentare relazioni strutturali e/o funzionali sottese tra *dati*, *informazioni*, *conoscenza* e *sapere*

“ Typically information is defined in terms of data, knowledge in terms of information, and wisdom in terms of knowledge ”



- modello rappresentato anche come *concatenazione*, come un *framework*, con una serie di *grafi* e come un *continuum*

<sup>1</sup> su [wikipedia](#)

### 1.3.1 Dato

*simboli* o *segni*, che rappresentano stimoli o segnali e che rimangono inutili fino a quando non sono messi in una qualche forma:

- *universali*, prodotti dall'osservazione  
o
- *soggettivi*, le osservazioni stesse

#### *fatti*

osservazioni discrete, oggettive, non organizzate o elaborate che difettano di un contesto interpretativo; sono *veri*, *oggettivi*, o *almeno verificabili*

#### *segnali*

nel dominio soggettivo, sono *stimoli sensoriali*, o letture di *segnali* (attraverso sensi/sensori)

## *simboli*

insiemi di segni che rappresentano le *percezioni* di proprietà di oggetti, eventi, dell'ambiente: simboli *registrati* (catturati, immagazzinati) necessari alla comunicazione

### 1.3.2 Informazione

L'**informazione** può essere definita come dati che sono dotati di *significato* / *scopo*:

- a differenza dei dati, si ottiene *per descrizione* e si differenzia per la sua *utilità*
- viene *inferita* dai dati
  - rispondendo a specifiche domande (e.g. le 5 W)
  - rendendoli quindi utili a prendere *decisioni* (o intraprendere *azioni*)

Distinzioni:

#### **strutturale vs. funzionale**

informazione come:

- dati organizzati per conferire loro rilevanza per determinati scopi o contesti risultando quindi significativi e utili
- formulazione alternativa come “dati che ci cambiano”

#### **simbolica vs. soggettiva**

informazione come:

- *universale*, come simboli o segni;
- *soggettiva*, significato al quale vengono collegati i simboli;
- entrambe

---

#### **Esempi**

- informazione che rappresenta uno stato di *consapevolezza*, fenomeno che rappresenta sia un processo che un prodotto;
- informazione correlata al *significato* e all'*intenzione*
  - contenuto dei DB, del web, ecc.
  - o significato di enunciati nelle intenzioni del parlante / dello scrivente compreso / travisato dall'ascoltatore / lettore

### 1.3.3 Conoscenza

**Conoscenza** come informazione elaborata, organizzata, o altrimenti applicata, messa in atto

- *commistione* di esperienza sistematizzata, valori, informazione contestuale, comprensione profonda e intuizione ben fondata
- fornisce un *ambiente* e una *struttura* per la valutazione e l'acquisizione di nuove esperienze e informazioni
  - nei singoli *agenti*, origina e viene applicata a livello *mentale*
  - nelle *organizzazioni* spesso incorporata non solo attraverso documenti anche in senso esteso, e loro sistemi di memorizzazione, ma anche nelle procedure (routine) organizzative, processi, pratiche e norme

### Conoscenza *elaborata*

definizioni alternative:

- sintesi di più sorgenti di informazioni nel tempo
- organizzazione + elaborazione portano alla comprensione, all'esperienza e all'apprendimento
- commistione di informazione di contesto, valori, esperienza e regole
- informazione connessa attraverso relazioni

### Conoscenza *procedurale*

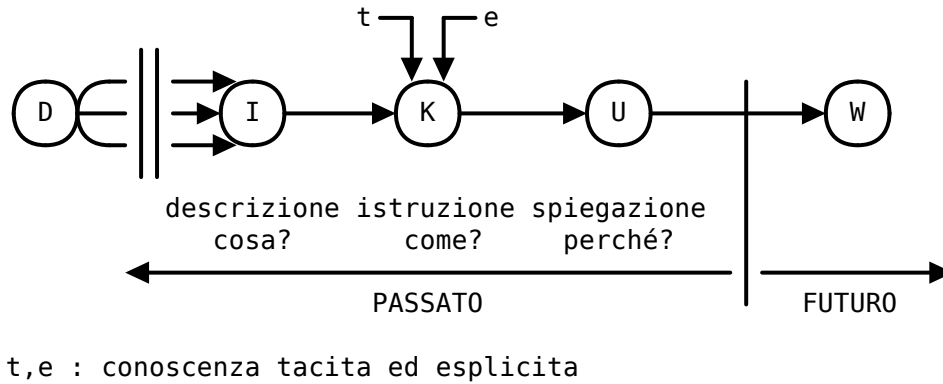
definita come:

- “know-how” e anche “know-who” e “know-when”, raggiunti tramite un'esperienza pratica
  - azione e non la descrizione di un'azione
- applicazione di dati e informazioni
- esperienza, qualità, expertise, capacità

### Conoscenza *proposizionale*

descritta come strutturazione delle credenze e internalizzazione:

- in termini proposizionali, come credenze giustificabili (e loro segni)
  - può diventare a sua volta informazione
- soggettivamente:
  - pensiero caratterizzato dalla credenza che essa sia vera
  - empirica ma anche non-empirica,
    - logica, matematica, filosofica, ...



