

# **INGEGNERIA DEI SISTEMI BASATI SU CONOSCENZA: INTRODUZIONE**

Nicola Fanizzi

**Ingegneria della Conoscenza**

CdL in Informatica • *Dipartimento di Informatica*

Università degli studi di Bari Aldo Moro

**Introduzione**

Definizione

Sistemi Intelligenti

AI fra Scienza e Ingegneria

**Progettazione di Sistemi Basati su Conoscenza**

Computazione

Compiti / Problemi

Soluzioni

Classi di Soluzioni..

Rappresentazioni

Ipotesi di Newell & Simon

Modelli del Mondo: Livelli di Astrazione  
Livelli

**Dimensioni della Complessità**

**Sistemi Intelligenti Basati su Conoscenza**

KBS come Agenti Intelligenti

Conoscenza e Ragionamento / Azione

Fase di Progetto e Offline

Ontologie

Attività Online

**Appendice: Dimensioni della Complessità**

# **INTRODUZIONE**

## Ingegneria della Conoscenza – possibili definizioni:

*“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.” [7]*

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

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

### **Piramide DIKW** (*gerarchia della conoscenza*)

[8, DIKW]

classe di modelli atti a rappresentare relazioni strutturali e/o funzionali sottese tra *dati*, *informazioni*, *conoscenza* e *sapere*



- rappresentazioni alternative anche come *concatenazione* (o come *framework*, con una serie di *grafi* e come *continuum*)

**Dati:** *simboli / segni* che rappresentano stimoli o segnali

- *fatti, segnali, simboli*

**Informazione:** dati dotati di *significato / scopo*

- *inferita* dai dati,
- *strutturale vs. funzionale*
- *simbolica vs. soggettiva*

**Conoscenza:** informazione elaborata / organizzata / applicata

- *elaborata*
- *proposizionale*
- *procedurale*

Ambito:

**Intelligenza Artificiale (AI):** mira a studiare e comprendere i principi che rendono possibile un comportamento intelligente in sistemi artificiali

Ipotesi:

*ragionamento  $\approx$  computazione*

- collegata alla tesi di *Church-Turing*:
  - *livello di astrazione* nel quale: ragionamento = manipolazione di simboli
    - azioni/decisioni di un sistema spiegate in termini dei suoi input
  - *artificiale* come *computazionale*
    - progettazione di *artefatti SW intelligenti*, utili a scopi precisi

# INTELLIGENZA

In generale:

Intelligenza dei Sistemi  $\neq$  Intelligenza Umana

- visione *olistica* vs visione *riduzionistica*
  - e.g. anche una *comunità organizzata* può esibire un *comportamento intelligente*
    - *ant colony, swarm intelligence* (algoritmi correlati)
      - ricerca del cibo, adattamento ai cambiamenti



*“È possibile creare macchine in grado di pensare?”*

domande  
analoghe

*“È possibile creare macchine in grado di volare?”*

## Approccio *Scientifico* e Prospettiva *Tecnologica*

- Al come *scienza*: tesa a comprendere i principi del ragionamento
  - ideare *teorie confutabili* (metodo scientifico) sulla soluzione algoritmica di problemi
    - supportate da *implementazioni* per l'essenziale *verifica sperimentale*
    - *Rasoio di Occam*: preferire sempre teorie e implementazioni più semplici
- Al come disciplina *ingegneristica*: mira a costruire tecnologie per risolvere specifici problemi
  - creare *sistemi* SW intelligenti *basati su conoscenza* (KBS) *testabili*
    - qualità valutabili attraverso tecniche standard dell'informatica

# **PROGETTAZIONE DI SISTEMI BASATI SU CONOSCENZA**

Momenti da distinguere:

1. elaborazione *in fase di progetto*
  - a cura del progettista
2. computazione *offline*: precedente all'osservazione del mondo
  - compilazione / apprendimento di conoscenza
  - basato su *conoscenza di fondo* (BK) data o definita in precedenza
  - per ricavare una *base di conoscenza* (KB) in forma utilizzabile nel seguito
3. computazione *online*: dall'osservazione alla decisione/azione
  - ottiene informazioni dalle *osservazioni*
  - prende decisioni (su azioni da svolgere) usando la sua KB

conoscenza nella mente del progettista

≠

conoscenza nella KB del sistema

casi estremi:

- *sistema specializzato* nel suo dominio/task  
ma inutile al di fuori di questo contesto
- *sistema flessibile* che si adatta ai contesti

