

# 1. Ingegneria della Conoscenza

## Dispensa ICon

versione: 19/10/2024, 11:50

**Sistemi Intelligenti · Dimensioni dello Spazio di Progettazione · Progettazione di Sistemi Intelligenti · Sistemi Intelligenti Basati su Conoscenza**

Alcune possibili definizioni dell'**Ingegneria della Conoscenza**:

*“Knowledge engineering studies the representation, acquisition, reasoning, decision-making, and application of knowledge, including big data, machine learning, data mining and knowledge discovery, uncertain reasoning, knowledge mapping, machine theorem proving, expert system, machine game, digital library, etc.” [Shi21]*

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

## 1 Sistemi Intelligenti

---

L'**Intelligenza Artificiale** (AI) può essere definita come la disciplina che studia sintesi e analisi degli *agenti computazionali*, ossia dei sistemi che agiscono in modo *intelligente*:

Essi svolgono azioni in un ambiente (come fanno, ad esempio, batteri, animali, termostati, aerei, robot, aziende, paesi) e vanno giudicati solo per le loro azioni, quindi a parità di effetti si apprezzerà la stessa efficacia. La loro intelligenza va misurata secondo una *scala* adeguata.

Aspetti salienti dell'*azione intelligente* sono i seguenti:

- l'appropriatezza rispetto alle circostanze, agli obiettivi, ai limiti (della percezione o del calcolo);
- la considerazione delle conseguenze a breve / lungo termine delle azioni;
- la apprendimento dall'esperienza;
- la flessibilità rispetto ai cambiamenti (dell'ambiente, degli obiettivi).

## Agenti Computazionali e Sistemi Intelligenti

Un **agente computazionale** è un agente le cui decisioni / azioni siano spiegabili in termini di computazione.

Esso è implementato su sistema *fisico* e presenta le seguenti *limitazioni*:

- non è onnisciente né onnipotente;
- le capacità di osservazione sono specializzate e vincolate dal dominio applicativo;
- la sua memoria è finita e a tempo limitato.

# Relazione fra AI e Altre Discipline

Un classico parallelo fra *Intelligenza Artificiale* e *Aerodinamica* si riassume nell'analogia fra le risposte alle seguenti domande:

*"È possibile creare macchine in grado di pensare?"*

*"È possibile creare macchine in grado di volare?"*

Così come non c'è stato bisogno di copiare nel dettaglio il volo degli uccelli per costruire aerei in grado di assolvere perfettamente al compito, parimenti alcuni compiti tipici dell'intelligenza umana si possono svolgere costruendo macchine (hw/sw) per tali scopi.

La disciplina dell'AI può essere inquadrata sotto diverse prospettive:

- prospettiva *scientifica*: AI come disciplina finalizzata a comprendere i principi del comportamento intelligente
  - analizzando sistemi naturali e artificiali;
  - formulando *teorie (ipotesi confutabili)* sulla costruzione di agenti intelligenti che possano essere supportate da *implementazioni* per la loro *verifica sperimentale*;
- prospettiva *ingegneristica*: AI come studio di nuove tecnologie atte risolvere specifici problemi tipicamente risolti da agenti umani:
  - progetto e sintesi di *sistemi intelligenti basati su conoscenza (KBS) testabili*: qualità valutabili attraverso metodi standard dell'ing. informatica;
  - applicazioni costruite per la loro *utilità* in un dato dominio.

Altri obiettivi dell'AI:

- **intelligence augmentation**: accrescimento dell'intelligenza e della creatività umane. Alcuni esempi:
  - sistema diagnostico: aiuta i medici prendere decisioni migliori;
  - motore di ricerca: aiuta a superare i limiti della memoria umana;
  - traduzione automatica: aiuta la gente a comunicare.
- **integrazione** dei compiti agenti umani e sistemi intelligenti che collaborano alla soluzione di problemi; si parla di modalità **human-in-the-loop**:
  - le decisioni di agenti artificiali possono fornire consigli utili allo svolgimento dei compiti di quelli umani;
  - gli uomini possono guidare o dare feedback ad agenti artificiali, specie per decisioni da prendere rapidamente o ripetutamente (lo specifico delle macchine).

## Intelligenza: Naturale e Artificiale

Si suppone che l'intelligenza del sistema / dell'agente sia definita solo dal suo comportamento esterno: se un agente *si comporta intelligentemente* allora è intelligente.