## 2 Rappresentazione e Ragionamento

Per esperienza: le prestazioni su compiti che sembrano coinvolgere *forme di intelligenza* richiedono un certo quantitativo di *conoscenza*:

- Informalmente, informazioni su un dato dominio o soggetto o anche come svolgere date azioni
- Scopo: formalizzare e raffinare la nozione di conoscenza derivante dal senso comune ai fini dello sviluppo di un *framework teorico-pratico* per la sua rappresentazione e uso

### Compiti

- Gli agenti umani hanno bisogno di molta conoscenza per svolgere compiti anche molto semplici
- Le macchine sono in genere più indicate per compiti che non richiedano molta conoscenza (es. op. matematiche)
  - diventano sempre più brave in compiti *knowledge-intensive*, come il riconoscimento delle facce, le diagnosi mediche, la piena comprensione del linguaggio o le argomentazioni di tipo legale
- S'intende investigare la costruzione di sistemi SW dotati di conoscenza riguardo il mondo
  - capaci di acquisirla e usarla per risolvere i problemi sottoposti

### Problematiche:

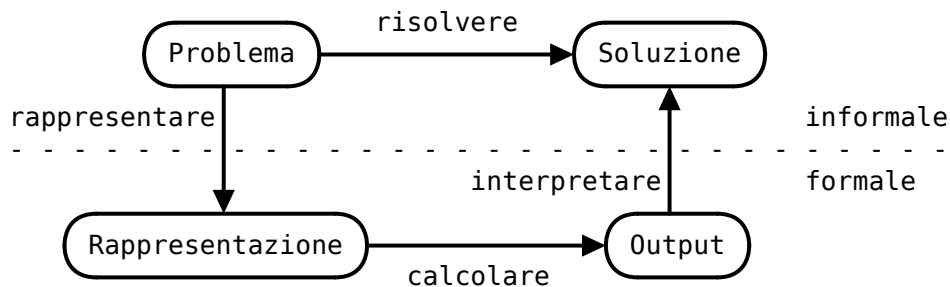
- come acquisire e rappresentare la conoscenza su un dominio
- come usarla per rispondere a domande e/o risolvere problemi

### Logica uso:

- Specifica precisa del *significato*
  - più importante della particolare sintassi
- *Notazione*: e.g. sintassi Datalog/Prolog
  - semplice
  - con un risvolto operativo (es. [GNU-Prolog](#))

## 2.1 Rappresentazione della Conoscenza

Problema (compito) e soluzione spesso disponibili in modo informale



Soluzione attraverso le macchine: necessità di formalizzazione:

- delineare il *problema* e determinare cosa costituisca una *soluzione*
- *rappresentarlo* in un linguaggio su cui la macchina possa ragionare
- far *calcolare* alla macchina un *output*, come risposta all'utente (o come sequenza di azioni messe in atto sull'ambiente)
- *interpretare* l'output come *soluzione* al problema

### 2.1.1 Conoscenza

#### Conoscenza

Informazione sul dominio da usarsi per risolvere problemi nel dato dominio:

- Uno *schema di rappresentazione* per le macchine va formalizzato: *base di conoscenza* (KB)
  - rappresentazione interna al sistema

Obiettivi:

- *ricchezza espressiva* sufficiente alla risoluzione del problema
- *vicinanza* ai termini del problema:
  - compatta, naturale, manutenibile
  - facilita l'individuazione della relazione dominio e sua rappresentazione per determinarne la correttezza
    - piccole modifiche al problema → piccole modifiche alla rappresentazione
- atta ad essere *elaborata* in modo *efficiente*
  - capacità di esprimere caratteristiche del problema sfruttabili per risparmiare risorse
    - eventualmente con compromessi con l'accuratezza
- *acquisibile* da utenti, dati e/o esperienza pregressa

Schemi di rappresentazione: possono partire da alcuni degli obiettivi e man mano essere estesi

- ad es., da quelli per l'apprendimento si possono espandere per ammettere maggiori capacità



risolutive o di inferenza

- alcuni puntano sull'espressività per poi aggiungere inferenza e capacità di apprendimento
- alcuni partono dalla trattabilità dell'inferenza per essere rese più tardi più naturali e facilmente acquisibili

Domande-guida:

- Cosa costituisce una soluzione del problema?  
Quanto dev'essere buona una soluzione?
- Come rappresentare il problema?  
Quali distinzioni vanno fatte per risolverlo?  
Quale conoscenza specifica del mondo è necessaria?  
Come acquisire tale conoscenza da esperti o dall'esperienza?  
Come verificarla, mantenerla e migliorarla?
- Come si calcola un output interpretabile come soluzione al problema?  
Cosa va minimizzato in termini di prestazioni (caso medio/pessimo)?  
È essenziale che un utente possa capire la risposta fornita?

## 2.1.2 Soluzioni

Come nell'ing. del SW anche qui il progettista deve definire cosa costituisca una **soluzione**:

- Spesso come *raffinamento della specifica* del problema
  - problemi poco specificati ma lacune che non possono essere colmate arbitrariamente
- Automatizzare forme di *senso comune* può fornire le motivazioni per diversi problemi da risolvere:
  - conclusioni dettate dal senso comune riguardanti *assunzioni non dichiarate*
- Dato un problema ben definito, nel passo successivo occorre stabilire se conti il fatto che la risposta fornita sia errata o incompleta
- A volte non serve una generica soluzione ma la migliore secondo un dato criterio

*Classi di soluzioni comuni:*

### Soluzione Ottimale

migliore secondo una *misura di qualità*:

- tipicamente *ordinale*
- anche *cardinale*, lì dove contano anche grandezze relative in situazioni come la combinazione di criteri multipli o nel ragionamento sotto incertezza,
  - Ad es. l'*utilità*

### Soluzione Soddisfacente

soluzione buona secondo una descrizione di quelle adeguate:

- quando non serve la soluzione migliore (perché costosa)

## Soluzione Approssimativamente Ottimale

di qualità *prossima* alla migliore ottenibile teoricamente:

- data una misura cardinale che ammetta approssimazioni
- conveniente per ragioni di efficienza a seconda degli algo.

## Soluzione Probabile

quella che *verosimilmente* può essere una soluzione, con un certo *grado* di certezza:

- forma di approssimazione precisa di soluzione soddisfacente
- a volte si distingue il tasso di errore per *falsi-positivi* (proporzione di risposte fornite non corrette) da quello di *falsi-negativi* (proporzione di risposte corrette non fornite)
  - categorie non esclusive

## 2.1.3 Rappresentazioni del Problema

Sistemi simbolici (computer, menti umane, ecc.)

