

Corso di Ingegneria della Conoscenza

Progettazione di sistemi basati sulla conoscenza

Andrea Bonarini

Politecnico di Milano



Dipartimento di Elettronica e Informazione
Artificial Intelligence and Robotics Laboratory

Tel. (02) 2399 3525

Cos'è un KBS ?

Un KBS

è un sistema software

in cui viene rappresentata conoscenza

che viene utilizzata per risolvere problemi.

Un KBS è un sistema software

Ciclo di vita

come sono organizzate le varie attività necessarie per progettare, costruire e mantenere un prodotto

→ analogo a quello di sistemi software tradizionali

→ con peculiarità dovute alla presenza della conoscenza

KBS richiede rappresentazione della conoscenza

Identificazione di:

→ cosa rappresentare (contenuti e struttura)#

→ come rappresentare (formalismi e strumenti)#

KBS risolve problemi

Identificazione di:

- qual'è il problema (contenuti e struttura)#
- come si può risolvere (strategie)#

Agenti coinvolti

→ **Knowledge Engineer**

Responsabile del progetto, gestisce il project team, raccoglie, analizza e formalizza la conoscenza, producendo il modello concettuale

→ **Knowledge Programmer**

Partendo dal modello concettuale, implementa il sistema

→ **Esperto**

Fornisce la conoscenza

→ **Utente**

Usera' il sistema, dice cosa vorrebbe e se e' soddisfatto

→ **Management/Cliente**

Fornisce le risorse, decide il successo del progetto

Ciclo di vita per KBS

5 fasi:

- 1. Studio di plausibilità**
- 2. Costruzione del prototipo dimostrativo**
- 3. Costruzione del prototipo completo**
- 4. Implementazione ed installazione del sistema finale**
- 5. Manutenzione ed estensione**

Fase 1: studio di plausibilità

Obiettivi:

- analisi dell'area applicativa, identificazione di un dominio opportuno, selezione del problema da affrontare
- identificazione delle principali specifiche tecniche e funzionali del KBS e verifica della plausibilità dell'applicazione
- prime decisioni tecniche, primo schema di progetto, prima pianificazione approssimativa

Output: rapporto sulla plausibilità

Definizione di plausibilità

5 aspetti rilevanti:

- fattibilità tecnica dell'applicazione, considerando le caratteristiche specifiche del dominio e del problema, e lo stato dell'arte della tecnologia
- impatto sull'organizzazione, ed eventuali modifiche della stessa
- implementabilità, considerando le risorse umane e materiali
- analisi costi/benefici, considerando anche benefici indiretti e nel tempo
- opportunità ambientale, considerando le opportunità di finanziamento, l'interesse del top management, l'attitudine all'innovazione, la competitività in ambito commerciale, l'immagine dell'azienda, ecc.

Fase 2: costruzione del prototipo dimostrativo

Obiettivi:

- identificare il problema nella sua complessità e verificare la validità delle scelte fatte nella fase 1
- ottenere reazioni, critiche e suggerimenti da parte degli utenti finali, per raffinare le specifiche
- ottenere coinvolgimento ed impegno da parte del management e del potenziale cliente
- ottenere coinvolgimento degli esperti e degli utenti

Output: prototipo dimostrativo

Fase 3: costruzione del prototipo completo

Obiettivi:

- realizzazione di un prototipo completo e funzionante che risponda alle specifiche anche solo parzialmente, in quanto:
 - è installato in laboratorio, non nell'ambiente definitivo
 - è stato testato solo con esempi realistici, preparati ad hoc
 - non è ingegnerizzato ed è ancora inserito nell'ambiente di sviluppo

Output: prototipo completo, nuova versione del rapporto sulla plausibilità, contenente anche criteri di valutazione e validazione, pianificazione del progetto e specifiche tecniche

Fase 4: implementazione ed installazione del sistema finale

Obiettivi:

- realizzazione del sistema finale, con le funzionalità definite, operativo nell'ambiente definitivo
- analisi dettagliata dell'ambiente finale
- ampliamento del prototipo, oppure delivery system, oppure reimplementazione del prototipo.

Output: sistema finale, completo di manualistica (user, reference, maintenance manuals)#

Fase 5: manutenzione ed estensione

Obiettivi:

- supportare l'uso del KBS
- correggere errori e mancanze che possono emergere
- monitorare le performance del KBS, raccogliendo critiche, necessità, suggerimenti, per mantenere il KBS sempre in grado di rispondere alle esigenze degli utenti nel tempo.

Output: revisioni, raffinamenti, estensioni