Intelligenza <i>naturale</i>	Intelligenza <i>artificiale</i>
<i>emerge</i> spontaneamente in natura	viene <i>progettata</i> per specifici scopi

In AI si mira alla progettazione di *artefatti SW intelligenti*, non generici ma utili a scopi precisi.

L'ipotesi fondamentale può essere riassunta con la seguente formula:

i.e. il ragionamento si riduce a mera computazione.

Essa è collegata alla tesi di *Church-Turing*: esiste un *livello di astrazione* nel quale il ragionamento coincide con la manipolazione di simboli [NS76] In tal caso l'aggettivo *artificiale* andrebbe interpretato come *computazionale*.

## Conoscenza e Senso Comune

L'interazione di un agente situato in un ambiente necessita di **senso comune**

*“the ability to make effective use of ordinary, everyday, experiential knowledge in achieving ordinary, practical goals” — [BL22]*

In termini generali potremmo definire la **conoscenza** come informazione stabile, non transitoria. Normalmente essa non è dichiarata esplicitamente: non risulta necessario esplicitare quello che tutti sanno. Ad es. conoscenze pratiche come andare in bici, riconoscere un volto. Nel caso delle macchine si mira alla *formalizzazione* della conoscenza di senso comune.

## Intelligenza e Sistemi Complessi

Oltre che negli uomini, forme di intelligenza possono essere riscontrate anche negli animali (perfino nei batteri) o addirittura nelle **organizzazioni** risultando addirittura superiori a quella del singolo agente.

Vale il noto dualismo fra visione *olistica* e visione *riduzionistica*: una *comunità organizzata* può esibire un *comportamento intelligente*, ad es. si possono considerare i modelli *ant colony* (*formicai*), o anche la *swarm intelligence* (*sciami*) e gli algoritmi correlati.

Vi è un'interazione fra le fonti dell'intelligenza naturale:

- *biologia*: capacità evolutive per sopravvivere in diversi ambienti;
- *cultura*: sapere ereditato/appreso, ad es. un linguaggio e altri strumenti/concetti utili;
- *apprendimento continuo*: di nuove conoscenze/capacità per l'intera esistenza.

Nel progetto di un agente occorre decidere cosa programmare o imparare: è inverosimile partire da zero e imparare tutto quello che serve a portare a termine il proprio compito; più utile imparare a *migliorare* il comportamento.

## 2 Dimensioni dello Spazio di Progettazione

---

La *complessità* dei sistemi intelligenti varia e dipende dalle finalità e dai diversi contesti / ambienti in cui sono immersi. Ciò vale dai casi semplici, ad es. i termostati, a quelli più complessi, come le organizzazioni (le aziende).

Le **dimensioni della complessità** nella progettazione:

- definiscono uno *spazio di progettazione*: si hanno diversi sistemi a seconda delle loro tipologie;
- forniscono forme di *decomposizione sommaria* dello spazio, cui andranno aggiunte altre scelte da fare;

- interagiscono fra loro: dimensioni combinabili anche se studiate separatamente.

Esse sono riassunte nella seguente tabella:

Dimensione	Valori
<i>Modularità</i>	piatta / modulare / gerarchica
<i>Orizzonte</i> pianificazione	nessuna pianificazione / #tappe finito (miopia) / indefinito / infinito (processo)
<i>Rappresentazione</i> del mondo	stati / feature / individui + relazioni #stati: $n_s / 2^{n_f} / 2^{n_i^2} \cdot n_r$ (#stati, #feat., #ind. e #rel. binarie)
<i>Incertezza</i>	<i>osservabilità</i> del mondo: totale / parziale <i>effetto</i> : deterministico / stocastico
<i>Preferenze</i>	obiettivi (da raggiungere / mantenere) / compromessi (preferenze cardinali / ordinali)
<i>Apprendimento</i>	conoscenza data / appresa
<i>Limiti</i> risorse	razionalità perfetta / limitata
<i>Numero</i> di attori	singolo / multipli agenti (distribuiti, cooperativi / competitivi)
<i>Interattività</i>	<i>ragionamento</i> (computazione) online / offline

## Modularità

**Modularità** grado di decomposizione di un sistema in *moduli* interagenti da prendere in considerazione separatamente:

- serve a *dominare la complessità*, nei sistemi SW ma anche nelle organizzazioni;
- tipicamente espressa come decomposizione gerarchica: ogni modulo è organizzato in sotto-moduli a loro volta organizzati gerarchicamente fino al livello base; astrazione procedurale e OOP servono a sfruttare modularità e astrazione.