### Simbolo

pattern significativo manipolabile come singola unità

- ad es. parole scritte, frasi, gesti, segni su carta o sequenze di bit

Un *sistema di simboli* serve a creare, copiare, modificare e distruggere simboli

“A physical symbol system has the necessary and sufficient means for general intelligent action” — Newell & Simon<sup>2</sup>

Ipotesi (empirica) forte:

- ogni sistema intelligente è necessariamente un sistema simbolico fisico ed esso rappresenta tutto quello che serve perché un'azione possa dirsi intelligente
- non implica invece che serva un corpo per percepire e agire sul mondo

Un sistema intelligente manipola simboli per produrre azioni:

- simboli che si riferiscono ai vari oggetti del mondo
- ma altri simboli potrebbero rimandare a concetti utili che possono avere un significato per l'esterno o meno
  - stati interni del sistema

<sup>2</sup> Newell, A. and Simon, H.A. (1976). *Computer science as empirical enquiry: Symbols and search*. Communications of the ACM, 19: 113–126.

Un sistema di simboli serve a *modellare* il mondo

## modello del mondo

rappresentazione delle specifica di quanto risulti vero nel mondo o della sua dinamica:

- non serve un modello troppo dettagliato perché risulti utile
- i modelli sono *astrazioni*
  - rappresentano solo parte del mondo (eliminati i dettagli)
- *livello* di astrazione (diversa precisione nel dettaglio)
  - fornisce un *ordine parziale* d'astrazione
    - più basso il livello maggiori i dettagli
- un sistema può ammettere *più modelli*, anche in *contraddizione* fra loro
  - si giudicano in base all'*utilità* più che sulla correttezza

---

**Esempio** — robot con un modello della pianta d'un edificio ignorando le distanze, la sua grandezza gli angoli di sterzata ecc.

---

- potrebbe prendere in considerazione tali caratteristiche in una rappresentazione di basso livello

Fattori nella scelta del *livello d'astrazione*:

- descrizioni *high-level* più semplici per gli umani
  - specifica + comprensione
- descrizioni *low-level* più accurate e predittive
  - dettagli essenziali per la soluzione del problema
- più basso il livello, più è difficile il ragionamento
  - comporta più passi e più piani d'azione da scegliere
  - le info di basso livello da prendere in considerazione potrebbero essere sconosciute

Modello a più livelli d'astrazione:

- ad es. sistemi biologici / informatici
  - livello neurale
    - livello biochimico
    - livello chimico
    - livello fisico
- anche la Scienza stessa è strutturata a livelli gerarchici

Livelli comuni tra sistemi biologici e di calcolo:

- *livello della conoscenza* livello d'astrazione che considera quanto il sistema sa e crede e ritiene siano i suoi fini (riguardano il mondo esterno)
  - quello che sa ma non come ragiona
  - non si specifica come calcolare la soluzione o quale strategia adottare
- *livello simbolico* descrizione in termini di come ragiona il sistema (internamente sul mondo esterno)
  - per implementare il livello precedente si manipolano simboli per produrre risposte

- attestato da esperimenti dei cognitivisti

## 2.1.4 Ragionamento

La manipolazione di simboli produce risposte/azioni attraverso il **ragionamento**

- Rappresentazioni AI diverse dai programmi nei ling. tradizionali perché specificano *COSA* dev'essere calcolato e non *COME* vada fatto
  - ad es. specificare di voler comprendere e comunicare lo stato di salute, il disturbo, la malattia più probabile per un dato paziente
- Il ragionamento si traduce spesso in una *ricerca* in uno spazio di possibilità
- Decisioni basate su 3 aspetti:
  1. calcolo in fase di progettazione del sistema
    - svolto dal suo progettista
  2. computazione da farsi prima dell'osservazione del mondo (e conseguente risposta)
    - svolto dal sistema prima di agire
    - può comportare una fase di compilazione e di apprendimento:
      - offline, si considerano conoscenza di fondo (BK) e dati, che sono compilati in forma usabile: base di conoscenza
      - utile sia in fase di progettazione sia offline
  3. computazione in fase di risposta/azione
    - svolta durante l'osservazione dell'ambiente (e la conseguente azione/risposta)
    - le osservazioni (ottenute online) sono usate assieme alla base di conoscenza per determinare il da farsi

Vanno distinte la conoscenza nella mente del progettista da quella nella mente del sistema intelligente; casi limite:

- agente altamente specializzato  
lavora bene nel suo ambiente ma è inutilizzabile al di fuori
  - es. termostato
- agente molto flessibile al variare di ambienti e compiti (a run-time)
  - es. insetti
  - molto più difficili da progettare

*Strategie di costruzione:*

- semplificare gli ambienti e costruire sistemi di ragionamento complessi per tali ambienti
  - es. robot in fabbrica
  - si possono dimostrare proprietà dei sistemi o ottimizzarli per particolari situazioni
  - serve un modello del sistema e del suo ambiente
  - un osservatore/progettista deve poter ragionare su tale modello
    - ad es. il progettista può dimostrare se il sistema raggiungerà uno scopo, se eviterà di imbattersi in situazioni pericolose, se potrà restare immobilizzato da qualche parte

(*liveness*), o se potrà a termine tutti i suoi compiti (*fairness*)

- agenti semplici per ambienti complessi/naturali
  - ad es. ispirandosi agli insetti
  - aggiungere più capacità di ragionamento al crescere della complessità dei compiti
  - vantaggio: similarità con gli ambienti umani

## 2.2 Sistemi RR

---

Un **sistema per la rappresentazione e il ragionamento** (RRS) è composto da:

- un *linguaggio* per comunicare con la macchina
- un modo per assegnare un *significato* al linguaggio
- *procedure* per fornire per via automatica *risposte* a specifici input al sistema espresse in tale linguaggio

### 2.2.1 Linguaggio

Livelli:

- Linguaggio *di programmazione* di non alto-livello (Fortran, C/C++) il significato di frasi è puramente definito in termini dei passi da svolgere per eseguire un programma
  - computazione determinata dall'input e dal programma
  - difficile mappare un enunciato informale di un problema su una sua rappresentazione per i RRS
- Linguaggio *naturale* (Italiano, Inglese) per le frasi che descrivono il problema
  - facile mapping sulla rappresentazione del problema
  - computazione da eseguire sulla macchina alquanto difficile da determinare
- *Compromesso*
  - non grande distanza tra specifica naturale del problema e sua rappresentazione
  - si richiede che il calcolo richiesto da un determinato input sia determinabile efficacemente