# STRATEGIE DI COSTRUZIONE

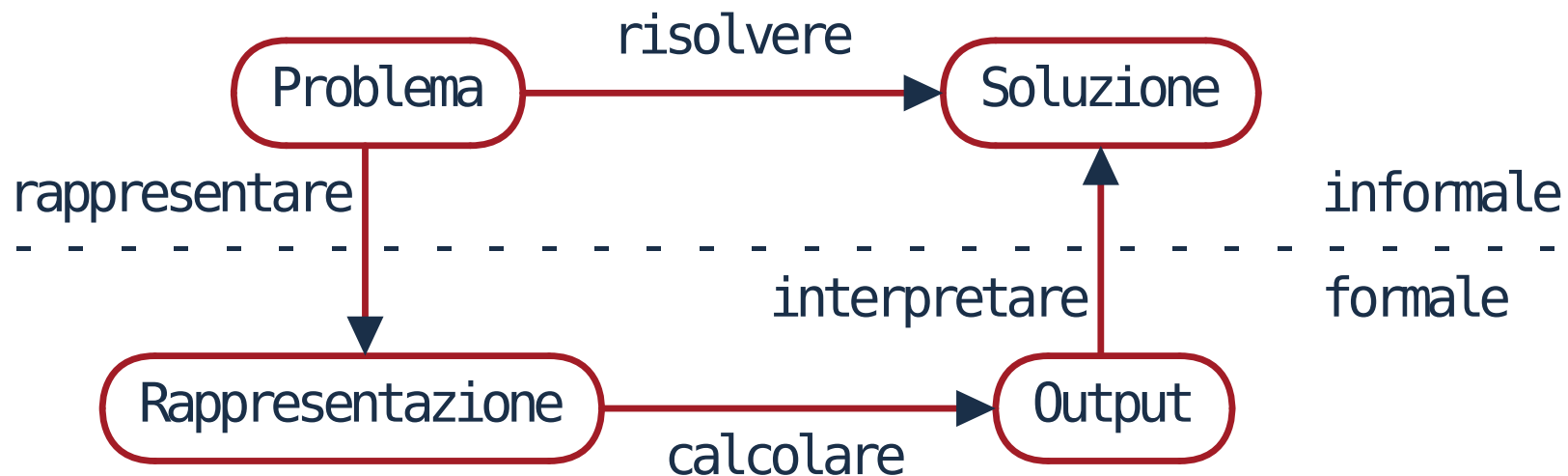
alternative:

- modello dell'ambiente/compito *semplificato*  
utile a costruire sistemi di ragionamento *complessi*
  - aiuta nella dimostrazione di proprietà o nell'ottimizzazione
- costruire sistemi/agenti *semplici* per contesti *complessi* (naturali)
  - es. ispirandosi a comportamenti semplici come quello degli insetti
  - auto a guida autonoma

In AI importante rappresentare **COSA** vada calcolato

- **COME** (ragionamento): ricerca in uno spazio di soluzioni possibili

Descrizioni di **compiti** (*problemi*) disponibili solo a livello informale



**Schema generale** di risoluzione computazionale ← formalizzazione

# RAPPRESENTAZIONE

*Conoscenza* — Informazione sul dominio utile a risolvere problemi

Nelle macchine:

- formalizzata in termini di un **linguaggio di rappresentazione**
  - **base di conoscenza**: rappresentazione interna al sistema
    - codificata attraverso idonee strutture dati



## PROPRIETÀ DEGLI SCHEMI DI RAPPRESENTAZIONE

- **ricchezza espressiva** sufficiente alla risoluzione del problema
- **vicinanza** ai termini del problema
  - compatta, naturale, manutenibile:
  - relazione fra dominio e sua rappresentazione → verifiche di correttezza
    - piccole modifiche al problema → piccoli cambi di rappresentazione
- **efficienza** nell'elaborazione (trattabilità)
  - caratteristiche del problema sfruttabili per risparmiare risorse
    - compromessi con l'accuratezza
- **acquisibilità** dagli utenti, da dati e/o da esperienza pregressa

## LINGUAGGI DI RAPPRESENTAZIONE

Inizialmente pensati per un solo obiettivo  
man mano estesi per coprirne altri:

- linguaggi per l'*apprendimento*  
poi estesi per ammettere maggiori capacità risolutive o di inferenza
- linguaggi *espressivi* (rappresentazione)  
estesi in seguito per aggiungere capacità di inferenza e apprendimento
- linguaggi semplici ai fini della *trattabilità* del calcolo  
poi arricchiti e resi più naturali per facilitare l'acquisizione di conoscenza

Come nell'ingegneria del SW,  
il progettista deve definire cosa costituisca una **soluzione**:

- spesso come *raffinamento della specifica* del problema ma con *lacune* non colmabili arbitrariamente
- forme di *senso comune* da automatizzare per trarre conclusioni riguardanti *assunzioni non dichiarate*
- spesso diverse soluzioni possibili
  - dato un problema ben definito, quanto conta che la risposta sia errata o incompleta?
    - a volte serve la *migliore* soluzione rispetto a un dato criterio

## CLASSI DI SOLUZIONI..

### Soluzione Ottimale

migliore secondo una *misura di qualità*:

- *ordinale*, tipica
- *cardinale*, se contano anche le grandezze relative:
  - ad es. nella combinazione di criteri multipli o nel ragionamento in presenza di incertezza
  - ad es. grado di *utilità*

### Soluzione Soddisfacente

buona secondo una specifica delle risposte *adeguate*:

- quando non serve la migliore perché costosa

## ..CLASSI DI SOLUZIONI

### Soluzione Approssimata

soluzione di qualità *prossima* a quella ottimale:

- in base a una *misura cardinale*
- conveniente per ragioni di efficienza anche in virtù degli algoritmi adottati

### Soluzione Probabile

*verosimilmente* una soluzione,  
con un determinato *grado* di certezza:

- forma di approssimazione precisa di soluzione soddisfacente
- si possono distinguere i tassi di errore per:
  - *falsi-positivi* (risposte fornite non corrette)
  - *falsi-negativi* (risposte corrette non fornite)
    - categorie non esclusive

## *Problematiche*

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

## **Sistemi simbolici fisici** incorporati nella mente umana / nelle macchine

- **Simbolo** — *pattern* significativo manipolabile come *singola unità*
  - parole scritte, frasi, gesti, segni su carta o sequenze di bit
- **Sistema di simboli** serve a creare, copiare, modificare e distruggere simboli

## IPOTESI DI NEWELL & SIMON

*“A physical symbol system has the necessary and sufficient means for general intelligent action” – Newell & Simon [5]*

- ipotesi (empirica) forte:
  - ogni sistema intelligente è necessariamente un sistema simbolico fisico
  - è tutto quanto serve per un comportamento *intelligente*
  - compatibile con l'uso di un *corpo* per percepire/agire il/sul mondo

Un sistema intelligente per ragionare manipola simboli:

- simboli *interni* che si riferiscono ai vari oggetti del mondo *esterno*
  - rimandano a concetti *utili* ai compiti prefissati

# MODELLI DEL MONDO: LIVELLI DI ASTRAZIONE

Rappresentazione del mondo:

(statica) ciò che va considerato *vero* o (dinamica) il suo *funzionamento*:

- utile anche se non troppo dettagliata → **astrazione**
  - rappresentare solo parte del mondo (trascurando i dettagli)
- **livello di astrazione** (diversa precisione nel dettaglio): *ordine parziale*
  - basso livello → maggiori dettagli
- ammissibili *più modelli* a diversi livelli, anche in *contraddizione* fra loro
  - si giudicano più in base all'*utilità* che per la loro correttezza

**Esempio** — Per un robot, modello della pianta d'un edificio con pos. relative, *ignorando* distanze, grandezze, angoli di sterzata, ecc.

- caratteristiche da considerare in una rappresentazione di basso livello



## MODELLO A SINGOLO LIVELLO D'ASTRAZIONE

Descrizioni di *alto livello* più semplici per gli umani

- specifica + comprensione

Descrizioni di *basso livello* più accurate e predittive

- dettagli essenziali per la soluzione del problema

Più basso il livello, più complesso il ragionamento

- comporta più passi e più piani d'azione da scegliere
- potenzialmente sconosciute le info di basso livello da considerare

# MODELLO A PIÙ LIVELLI D'ASTRAZIONE

Ad esempio, sistemi biologici / informatici

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

La *Scienza* stessa è strutturata in livelli gerarchici

## LIVELLI

Comuni a sistemi *biologici* e *computazionali*

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

# **DIMENSIONI DELLA COMPLESSITÀ**

Variabile: finalità e contesti / ambienti diversi

- dai singoli termostati alle organizzazioni (aziendali)