Le *strutture* possibili sono le seguenti:

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

Le modalità di *ragionamento* dipendono dal tipo di struttura:

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

## Schema di Rappresentazione

Lo **schema di rappresentazione** riguarda la *descrizione del mondo*. Le alternative possibili sono le seguenti:

- stati;
- caratteristiche (feature) / proposizioni;
- individui e relazioni.

## Stati

Gli **stati** distinti del mondo hanno un impatto sul comportamento del sistema. Essi sono fattorizzabili in stati *interni* (delle credenze) e stati *dell'ambiente*. Nello schema più

semplice il 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 passa nello stato spento.
- 

## Caratteristiche

Il ragionamento si basa su **caratteristiche** (*feature*) degli stati o *proposizioni* booleane in vece della loro enumerazione. Uno stato può essere descritto in termini di *caratteristiche*, 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, ad esempio 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.
- 

## Proposizioni

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

---

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

- immagini b/n, risoluzione  $30 \times 30$ ;
  - *azione*: determinare le lettere tracciate;
  - $2^{900}$  stati dell'immagine quindi  $26^{2900}$  funzioni dalle immagini alle lettere
    - proibitivo rappresentarle tutte in termini di spazio degli stati;
    - meglio definire alcune caratteristiche dell'immagine (es. segmenti) e definire le funzioni in termini di esse.
-

# Relazioni e Individui

Nella descrizione di mondi complessi una relazione su un singolo individuo è una **proprietà**. Si può definire una *caratteristica* per ogni possibile relazione tra gli individui, ad es.  $R(a, b)$ , *Anna studia Informatica*.

---

**Esempio** — domotica (cont.):

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

*Descrizioni relazionali* più convenienti 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;
- si possono considerare intere *classi di individui* senza enumerarne caratteristiche/proposizioni, o addirittura i numerosissimi stati;
- permettono di ragionare su *infiniti* individui, ad esempio l'insieme dei numeri, o l'insieme di tutte le stringhe, cosa impossibile in termini di stati o feature.

## Incertezza

Si progetta anche tenendo conto dell'**incertezza** insita nel dominio considerato:

1. percezione / osservazione
2. effetti delle decisioni / azioni

## Osservazione Incerta

A volte è possibile l'*osservazione diretta* dello stato del mondo, ad es. nei giochi da tavolo, carte.

Più spesso accade che la percezione dello stato sia *difettosa* o *parziale / indiretta*: al più si può disporre di 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 osservazioni. Uno stato può essere:

- *pienamente osservabile*: si può riconoscere dalle osservazioni, assunzione spesso fatta per ragioni di trattabilità dei problemi
- *parzialmente osservabile*: non osservato direttamente; è possibile che più stati portino alle stesse osservazioni oppure le osservazioni sono *rumorose* (*noisy*)

## Effetto Incerto

In certi casi è possibile conoscere sempre l'effetto delle azioni/decisioni: dato uno stato e un'azione/decisione, si può predire *precisamente* lo stato risultante dall'applicazione





- *selezione* dei dati da raccogliere;
- *rappresentazione* dei dati e dei modelli;
- *selezione dei learning bias* appropriati;
- uso della conoscenza appresa per modificare decisioni / azioni dell'agente.

## Limitatezza delle Risorse Computazionali

Le **limitazioni** sulle risorse computazionali disponibili spesso impediscono di prendere le migliori decisioni (sulle azioni da svolgere): non è possibile trovare la migliore decisione in modo sufficientemente rapido dati limiti di memoria, ad es., potrebbe essere inutile attendere 10' per stabilire quale fosse la decisione migliore 10' prima. Si fanno quindi compromessi sulla qualità della soluzione da cercare spesso necessari: spesso è preferibile una soluzione ragionevole ma rapida che una migliore calcolata quando è troppo tardi perché il mondo esterno nel frattempo è cambiato.

Tale dimensione determina se il sistema ragiona per prendere la migliore decisione: senza tener conto dei limiti, *razionalità perfetta*, oppure tenendo conto dei limitati, *razionalità limitata*.

I *limiti* riguardano il *tempo*, la *memoria* e la *precisione* (numerica), il grado di approssimazione.