*Fattori* da considerare nella scelta del linguaggio:

- specifica dei problemi
- significato da associare
- computazione appropriata all'input

### 2.2.2 Esempio | DB come RSS

In un database:

- si possono specificare alla macchina fatti circa un dato dominio
- per poi porre interrogazioni per ritrovarli
- semantica: consente di decidere sulla *verità* di date informazioni in una *base di conoscenza*
  - le informazioni passano dal livello dei dati a quello della conoscenza

Si prenderanno in considerazione RRS forme di rappresentazione più flessibili e procedure di risposta alle domande più sofisticate:

- tipicamente basate su ricerca su tabelle (indicizzate)
- non si può chiedere cos'altro sia verso, o verosimile per il dato dominio

### 2.2.3 Ontologie e Concettualizzazioni

Prerequisito fondamentale per l'uso di RRS:

*rappresentazione del dominio* del problema da risolvere / compito da svolgere

- di che *tipi di cose* consista il dominio
- da quali *legami* siano correlate

Nessuna teoria generale ma ricette specifiche per i vari domini, tuttavia:

- Descrizione del mondo in termini di individui (le cose) e relazioni tra di essi → **ontologia**
- *Nozione di relazione* intesa che include proposizioni vere o false indipendentemente dai singoli individui
  - proprietà dei singoli individui
    - o relazioni tra due (o più) di essi
  - qualunque *cosa nominabile*, sia concreta sia astratta, può essere considerata come individuo
    - ad es., persone, colori, emozioni, numeri, e istanti temporali
  - Cosa sia una “cosa” dipende dall'osservatore
    - tanto quanto risulta una proprietà del mondo
      - osservatori diversi, o anche uno stesso osservatore ma con diversi fini, può definire il mondo in modi diversi
- Per ogni compito/dominio, vanno identificati individui e relazioni specifici per esprimere quello che è vero nel modo in esame
  - da ciò dipende l'abilità nel risolvere i problemi nel dominio
  - si assume che l'*ingegnere della conoscenza* che modella il dominio decida la sua ontologia
    - sistema SW capace di decidere l'ontologia raggiungerebbe un'intelligenza a livello di quella umana
      - target ancora difficile per il machine learning

## 3 Dimensioni della Complessità

---

**Complessità** degli agenti/dei sistemi variabile:

- dai termostati alle aziende (con finalità diverse e immerse in ambienti competitivi)

**Dimensioni** della complessità nella progettazione:

- da combinare anche se studiate separatamente
- definiscono uno *spazio di progettazione* di sistemi intelligenti
  - diversi sistemi a seconda dei loro valori
- forniscono una *suddivisione sommaria* dello spazio di progettazione
  - ma vanno aggiunte altre scelte

### 3.1 Modularità

---

**Modularità** dimensione entro la quale un sistema può essere decomposto in *moduli* interagenti considerabili separatamente:

- serve a *dominare la complessità*
  - evidente nei sistemi di interesse per l'informatica ma in ogni campo in cui si trattino organizzazioni
- tipicamente espressa come decomposizione gerarchica
  - se i moduli sono organizzati in moduli più piccoli a loro volta organizzati gerarchicamente fino al livello delle op. primitive
  - l'astrazione procedurale e l'OOP servono a sfruttare modularità e astrazione

Strutture possibili:

- *piatta*: nessuna struttura organizzativa;
- *modulare*: sistema decomposto in moduli che interagiscono considerabili separatamente;
- *gerarchica*: sistema modulare, dove i moduli possono essere decomposti in sotto-moduli interagenti, a loro volta decomponibili gerarchicamente

*Ragionamento*:

- struttura piatta / modulare: *singolo* livello di astrazione
- struttura gerarchia: *più* livelli d'astrazione
  - più basso il livello nella gerarchia più basso il livello di astrazione

---

**Esempio** — organizzazione di una vacanza

---

All'inizio si può ignorare l'aspetto gerarchico per concentrarsi anche sulle altre dimensioni della complessità

- utile per problemi dalle modeste dimensioni

Si passa alla struttura gerarchica quando i problemi / i sistemi diventano più complessi

## 3.2 Schema di Rappresentazione

---

### Schema di Rappresentazione

riguarda la *descrizione del mondo*

- stati differenti del mondo hanno un impatto sul comportamento del sistema
  - fattorizzabile in stato *interno* (stato delle credenze) e stato *dell'ambiente*
  - a livello di massima semplicità un sistema ragiona esplicitamente in termini di stati identificati individualmente
- 

### Esempio — termostato (6 stati):

- stati *interni*: spento / riscaldamento
  - stati *ambiente*: freddo, confortevole, caldo
    - ambiente freddo → deve passare o restare in modalità riscaldamento
    - ambiente molto caldo → può passare nello stato spento
    - ambiente è confortevole → dovrebbe rimanere nello stato (interno) corrente
  - *azioni*: riscalda nello stato riscaldamento altrimenti è spento
- 

Conviene ragionare in termini di *caratteristiche* degli stati o *proposizioni* booleane anziché enumerarli

- Uno stato può essere descritto in termini di **caratteristiche** (*feature*) con un valore per ogni stato
- 

### Esempio — sistema per la domotica

- caratteristiche:
    - posizione degli interruttori,
    - stato di ogni interruttore (in funzione/in corto, fuori uso)
    - stato dei punti luce
      - es. pos\_s2 con valore up se l'interruttore s2 è acceso e down se spento
  - stato della casa descritto in termini dei valori di ciascuna delle caratteristiche
- 

- Una **proposizione** è una caratteristica *booleana* (valori *vero/falso*)
    - NB 30 proposizioni codificano  $2^{30} \approx 10^9$  stati
      - più facile specificare e ragionare con 30 proposizioni che con oltre un miliardo di stati
    - rappresentazione *compatta* degli stati:
      - indica che sono state comprese *regolarità* importanti sul dominio
-