**Dimensioni** della complessità nella progettazione:

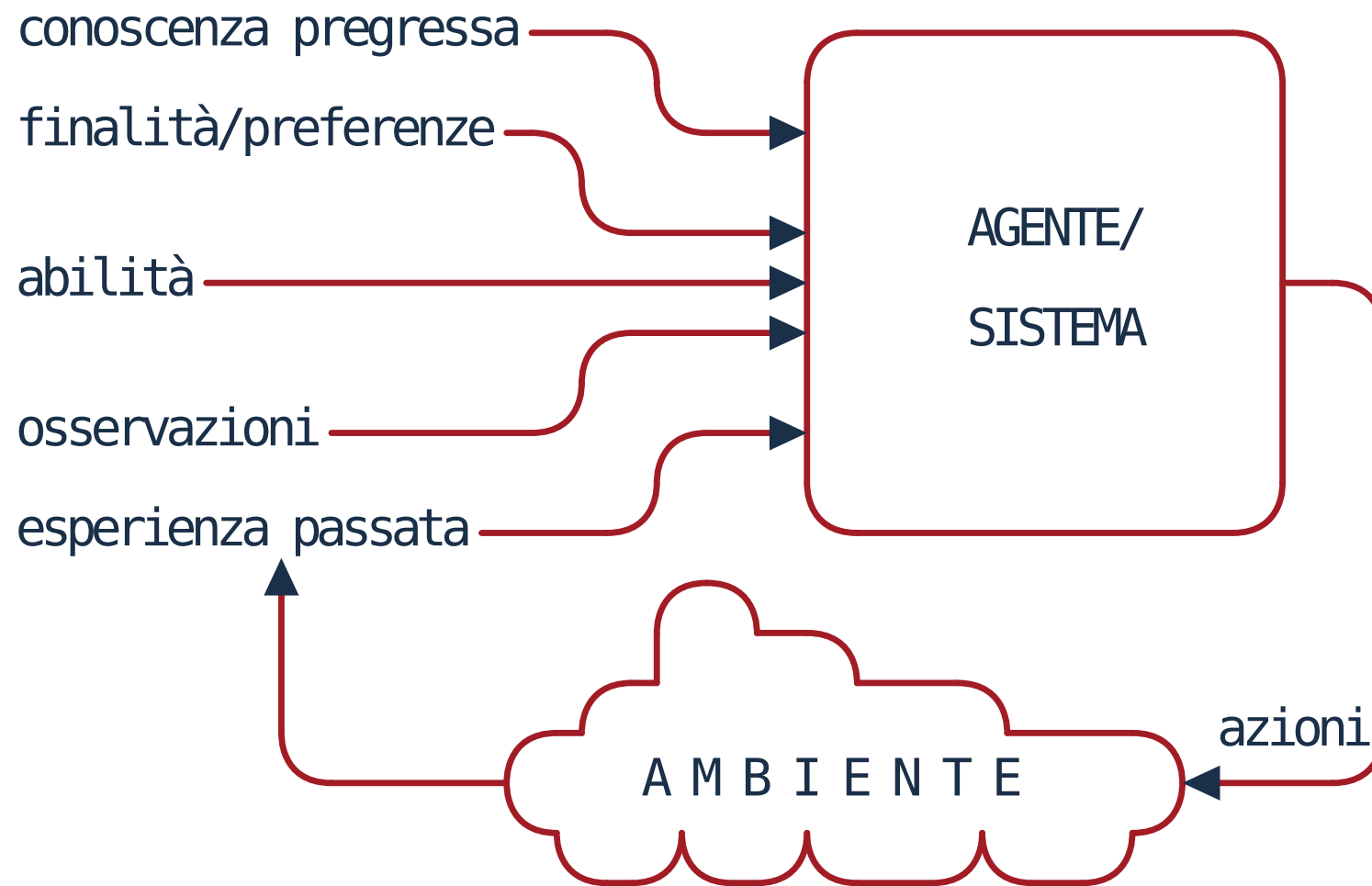
- definiscono uno *spazio di progettazione*
  - diversi sistemi a seconda delle loro tipologie
- forniscono una *decomposizione sommaria* dello spazio
  - ma vanno aggiunte altre scelte
- interazioni: combinabili anche se studiate separatamente

Dimensione	Valori
Modularità	piatta, modulare, gerarchica
Schema di rappresentazione	stati / feature / relazioni e individui stati: $n_s$ / $2^{n_f}$ / $2^{n_i^2} \cdot n_r$ (rel. binarie)
Incertezza sull'osservazione	totalmente / parzialmente osservabile
Incertezza sull'effetto	deterministico, stocastico
Preferenze	finalità, preferenze complesse
Apprendimento	conoscenza data, conoscenza appresa
Limiti alle risorse computazionali	razionalità perfetta, razionalità limitata
Orizzonte	senza pianificazione, a stage finite, indefinite, infinite
Numero di attori	singolo sistema/agente, sistemi/agenti multipli (distribuiti)

Approfondimenti in [appendice](#)