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

- in qualunque momento produce la *miglior soluzione corrente*;
- si assicura che la qualità non decresca, si conserva la migliore soluzione trovata da restituire su richiesta;
- l'attesa può avere un *costo*: a volte meglio decidere/agire subito anziché di aspettare una soluzione probabilmente migliore.

In caso di razionalità limitata, si deve decidere se aspettare e ragionare un po' più a lungo, ma è difficile giudicare la politica migliore: anche il tempo speso per decidere è da sottrarre a quello di ricerca della soluzione; ciò motiva l'uso di forme di *ragionamento approssimato*.

## 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. nel richiamo degli animali (ad es. cani) e la loro ricompensa immediata, essi di solito non hanno 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 *fasi* successive.

Si possono distinguere sistemi/agenti SW:

- *senza pianificazione*: nelle decisioni (sulle azioni) non considerano il futuro e il fattore-tempo non è preso in considerazione;
- *a orizzonte finito*: interessano solo un numero prefissato di passi, ad es., un dottore che deve curare il paziente per il quale c'è tempo per le analisi, quindi si prevedono due fasi: analisi e cura; nel caso in cui conti una sola fase, si preferisce un agente/algoritmo *greedy* o *miope*;
- *a orizzonte indefinito*: numero di passi finito, ma non predeterminato, ad es., un agente che deve recarsi in un luogo lontano ma non sa quanti passi ci vorranno (ad es. numero di treni da prendere);
- *a orizzonte infinito*: sempre attivo (*processo*), ad es. un modulo di stabilizzazione nei robot che si fermasse una volta raggiunta la stabilità, porterebbe in seguito a frequenti cadute.



# Numero di Agenti

Difficoltà aggiuntive degli ambienti con altri agenti / sistemi:

- occorre ragionare sugli altri agenti secondo *strategie*: potrebbero cercare di confondere e manipolare o potrebbero cooperare; conviene agire in modo casuale se 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 complesso; 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:

- ragionamento da *agente singolo*: gli altri come 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 con *fini/preferenze* che dipendano da quello dalla *comunicazione* con gli altri;
  - più difficile se gli agenti possono agire *simultaneamente* o se l'ambiente è solo *parzialmente osservabile*.

## Interazione delle Dimensioni

Non si possono studiare le dimensioni separatamente perché sono soggette a *interazioni complesse*:

- *Rappresentazione e modularità*: avendo una gerarchia di moduli semplici, il ragionamento può essere svolto su insieme finito di stati; altri livelli di astrazione richiedono il ragionamento su relazioni. Ad esempio per un robot per le consegne, il modulo per il bilanciamento avrà pochi stati, il modulo per la consegna che ragiona sulle priorità deve basarsi su più individui (persone, pacchi, stanze,...) e loro relazioni; a livello più alto il modulo che ragiona sull'intera attività giornaliera, con pochi stati per le diverse fasi del giorno come, ad esempio, impegnato, disponibile e ricarica.
- *Orizzonte e modularità*: ad es. ad alto livello, un cane robotico può ricevere una ricompensa quando risponde al richiamo; quando decide sui singoli movimenti, il momento del premio potrebbe essere lontano quindi l'orizzonte risulta potenzialmente indefinito.
- *Incertezza sull'osservazione e complessità del ragionamento*: molto più facile ragionare quando si conosce lo stato del mondo; l'incertezza su individui e relazioni è più complicata da trattare.
- *Incertezza sugli effetti e modularità*: ad un dato livello della gerarchia, una decisione può essere *deterministica* mentre ad un altro potrebbe essere *stocastica*. Ad esempio, in un volo per Parigi si può adottare un'astrazione decrescente: si sa la città in cui ci si trova (in aeroporto non si conosce bene la posizione), si sa di essere con i piedi per terra (*equilibrio*). 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.
- *Molteplicità e modularità*: agente progettato attraverso più sistemi che interagiscono condividendo il *fine comune* di rendere intelligente il comportamento dell'agente al livello superiore; secondo alcuni [Min86], l'intelligenza è una caratteristica emergente da una "società" di agenti non intelligenti.