**Esempio** — sistema per il riconoscimento delle lettere:

- immagini  $b/n$ ,  $30 \times 30$
  - *azione*: determinare le lettere tracciate
  - **NB**  $2^{900}$  stati dell'immagine quindi  $26^{2^{900}}$  funzioni dalle immagini alle lettere
    - proibitivo rappresentare tutte queste funzioni in termini di spazio degli stati
    - meglio definire alcune caratteristiche dell'immagine (es. segmenti) e definire le funzioni in termini di tali feature
- 

Nella descrizione di mondi complessi le caratteristiche dipendono dalle *relazioni* e dagli *individui*

- una relazione su un singolo individuo è una **proprietà**
  - si può definire una *caratteristica* per ogni possibile relazione tra gli individui
- 

**Esempio** — domotica (cont.):

- *individui*: luci e interruttori
  - *relazioni*: posizione e connesso\_a
    - invece della caratteristica `posizione_s1 = up`, si può usare la relazione (proprietà) `posizione(s1, up)`
    - consente di ragionare su tutti gli interruttori o sapere quelli che possono essere usati, ecc.
- 

**Esempio** — iscrizione studenti ai corsi:

- voto: relazione/feature che assegna il voto a ognuno degli studenti per ogni corso seguito
  - superato: relazione sulle coppie studente-corso, che dipende da voto
  - più facile ragionare in termini di singoli studenti, corsi e voti con tali relazioni
    - una volta definita la dipendenza tra superato e voto, la si applica a ogni studente/corso
    - definibile PRIMA di conoscere gli studenti
- 

Può essere ancora più conveniente trattare *descrizioni relazionali* in termini di individui e relazioni invece di caratteristiche e proposizioni:

- ad es., con una sola relazione binaria e 100 individui si possono rappresentare  $100^2 = 10000$  proposizioni e  $2^{10000}$  stati
  - ragionando su relazioni e individui, si possono considerare intere *classi di individui* senza enumerarne caratteristiche/proposizioni, o addirittura i numerosissimi stati
  - un sistema potrebbe dover ragionare su insiemi di individui *infiniti*
    - ad es. l'insieme dei numeri, o l'insieme di tutte le stringhe
    - impossibile in termini di stati o feature
- 

Riassumendo, nella dimensione dello **schema di rappresentazione**, si può ragionare in termini di:

- *stati*

- *feature* proposizionali o
- *descrizioni relazionali*
  - su individui e relazioni

### 3.3 Orizzonte

---

L'**orizzonte** misura quanto lontano (nel tempo) sia prevista la pianificazione del lavoro, ossia quanto in avanti ci si spinga a considerare le conseguenze delle azioni:

- ad es. richiamo animali e ricompensa immediata
  - un cane di solito non ha finalità che vanno molto in là nel futuro al contrario delle persone
- a volte non serve considerare il tempo nel ragionamento ma si va avanti per *stage* successivi

In tale dimensione, si possono avere sistemi/agenti:

- senza pianificazione: che non considerano il futuro quando prendono decisioni (sulle azioni)
  - il fattore-tempo non viene coinvolto
- a orizzonte finito: interessano solo un numero prefissato di passi
  - ad es., dottore che deve curare il paziente ma c'è tempo per dei test, quindi due fasi:
    - test e cura
  - nel caso in cui conti una sola fase l'agente/algo. viene detto *greedy* o *miope*
- orizzonte indefinito: si considera un numero di passi finito, ma non predeterminato
  - ad es., agente che deve recarsi in un luogo lontano ma non sa quanti passi ci vorranno (e.g. n.ro di treni da prendere)
- orizzonte infinito: sempre attivo (*processo*)
  - ad es., modulo di stabilizzazione di un robot: se si fermasse una volta raggiunta la stabilità dopo il robot potrebbe cadere di continuo

### 3.4 Incertezza

---

A volte si deve progettare tenendo conto dell'*incertezza* insita nel dominio considerato

Due dimensioni:

- sulla percezione / osservazione
- sugli effetti delle decisioni / azioni

#### 3.4.1 Osservazione Incerta

A volte l'*osservazione diretta* dello stato del mondo è possibile

- ad es. nei giochi da tavolo, carte

Più spesso accade che la percezione dello stato sia *difettosa* o parziale / indiretta:

- al più si può avere una *distribuzione* di probabilità sull'insieme degli stati possibili su quanto si osserva
- ad es. dati i sintomi, un medico potrebbe non sapere esattamente cos'abbia un paziente ma potrebbe diagnosticare malattie con diversi livelli di certezza

L'**incertezza sulla percezione** riguarda la possibilità di determinare lo stato del mondo attraverso le osservazioni:

- *pienamente osservabile*: si può conoscere lo stato dalle osservazioni
  - assunzione spesso fatta per ragioni di trattabilità dei problemi
- *parzialmente osservabile*: non si osserva direttamente lo stato
  - più stati possibili possono portare alle stesse osservazioni
  - oppure le osservazioni sono *rumorose* (noisy)

### 3.4.2 Effetto Incerto

A volte è possibile che si conosca sempre l'effetto delle azioni/decisioni

- ossia, dato uno stato e un'azione, si può predire precisamente lo stato risultante dall'applicazione della decisione/azione
  - ad es., lavorando con un file system si conoscono gli effetti di una cancellazione di un file noto il suo stato

A volte è difficile fare tali previsioni

- al più si può avere una distribuzione di probabilità sugli effetti possibili
  - ad es., nel richiamare il proprio cane, noto il suo stato e l'esperienza pregressa si ha un'idea su quello che farà
    - a volte funziona anche con cani altrui

La dimensione dell'**incertezza sugli effetti** prevede che la loro *dinamica* possa essere

- *deterministica* — stato risultante determinato esattamente dall'azione e dallo stato precedente
- *aleatoria* — probabilità sui possibili stati risultanti
  - ha senso solo se il mondo è completamente osservabile
    - altrimenti sistema stocastico modellato come deterministico ma con effetti che dipendono da feature non osservate

## 3.5 Preferenza

---

Gli agenti sono spesso utilitaristici:

- scelta di un'azione dettata da risultati attesi più desiderabili

- *finalità semplici* o *preferenze complesse*
  - stato da raggiungere o proposizione da avverare
  - ad es. il medico può tenere in conto l'aspettativa e la qualità di vita, costi (per sé, il paziente, la società), la possibilità di dover giustificare le decisioni in caso di giudizio legale, ...