# **SISTEMI INTELLIGENTI BASATI SU CONOSCENZA**

**Agente Intelligente:** percezione—ragionamento—azione  
immerso in un **ambiente** (il suo mondo)





## Comportamento in base a:

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

## Nella scatola nera *stato interno delle credenze*, o **belief state**:

- rappresentazione delle cose che si ritengono vere circa l'ambiente
- cosa si è imparato, obiettivi intermedi presenti e futuri
- aggiornato in base agli stimoli
- serve a prendere decisioni (sulle prossime azioni)

# Esempi

***robot*** unità di calcolo+sensori+attuatori

***sistema esperto*** puramente computazionale (SW), info fornita da umani:

- diagnostica
- infobot

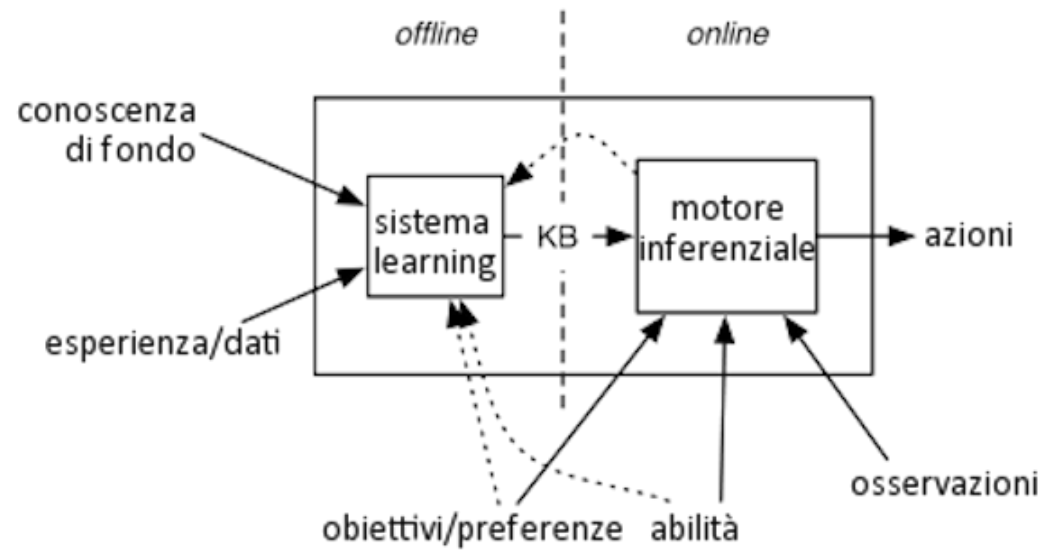
## Sistema Basato sulla Conoscenza (KBS)

modello del dominio atto a risolvere problemi specifici

**Conoscenza** informazioni su un dominio usabili per decidere / agire su di esso, ma anche credenze (belief) sullo stato del mondo:

- info generali e *persistenti* considerate come vere a lungo termine + credenze *transienti*, destinate a essere cambiate più frequentemente
- entrambe misure del credito
- in AI, conoscenza giustificata dalla sua *utilità* (più che dalla verità)