- *Apprendimento e rappresentazione*: spesso l'apprendimento lavora sulla rappresentazione di feature: determina modelli che portano a migliori predizioni dei valori di una specifica feature-obiettivo (problemi di *classificazione*, *regressione*); ma si può lavorare anche su individui e relazioni, ovvero l'apprendimento di gerarchie di modelli, in domini parzialmente osservabili e attraverso sistemi multipli. Tutte queste diverse modalità interagiscono con diverse altre dimensioni.
- *Modularità e razionalità limitata* per un ragionamento più efficiente; il formalismo può diventare più complicato ma per costruire sistemi complessi servono: la decomposizione in componenti più piccole e delle approssimazioni per poter decidere in tempi accettabili anche in regime di memoria limitata.

## 3 Progettazione di Sistemi Intelligenti

### 3.1 Semplificazione Ambienti / Sistemi

In generale, la conoscenza del sistema / dell'agente non corrisponde a quella nella mente dell'esperto e/o del progettista.

I casi estremi possibili sono i seguenti:

- *sistema specializzato* nel suo dominio/task, poco utile fuori del suo contesto, come ad es. un termostato, un robot-vernicatura;
- *sistema flessibile* che si adatta ai contesti e accetta nuovi task (a runtime), come ad es. degli insetti.

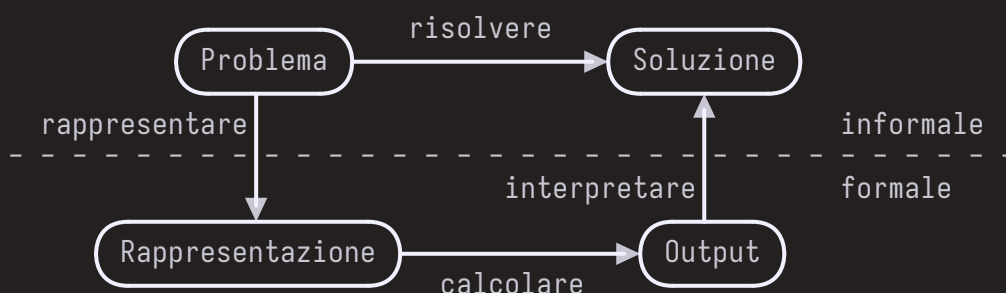
Le strategie di costruzione alternative sono:

- un modello dell'ambiente/compito *semplificato* utile a costruire sistemi di ragionamento *complessi*; ciò facilita la dimostrazione di proprietà dei sistemi e la loro ottimizzazione
- sistemi *semplici* per contesti *complessi* (naturali) come ad esempio le auto a guida autonoma: dotate di *sensori* sofisticati ma poi sono poche le azioni a disposizione.

### 3.2 Compiti / Problemi

In AI importante rappresentare *COSA* vada fatto / calcolato (ad es. la diagnosi per un paziente, la localizzazione di una risorsa) e non *COME* (ragionamento) ridotto a ricerca nello spazio di soluzioni possibili.

Una descrizione di un **compito** (*problema*) dev'essere disponibile a livello informale. Lo *schema generale* di risoluzione computazionale prevede la sua *formalizzazione*.



### 3.3 Rappresentazione Interna

La **conoscenza** può essere definita come informazione *stabile, a lungo termine* su un dominio, a differenza delle cosiddette **credenze** (*belief*) sull'ambiente che sono più *transitorie*. Per poterci ragionare essa va rappresentata (nelle macchine) formalizzandola in termini di un **linguaggio di rappresentazione**. Una **base di conoscenza** risulterà come rappresentazione interna al sistema, codificata attraverso idonee strutture dati.

Le *proprietà* principali degli schemi di rappresentazione sono le seguenti:

- *ricchezza espressiva* sufficiente alla risoluzione del problema;
- *vicinanza* ai termini naturali del problema:
  - compatta, naturale, manutenibile;
  - relazione fra dominio e sua rappresentazione → verifiche di correttezza;
    - piccole modifiche al problema → piccoli cambi di rappresentazione
  - connessa alla *spiegabilità* dei meccanismi di ragionamento interni (XAI)
- *trattabilità*, propensione a un'elaborazione efficiente:
  - caratteristiche del problema sfruttabili per risparmiare risorse
  - compromessi con l'accuratezza
- *acquisibilità* dagli utenti, da dati e/o da esperienza pregressa.

### 3.4 Ipotesi di Newell & Simon

Si suppone che un **sistema di simboli fisico** sia incorporato nella mente umana o nelle macchine. Un **simbolo** è un *pattern* significativo manipolabile come *singola unità*, può essere un oggetto fisico, parte del mondo reale, come ad es. una parola scritta, una frase, un gesto, un segno sulla carta o una sequenza di bit. Un **sistema di simboli** serve a elaborare simboli, con azioni tipiche come la creazione, la copia, la modifica e la cancellazione.