La dimensione delle **preferenze** si divide in

- *finalità*, da raggiungere (*achievement goal*) in uno stato finale o di conservazione (*maintenance goal*) in ogni stato visitato
  - ad es., un robot può voler prendere alcuni oggetti, ma senza mettere in disordine la stanza o far male agli altri
- *preferenze complesse*: compromessi sul vantaggio derivate dei vari risultati, possibilmente anche in momenti diversi
  - preferenza *ordinale*: conta solo l'ordine
  - preferenza *cardinale*: conta anche la grandezza del valore
  - ad es., si preferisce il cappuccino al caffè e il caffè al tè (ordinale)  
compromesso tra il tempo d'attesa e il tipo di bevanda (cardinale)

### 3.6 Numero di Agenti

---

Difficoltà aggiuntive degli ambienti in cui vi siano altri agenti

- occorre ragionare sugli altri agenti secondo *strategie*
  - gli altri potrebbero cercare di confondere e manipolare o potrebbero cooperare
  - spesso conviene agire in modo casuale perché gli altri adottano strategie deterministiche
- anche quando si cooperi e vi sia un fine comune, il problema della *coordinazione* e della *comunicazione* rende il ragionamento multi-agente più arduo
  - ignorare le strategie degli altri potrebbe non risultare il modo migliore di ragionare

Dal punto di vista del singolo agente,  
la dimensione del **numero di agenti** prevede:

- il ragionamento da *agente singolo*: si assume che gli altri siano parte dell'ambiente
  - ragionevole se non ci sono altri agenti o se gli altri non cambieranno il comportamento in base alle sue azioni
- ragionamento *multi-agente*: si prende in considerazione il ragionamento altrui
  - in caso di agenti intelligenti i cui fini/ le cui preferenze dipendano, in parte, da quello che si fa se l'agente può comunicare con gli altri
  - più difficile se gli agenti possono agire simultaneamente o se l'ambiente sia solo parzialmente osservabile

### 3.7 Apprendimento

---

Il progettista può avere un buon modello dell'agente/sistema e del suo ambiente ma spesso non succede:

- si devono usare *dati da esperienze passate* e altre sorgenti di conoscenza per migliorare il modello e prendere migliori decisioni (sul da farsi)

La dimensione **apprendimento** determina se

- la *conoscenza* sia *data*
- la *conoscenza* vada *appresa* (dai dati o da passata esperienza)

Apprendere = trovare il modello migliore che si adatti ai dati

- caso semplice: regolare un insieme fisso di *parametri*
- caso più difficile: scegliere preliminarmente la migliore *rappresentazione*
  - feature, relazioni

Problematiche aggiuntive:

- utilizzo di *conoscenza di fondo* (BK)
- *selezione* dei dati da raccogliere
- *rappresentazione* dei dati e dei modelli
- *selezione* dei *learning bias* appropriati
- utilizzo della conoscenza appresa per modificare le azioni dell'agente

### 3.8 Limiti sulle Risorse Computazionali

---

Spesso ci sono limiti sulle risorse computazionali che impediscono di prendere le migliori decisioni sulle azioni da svolgere

- non è possibile trovare la migliore decisione / azione in modo sufficientemente rapido date le limitazioni della memoria
  - ad es., potrebbe essere inutile attendere 10' per derivare qual era la decisione migliore 10' prima
- spesso sono necessari compromessi sulla qualità della soluzione da cercare
  - meglio una soluzione ragionevole ma rapida che una migliore quando è troppo tardi perché il mondo esterno cambia nel frattempo

La dimensione dei **limiti sulle risorse computazionali** determina se un sistema abbia:

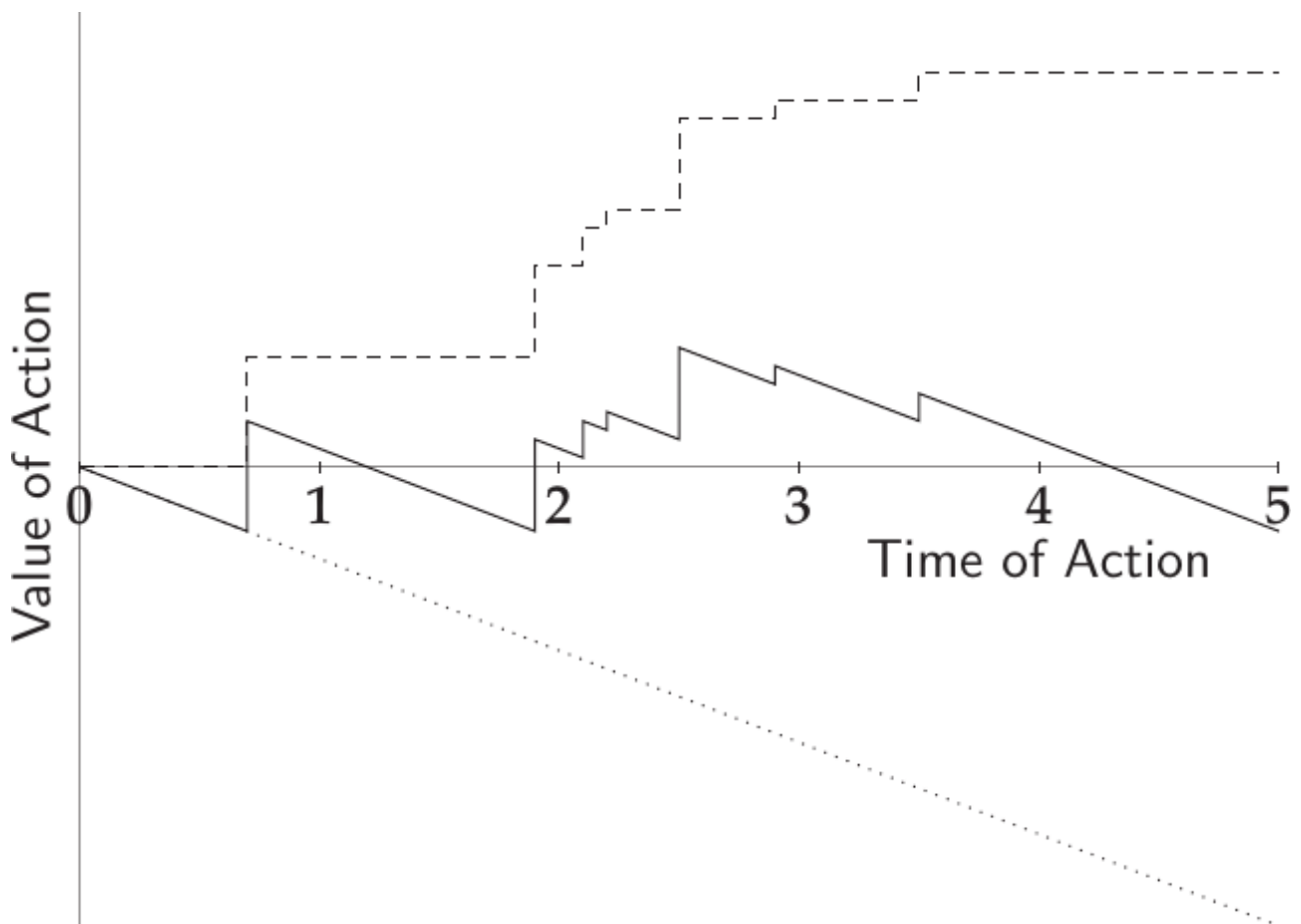
- una *razionalità perfetta*: ragiona per prendere la migliore decisione senza tener conto delle risorse limitate
- una *razionalità limitata*: ragiona per prendere la migliore decisione date le risorse limitate a disposizione

I *limiti* riguardano:

- il *tempo*
- la *memoria*
- la *precisione* (numerica) approssimata

Un algoritmo *anytime* produce soluzioni che migliorano con il tempo:

- in qualunque momento produce la miglior soluzione corrente
- con ancora più tempo questa potrebbe migliorare
  - si assicura che la qualità non decresca consentendo di immagazzinare la soluzione migliore trovata restituirla su richiesta
- l'attesa può avere un costo:
  - a volte meglio decidere/agire prima di aspettare una soluzione probabilmente migliore



Qualità di una soluzione in funzione del tempo per un algoritmo *anytime* — fonte [2]

In caso di razionalità limitata, si deve decidere se aspettare o pensare un po' di più

- difficile giudicare la politica migliore
- anche il tempo speso in tale decisione è da sottrarre a quello di ricerca della soluzione
- motiva il cosiddetto *ragionamento approssimato*

### 3.9 Interazione delle Dimensioni

Dimensione	Valori
Modularità	piatta, modulare, gerarchica
Schema di Rappresentazione	stati, feature, relazioni
Orizzonte	senza pianificazione, a stage finite, indefinite, infinite
Incertezza sull'Osservazione	pienamente osservabile, parzialmente osservabile
Incertezza sull'Effetto	deterministica, stocastica
Preferenze	finalità, preferenze complesse
Apprendimento	conoscenza data, conoscenza appresa
Numero di Agenti	singolo-agente, agenti multipli
Limiti Risorse Computazionali	razionalità perfetta, razionalità limitata

Non si possono studiare in modo indipendente perché soggette a *interazioni complesse*

- Rappresentazione e modularità
  - moduli semplici in una gerarchia tanto da poter ragionare in termini di un insieme finito di stati, mentre altri livelli di astrazione richiedono un ragionamento su relazioni
    - ad es., robot consegne:
      - modulo di bilanciamento con pochi stati
      - modulo che deve tener conto delle priorità di consegna di più colli a diverse persone deve ragionare su più individui (e.g., persone, pacchi, stanze,...) e loro relazioni
      - a livello più alto, un modulo può ragionare sull'intera attività giornaliera differenziando con pochi stati le diverse fasi del giorno (e.g., stati impegnato, disponibile e ricarica)
- L'orizzonte interagisce con la modularità
  - ad es. ad alto livello, un cane può ricevere una ricompensa quando risponde al richiamo
    - quando decide dove mettere le zampe, potrebbe volerci del tempo prima di ricevere il boccone per cui l'orizzonte potrebbe essere indefinito
- L'incertezza sull'osservazione probabilmente ha l'impatto maggiore sulla complessità del ragionamento:
  - molto più facile ragionare quando si conosce lo stato del mondo
  - l'incertezza sugli individui e le relazioni è più complicata da trattare
- L'incertezza sugli effetti interagisce con la modularità:
  - ad un dato livello della gerarchia una decisione può essere deterministica mentre ad un altro potrebbe essere stocastica
- I modelli di preferenza interagiscono con l'incertezza
  - serve un compromesso tra soddisfare un fine importante con una certa probabilità o un fine meno desiderabile con ancor maggiore probabilità