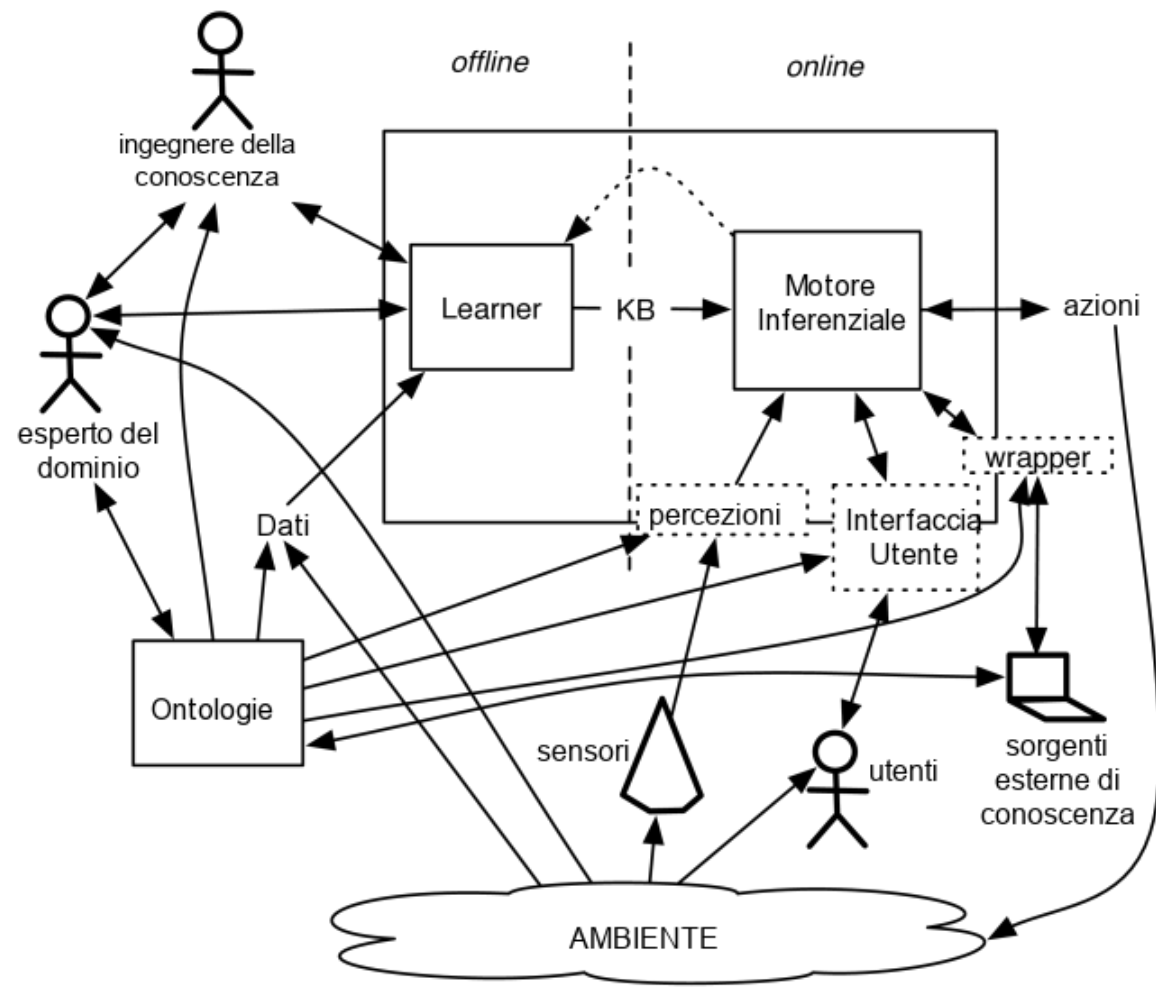
# Fasi dell'elaborazione



KBS — decomposizione offline/online

## **KB** coinvolta in momenti diversi:

- **Offline**, si costruisce (apprendimento) la base usando
  - conoscenza pregressa ed esperienze passate (e.g. dati)
    - spesso servono molti dati e conoscenza generale (a livello statistico)
  - passo non automatizzato nei **sistemi esperti**
- **Online**, si prendono decisioni / compiono azioni e si aggiorna la KB stessa
  - nel ragionamento si usano KB, osservazioni, obiettivi e abilità
    - KB, **long-term memory**
      - utile nel futuro, appresa dai dati e dall'esperienza pregressa
    - belief state, **short-term memory**
      - modello dell'ambiente attuale necessario tra successivi intervalli di tempo



dettaglio

## FASE DI PROGETTO E OFFLINE

Fondamentale rappresentare il *dominio* del problema da risolvere

- *tipi di cose* che compongono il dominio
- *legami* che le mettono in relazione

**Ontologia** — Specifica del significato dei simboli usati dal sistema

- le *cose che esistono* e il *vocabolario* usato per descriverle
- con rappresentazione esplicita in termini di stati completamente osservabili:  
mapping tra stato e mondo
  - definisce le feature degli individui e delle relazioni
    - la macchina sa ragionare manipolando simboli privi di significato, ma per interagire col mondo, deve sapere come tali simboli siano correlati
    - ad es., cosa succede nello stato 57
      - altrimenti numero senza significato

# ONTOLOGIE

Costruite da comunità, indipendentemente da particolari KB | applicazioni:

- vocabolario *condiviso* che consente comunicazione e interoperabilità dei dati fra diverse sorgenti (sensori, umani, DB)
- logicamente viene prima dei dati e della conoscenza pregressa
  - altrimenti un utente non saprebbe cosa fornire in input
- specifica i livelli d'astrazione: se cambia, devono cambiare i dati
  - ad es., ontologia ostacoli per robot (e.g., qualsiasi oggetto fisico)
    - espandendo l'ontologia per differenziare persone, sedie, tavoli ecc., si richiedono diversi dati sul mondo



## Altre attività *offline*

- ***Costruzione*** della KB combinando conoscenza di esperti e dati:  
l'**ingegnere della conoscenza** interagisce con un **esperto di dominio**
  - l'ingegnere s'intende di sistemi, non necessariamente del dominio
  - l'esperto conosce il dominio, non necessariamente i sistemi
- ***Combinazione*** di conoscenza di dominio e dati
- ***Test e debugging***