Un'ipotesi (empirica) forte avanzata è la seguente:

*“A physical symbol system has the necessary and sufficient means for general intelligent action” — Newell & Simon [NS76]*

Ogni agente intelligente sarebbe necessariamente un sistema di simboli fisico contenente tutto quanto serve per un comportamento *intelligente*. Questo è compatibile con l'uso di un *corpo* per percepire/agire il/sul mondo (come per gli agenti robotici).

Un sistema intelligente per ragionare manipola simboli. I simboli *interni* si riferiscono ai vari oggetti del mondo *esterno* e rimandano a concetti *utili* ai compiti prefissati.

Controversia: nei livelli intermedi, pattern non interpretabili = simboli?

### 3.5 Modelli: Livelli di Astrazione

Un **modello** è una rappresentazione del mondo che può essere statica, ciò che si crede essere *vero*, ovvero dinamica, il suo *funzionamento*. È utile lavorare a vari livelli di dettaglio, ossia di **astrazione**: si rappresenta la parte del mondo di interesse trascurando i dettagli inutili. Il **livello di astrazione** (diversa precisione nel dettaglio) determina un *ordine parziale* (maggiori dettagli a basso livello). Sono spesso ammissibili *più modelli* a

diversi livelli di astrazione, anche in *contraddizione* fra loro: andranno giudicati più in base alla loro *utilità* che per la loro correttezza.

---

**Esempio** — Per un robot, modello della pianta d'un edificio con posizioni relative, *ignorando* distanze, grandezze, angoli di sterzata, ecc., caratteristiche da considerare, invece, in una rappresentazione di basso livello.

---

## Scelta del Livello d'Astrazione

Le descrizioni di *alto livello* risultano più semplici per gli umani, sia nella loro specifica sia nella loro comprensione. D'altro canto descrizioni di *basso livello* sono più accurate e predittive: i dettagli essenziali per la soluzione del problema. Più basso è il livello, più complesso sarà il tipo di ragionamento richiesto comportando più passi e più piani d'azione da scegliere. Tuttavia tutte le informazioni di basso livello da considerare sono potenzialmente sconosciute.

Livelli d'astrazione comuni fra sistemi *biologici* e *computazionali* sono i seguenti:

- **livello della conoscenza** sul mondo esterno: considera quello che il sistema assume di sapere e i suoi obiettivi, ma non come ragiona, ossia non come calcolare la soluzione o quale strategia adottare;
  - **livello simbolico** interno (attestato da esperimenti dei cognitivisti): descrizione del modo di ragionare del sistema utile a implementare il livello precedente, manipolando simboli per produrre risposte.
- 

**Esempi** — Modelli a più livelli d'astrazione

*Sistemi biologici:*

- livello *neurale*
  - livello *biochimico*
    - livello *chimico*
    - livello *fisico*

*Sistemi SW* in Informatica

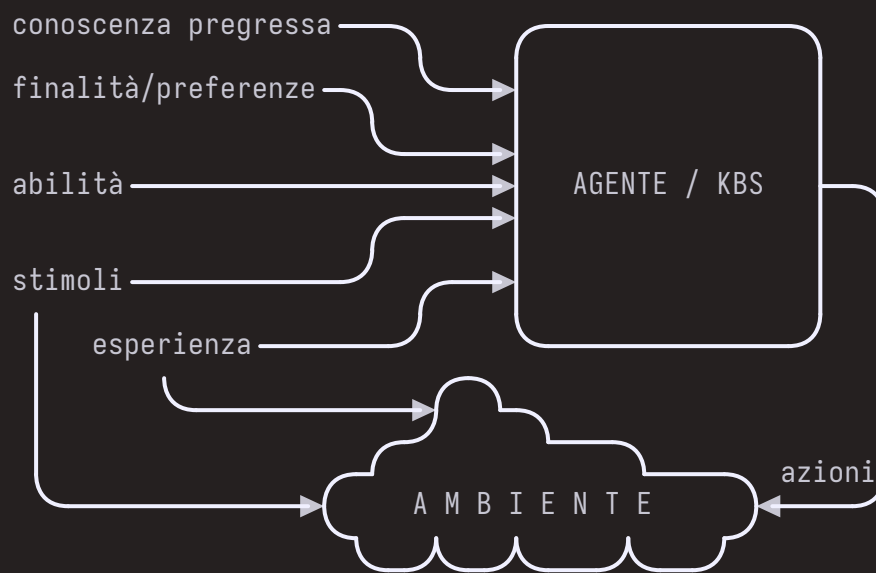
La stessa *Scienza* è strutturata in livelli gerarchici