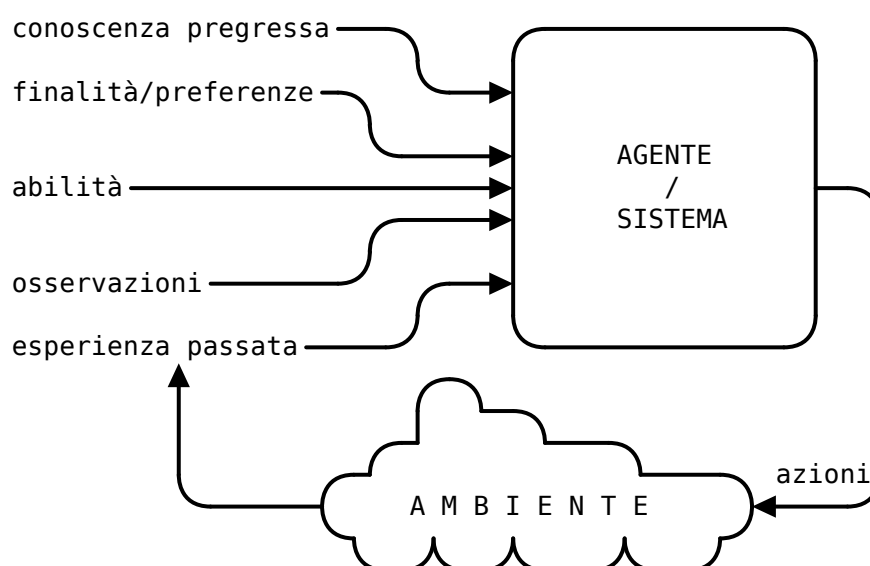
- Agenti multipli si accompagnano bene alla modularità
  - un agente singolo può essere progettato attraverso agenti multipli che interagiscono condividendo un fine comune di rendere intelligente il comportamento dell'agente di livello superiore
    - secondo alcuni<sup>3</sup>, l'intelligenza è una caratteristica emergente da una “società” di agenti non intelligenti
- Apprendimento e rappresentazione
  - spesso si riduce all'apprendimento da feature:
    - determinare i valori che determinano le migliori predizioni per i valori di un'altra feature (problemi di classificazione, regressione)
  - ma si può lavorare anche su individui e relazioni
  - numerosi lavori sull'apprendimento di gerarchie, apprendimento in domini parzialmente osservabili e attraverso agenti multipli
    - tutte queste modalità interagiscono con diverse altre dimensioni
- Modularità e razionalità limitata promettono di rendere il ragionamento più efficiente
  - il formalismo può diventare più complicato
  - ma per costruire sistemi complessi sono utili
    - la suddivisione in componenti più piccole
    - fare approssimazioni per poter decidere in tempi accettabili e in regime di memoria limitata

<sup>3</sup> Minsky, M. (1986). *The Society of Mind*. Simon and Schuster

## 4 Applicazioni

---

**Agenti Intelligenti:** percezione—ragionamento—azione  
immersi in un *ambiente*





- robot: unità di calcolo+sensori+attuatori
- sistema esperto: puramente computazionale (SW), info fornita da umani
  - diagnostica
  - infobot

## 4.1 Infobot

---

Un **infobot** è una specie di *robot*, che interagisce con un ambiente informativo anziché fisico

Compiti:

- estrarre info da una rete di sorgenti informative
  - Internet | enciclopedie multimediali
- determinare quale info serva per una query
  - in un linguaggio formale, da utenti esperti
  - in linguaggio naturale da un utente generico
- individuare le sorgenti informative, trovare le informazioni necessarie e presentarle in modo utile per l'utente

**Input** (cfr. figura):

- *Conoscenza pregressa* significato delle parole, tipi di sorgenti informative e come accedervi
- *Esperienza passata* informazione da ottenere, velocità dei vari server e info sulle preferenze-utente
- *Finalità* informazione da ricercare e compromessi sul dispendio di risorse e tra volume e qualità
- *Osservazioni* info presente sui siti al momento, link disponibili e carico sulle varie connessioni
- *Abilità* azioni primitive di cui è capace

**Output** (info utile alla comprensione da parte dell'utente, anche in caso di info mancante)

L'infobot deve essere capace di:

- Derivare info implicita nella/e base/i di conoscenza, (e interagire in NL)
- Cercare info rilevante in una varietà di basi di conoscenza
- Trovare buone rappresentazioni della conoscenza che assicurino una computazione efficiente delle risposte
- Spiegare come sia stata derivata una risposta o perché alcune info non erano disponibili
- Trarre conclusioni in caso di mancanza di conoscenza, determinare eventuali conflitti ed essere in grado di inferire conoscenza disgiuntiva
- Usare ragionamento per default sulle possibili diverse fonti di informazione
- Fare compromessi tra sorgenti a basso costo ma poco affidabili e sorgenti più costose ma più complete
- Imparare quale conoscenza sia disponibile e dove e l'info che l'utente ....

Due diversi infobot:

**unibot**

interagisce con un DB: info su corsi, tempistica, propedeuticità, regole, valutazioni

**webbot**

interagisce con il Web, ricerca info utili all'utente

Aspetto importante: *proattività*

## 4.2 Caratteristiche Comuni

---

Ad un certo livello di astrazione, 4 compiti:

**Modellazione dell'ambiente**

dev'essere in grado di modellare come le info debbano essere ottenute, quali risposte siano ammesse alle domande e quali info siano necessarie in base alla richiesta

**Ragionamento sulle evidenze (o percezioni)**

determinare com'è fatto il dominio, dove sono disponibili le info, data una info parziale sul contenuto delle varie sorgenti

**Azione**

dato un modello del mondo e uno scopo, determinare cosa vada fatto per raggiungerlo: consultare una o più basi di conoscenza per estrarre informazioni

**Apprendimento dall'esperienza passata**

imparare le caratteristiche dell'ambiente, ad es. i colli di bottiglia nella comunicazione in rete, e come risolvere i problemi più velocemente

C'è interazione nello svolgimento di tali compiti: vanno studiati in modo organico

## Riferimenti Bibliografici

---

- [1] D. Poole, A. Mackworth, R. Goebel: *Computational Intelligence: A Logical Approach*. Oxford University Press
- [2] D. Poole, A. Mackworth: *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press
- [3] S. J. Russell, P. Norvig: *Artificial Intelligence* Pearson. 3rd Ed. - cfr. anche ed. Italiana
- [4] J. Sowa: *Knowledge Representation: Logical, Philosophical, and Computational Foundations* Brooks Cole/Cengage

Link

[KE] [https://en.wikipedia.org/wiki/Knowledge\\_engineering](https://en.wikipedia.org/wiki/Knowledge_engineering)

[GNU-Prolog] <http://www.gprolog.org/>

*formatted by Markdeep 1.03* 