## **ATTIVITÀ ONLINE**

**Rendendosi disponibili info sulla particolare situazione corrente un sistema può ragionare / prendere decisioni / agire:**

- **osservazioni del dominio e spesso anche preferenze e obiettivi**
  - osservazioni da sensori, utenti e altre sorgenti (es. web)
  - non intervengono esperti o ing. della conoscenza

**Esempio — diagnosi medica:**

- **online solo lavoro sui dettagli su un dato paziente**
- **offline, acquisizione di conoscenza su interazione tra malattie e sintomi, debugging, compilazione**

## Attori/ruoli coinvolti:

- **utenti:** hanno l'esperienza necessaria a fornire info su specifiche situazioni
  - non esperti di dominio,  
non conoscono cosa sia effettivamente necessario al sistema
  - serve un'*interfaccia* adeguata
- **sensori** che forniscono info sull'ambiente
  - *passivi*: feed in continuo
    - es. termometri, telecamere, microfoni, ...
  - *attivi* controllati o interrogati
    - es. test clinici / telecamera che sa dire se una persona sia presente

## ..Ruoli coinvolti:

- **sorgenti esterne** di conoscenza interrogabili riguardo un dominio limitato (es. richiesta di una temperatura a un sito sul meteo, un DB)
  - vari protocolli e compromessi in termini di efficienza
  - **wrapper**: interfaccia tra sistema e sorgente esterna tramite che opera la traduzione tra diverse rappresentazioni
    - spesso progettati per trasmettere una stessa query a più sorgenti
      - es. info sui collegamenti operati da diverse compagnie aeree
    - aderendo a una ontologia comune i simboli condividono il significato:  
**interoperabilità semantica**

# RIFERIMENTI

- [1] D. Poole, A. Mackworth: *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press. 2nd Ed. [Ch.1, 2.4]
- [2] D. Poole, A. Mackworth, R. Goebel: *Computational Intelligence: A Logical Approach*. Oxford University Press
- [3] S. J. Russell, P. Norvig: *Artificial Intelligence* Pearson. 3rd Ed. - cfr. anche ed. Italiana [cap.1-2]
- [4] J. Sowa: *Knowledge Representation: Logical, Philosophical, and Computational Foundations* Brooks Cole/Cengage
- [5] Newell, A. and Simon, H.A. (1976). *Computer science as empirical enquiry: Symbols and search*. Communications of the ACM, 19: 113–126.
- [6] Minsky, M. (1986). *The Society of Mind*. Simon and Schuster
- [7] Shi, Z (2021). *Intelligence Science - Leading the Age of Intelligence*. Elsevier ([DOI](#))
- [8] Rowley, J. (2007). *The wisdom hierarchy: representations of the DIKW hierarchy*. Information and Communication Science. 33(2): 163-180

## 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

# **APPENDICE: DIMENSIONI DELLA COMPLESSITÀ**

**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 organizzato in sotto-moduli a loro volta organizzati gerarchicamente fino al livello base
    - astrazione procedurale e OOP servono a sfruttare modularità e astrazione



## *Strutture possibili*

**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

# 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
    - ad es. organizzazione di una vacanza

**Schema di Rappresentazione** — riguarda la *descrizione del mondo* alternative:

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

## STATI

**Stati** distinti del mondo hanno un impatto sul comportamento del sistema

- fattorizzabili in stati *interni* (delle credenze) e stati *dell'ambiente*

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

Ragionamento in termini di **caratteristiche** (*feature*) degli stati o *proposizioni* booleane in vece della loro enumerazione

- stato descrivibile 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
    - 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

# PROPOSIZIONI

- 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*
    - sono state comprese *regolarità* importanti sul dominio



## Esempio — sistema per il riconoscimento delle lettere:

- immagini b/n, ris.  $30 \times 30$
- **azione**: determinare le lettere tracciate
- $2^{900}$  stati dell'immagine quindi  
 $26^{2^{900}}$  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
    - 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
- per ragionare su *infiniti* individui
  - ad es. l'insieme dei numeri, o l'insieme di tutte le stringhe
  - impossibile in termini di stati o feature

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

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

## OSSERVAZIONE INCERTA

A volte 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 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

**Incertezza sulla percezione** riguarda la possibilità di determinare lo stato del mondo attraverso osservazioni:

- stato ***pienamente osservabile***: si può conoscere dalle osservazioni
  - assunzione spesso fatta per ragioni di trattabilità dei problemi
- stato ***parzialmente osservabile***: non osservato direttamente
  - 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 dell'azione/decisione
  - 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

**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



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), l'evenienza di dover giustificare le decisioni in caso di giudizio legale, ...