---

## 4 Sistemi Intelligenti Basati su Conoscenza

---

Un **agente intelligente** è un modello capace di percezione—ragionamento—azione che si trova immerso in un **ambiente** (il suo **mondo**). Un possibile schema è il seguente:



Il comportamento dell'agente/del KBS si basa su:

- **conoscenza pregressa** su agente e ambiente;
- **storia** dell'interazione con l'ambiente: **stimoli** (ad es. *osservazioni* sull'ambiente, azioni dell'ambiente sull'agente) ed **esperienza passata**, i.e. dati da cui imparare;
- **obiettivi** da raggiungere o **preferenze** sugli stati del mondo;
- **abilità**, azioni *primitive* di cui è capace.

Nella scatola nera si rappresenta un suo *stato interno delle credenze*, o **belief state**, comprendente la rappresentazione delle cose che si ritengono vere circa l'ambiente, cosa si è imparato, obiettivi intermedi presenti e futuri. Esso è aggiornato in base agli stimoli e serve a prendere decisioni (sulle prossime azioni).

---

## Esempi

### *Robot* (HW/SW)

- unità di calcolo
- sensori
- attuatori

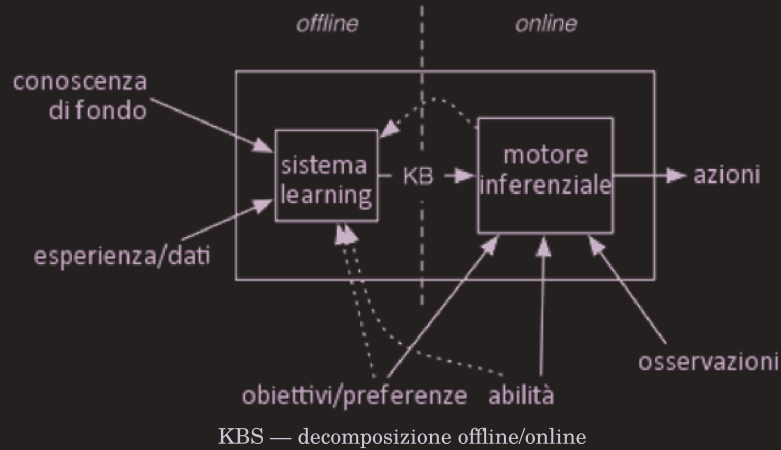
### *Sistema Esperto* (puramente computazionale, SW)

- basato su informazioni fornite da esperti di dominio codificate dai progettisti (tipicamente come sistemi a regole di produzione)
- finalità:
  - diagnostica
  - infobot

---

Fasi dell'elaborazione: il funzionamento di un KBS/agente intelligente può essere schematizzato in 2 fasi:

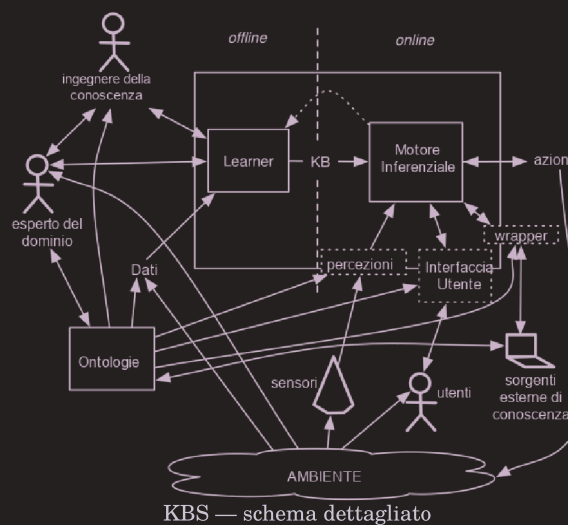




La *KB* viene coinvolta in momenti diversi:

- *Offline*, viene costruita/integrata (apprendimento) usando conoscenza pregressa ed esperienze passate (e.g. dati): spesso servono molti dati e conoscenza generale (a livello statistico). Tale attività non è automatizzata nei *sistemi esperti*.
- *Online*, si prendono decisioni / compiono azioni e si aggiorna la KB; nel ragionamento si usano KB, osservazioni, obiettivi e abilità del sistema.