## Le preferenze si caratterizzano come

- **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 non deve mettere in disordine la stanza o far male agli altri
- **preferenze complesse**: compromessi sul vantaggio derivate dei vari risultati, eventualmente 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)

Non sempre il progettista dispone di un buon modello del sistema e del suo ambiente:

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

La dimensione dell'**apprendimento** determina se

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

***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
- uso della conoscenza appresa per modificare decisioni / azioni dell'agente

**I limiti sulle risorse computazionali spesso impediscono di prendere le migliori decisioni (sulle azioni da svolgere)**

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

Tale dimensione determina se il sistema ragiona per prendere la migliore decisione

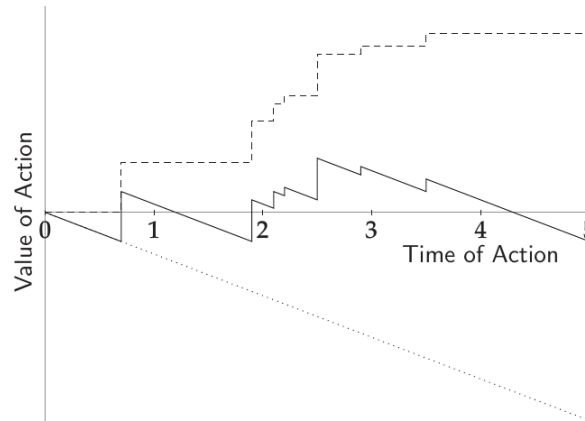
- senza tener conto dei limiti: *razionalità perfetta*
- tenendo conto dei limitati: *razionalità limitata*

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*
- 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



Qualità di una soluzione in funzione del tempo per un algoritmo *anytime*

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 per decidere è da sottrarre a quello di ricerca della soluzione
- motiva il cosiddetto *ragionamento approssimato*



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 *fasi* successive

## Si possono avere sistemi/agenti SW

- **senza pianificazione**: nelle decisioni (sulle azioni) non considerano il futuro
  - fattore-tempo non coinvolto
- **a orizzonte finito**: interessano solo un numero prefissato di passi
  - ad es., dottore che deve curare il paziente ma c'è tempo per le analisi, quindi due fasi: analisi e cura
  - nel caso in cui conti una sola fase, agente/algoritmo **greedy** o **miope**
- **a orizzonte indefinito**: numero di passi finito, ma non predeterminato
  - ad es., 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., modulo di stabilizzazione nei robot: se si fermasse una volta raggiunta la stabilità, dopo i robot cadrebbero di continuo

## 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*

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

- *Rappresentazione e modularità*

- moduli semplici in una gerarchia → ragionamento su insieme finito di stati
- altri livelli di astrazione richiedono il ragionamento su relazioni
- ad es. robot consegne:
  - modulo per il bilanciamento: pochi stati
  - modulo per la consegna ragiona sulle priorità deve ragionare su più individui (persone, pacchi, stanze,...) e loro relazioni
    - livello più alto, modulo che ragiona sull'intera attività giornaliera, pochi stati per le diverse fasi del giorno:
      - e.g., stati *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 → orizzonte potenzialmente indefinito

- ***Incertezza sull'osservazione e complessità del ragionamento***
  - molto più facile ragionare quando si conosce lo stato del mondo
  - incertezza su individui e relazioni 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 es. volo per Parigi: 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
    - compromesso tra soddisfare un fine importante con una certa probabilità o un fine meno desiderabile con ancor maggiore probabilità

- ***Molteplicità e modularità***
  - agente/sistema progettato attraverso più sistemi che interagiscono condividendo un ***fine comune*** di rendere intelligente il comportamento dell'agente al livello superiore
    - secondo alcuni [6], l'intelligenza è una caratteristica emergente da una “società” di agenti non intelligenti



- ***Apprendimento e rappresentazione***

- apprendimento spesso basato su feature
  - determinare i valori di feature che portano a migliori predizioni sui valori di una specifica altra feature: ***classificazione, regressione***
- ma si può lavorare anche su individui e relazioni
- apprendimento di gerarchie, in domini parzialmente osservabili e attraverso sistemi multipli
  - tutte modalità che interagiscono con diverse altre dimensioni

- ***Modularità e razionalità limitata*** → ragionamento più efficiente
  - il formalismo può diventare più complicato
  - ma per costruire sistemi complessi sono utili
    - suddivisione in componenti più piccole
    - approssimazioni per poter decidere in tempi accettabili e in regime di memoria limitata

# **APPLICAZIONI PROTOTIPICHE**✿

## 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 info 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

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 precedente):

- **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 richiede**

## 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*

Figure tratte da [1] salvo diversa indicazione

formatted by Markdeep 1.14 