Una KB come *long-term memory* è utile nel futuro: viene appresa dai dati e dall'esperienza pregressa; un belief state, *short-term memory*, è un modello dell'ambiente attuale necessario tra successivi intervalli di tempo.



## Applicazioni Prototipiche

### Domini Applicativi

Tipici domini applicativi:

- *robot per la consegna* all'interno di un edificio
  - accetta richieste, individua percorsi e alloca le risorse
  - decide le priorità e agisce evitando di danneggiare se stesso e gli altri
- *assistente per la diagnostica*
  - aiuta a capire i problemi e suggerisce rimedi

- *sistema di tutoring* per studenti
  - interagisce fornendo informazioni su un argomento di interesse, assegnando compiti/test e valutando le capacità
  - deve comprendere la materia, lo studente e come questo impari
- *assistente agli acquisti* di beni e servizi per conto dell'utente
  - sa riconoscere richieste e preferenze
  - sa fare compromessi tra obiettivi distinti

## Infobot

Un **infobot** è una specie di *robot*, che interagisce con un ambiente informativo anziché fisico. I suoi compiti tipici sono:

- estrarre informazioni da una rete di sorgenti informative (Internet | enciclopedie multimediali, ecc.);
- determinare quali informazioni servano per rispondere a 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 precedente):

- *conoscenza pregressa* significato delle parole, tipi di sorgenti informative e come accedervi;
- *esperienza passata* informazione da ottenere, velocità dei vari server e informazioni sulle preferenze-utente;
- *finalità* informazione da ricercare e compromessi sul dispendio di risorse e tra volume e qualità;
- *osservazioni* informazioni presenti sui siti al momento, link disponibili e carico sulle varie connessioni;
- *abilità* azioni primitive di cui è capace.

**Output:** informazioni utili alla comprensione da parte dell'utente, anche in caso di mancanza di parte della conoscenza.

L'infobot deve essere capace di:

- derivare informazioni implicite nella/e base/i di conoscenza (e interagire in linguaggio naturale);
- cercare informazioni rilevanti 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 informazioni non siano 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;
- trovare compromessi tra sorgenti a basso costo ma poco affidabili e sorgenti più costose ma più complete;
- imparare quale conoscenza sia disponibile e dove e le informazioni che l'utente richiede.

Due diversi infobot:

- **unibot** interagisce con un DB: informazioni su corsi, tempistica, propedeuticità, regole, valutazioni

- **webbot**: interagisce con il Web, ricerca informazioni utili all'utente

Altra caratteristica importante è la loro *proattività*.

## Riferimenti Bibliografici

---

- [BL22] R.J. Brachman & H.J. Levesque (2022): *Toward a new science of common sense*. In Proc. of the 36th AAAI Conference on Artificial Intelligence, AAAI-22 (DOI)
- [Min86] Minsky, M.: *The Society of Mind*. Simon and Schuster. 1986
- [NS76] Newell, A. and Simon, H.A.: *Computer science as empirical enquiry: Symbols and search*. Communications of the ACM, 19: 113-126. 1976
- [PM23] D. Poole, A. Mackworth: *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press. 3a ed. 2023 (Ch.5)
- [PMG86] D. Poole, A. Mackworth, R. Goebel: *Computational Intelligence: A Logical Approach*. Oxford University Press. 1986
- [Row07] J. Rowley: *The wisdom hierarchy: representations of the DIKW hierarchy*. Information and Communication Science. 33(2): 163-180. 2007 (DOI)
- [RN20] S.J. Russell, P. Norvig: *Artificial Intelligence*. Pearson. 4th Ed. 2020
- [Shi21] Z. Shi: *Intelligence Science - Leading the Age of Intelligence*. Elsevier. 2021 (DOI)

## Link

- [KE] *Knowledge Engineering* in ScienceDirect; in Wikipedia
- [DIKW] *Piramide DIKW* in Wikipedia
- [KBS] *Knowledge-based Systems* in Wikipedia
- [Prolog] linguaggio di programmazione, ad es. cfr. GNU Prolog, SWI Prolog online shell


Dispense ad esclusivo uso interno al corso.  
formatted by [Markdeep 1.17](#) 

Figure tratte dal libro di testo [PM23], salvo diversa indicazione.