

# Sistemi intelligenti distribuiti

Sistemi intelligenti distribuiti LS  
Prof. Andrea Omicini  
A.A. 2005-2006

# Cosa sono?

- Sistemi
- Intelligenti
- Distribuiti

# Il concetto di sistema

- Non distinguiamo più a priori sistemi SW e HW
  - almeno a un certo livello di astrazione
- Pensiamo in generale a sistemi artificiali
  - ossia costruiti dall'uomo direttamente o indirettamente
- Tra i sistemi artificiali mettiamo anche quelli naturali su cui l'uomo interviene
  - sistemi che sono artificiali in parte
- Utile interpretazione / astrazione
  - spiegare il comportamento dei sistemi in termini di
    - comportamento dei componenti
    - interazione tra componenti
    - interazione del sistema con l'ambiente

# Il concetto di intelligenza

- Un buon punto di partenza: Turing 1963 (sul sito)
- Discuteremo (poco) più avanti la storia della ricerca AI
  - ma questo è un altro discorso
- Per quello che riguarda il nostro corso, ci limitiamo ad adottare una nozione debole, un criterio “industriale”
  - diciamo intelligente il comportamento di un sistema che realizza un compito che richiederebbe altrimenti l’attività intelligente di un essere umano per essere portato a termine
- Quindi, questo non è un corso di AI “dura e pura”
  - è un corso in cui prendiamo da “Classic AI” quello che ci serve, e lo usiamo

# Il concetto di distribuzione

- Cosa è che è distribuito?
  - i sensori, gli attuatori, i dati, la capacità computazionale, i canali di comunicazione, ...
- Si perde l'unità spazio-temporale
  - non c'è più il tempo di sistema, né il luogo
  - i componenti di un sistema, ai vari livelli, sono solo parzialmente correlati temporalmente e topologicamente
- Che cosa cambia?
  - un numero di assunzioni non più valide, per esempio:
    - gli eventi in un sistema distribuito costituiscono in generale un insieme solo parzialmente ordinato
    - le interazioni possibili tra i componenti dipendono dalla topologia fisica / virtuale

# Sistemi complessi

- In generale, siamo interessati a capire / imparare a costruire sistemi complessi
  - SW/HW/anyware
- Internet come caso di studio
  - sistema distribuito, aperto, eterogeneo, decentralizzato, altamente dinamico, non predicibile
- Sistemi intelligenti pervasivi e ubiqui
  - cosa ci serve per costruirli?

# Ridurre la complessità

- Nessuna assunzione artificiosa per teorie disutili
  - ci interessa individuare approcci e strumenti efficaci, tanto teorici quanto tecnologico-applicativi
- Vogliamo trovare e studiare
  - modelli di componente che incapsolino le fonti di complessità, e consentano di veicolare tecniche “intelligenti” (da CS, OR, AI, ML, ...)
  - infrastrutture per governare l’interazione tra componenti e dominare la complessità dell’ambiente
  - linguaggi ricchi dal punto di vista del paradigma computazionale ma anche della tecnologia, per costruire e plasmare componenti e infrastrutture

# L'agente come modello di componente

- Premessa
  - non ci dimentichiamo del modello a oggetti, né delle corrispondenti metodologie e prassi
  - le conosciamo e sfruttiamo, ma vogliamo andare oltre: per i nostri scopi in questo corso, OO non basta
- La nozione di agente ha molte proprietà (controverse)
  - incapsula il controllo, supporta la distribuzione, veicola AI, introduce esplicitamente le astrazioni di società e ambiente, etc.
  - noi adotteremo quella/e nozione/i di agente adatta/e ai nostri scopi

# Le infrastrutture

- In generale, le infrastrutture sono portatrici non neutre di modelli, tecnologie e standard
- Implicitamente, promuovono o forzano metodi e processi di design e sviluppo caratteristici
  - implicitamente
    - per es., tramite un preciso modello di componente
  - esplicitamente
    - per es., fornendo tool di sviluppo specifici
- Danno forma allo spazio dell'interazione tra componenti, e (parzialmente) all'ambiente
- Sperimentaremo TuCSoN
  - ma siamo aperti a altre scelte, tipo JADE

# I linguaggi e i paradigmi

- **Premesse**
  - si presuppone una buona conoscenza del linguaggio Java, dei suoi connotati “architetture”, del paradigma OO, delle basi di ingegneria del software a oggetti (UML, RUP, Eclipse, ...)
  - si presuppone una conoscenza “di massima” del paradigma logico, e del linguaggio Prolog
- **Sperimenteremo l’uso di Prolog insieme a Java per costruire i nostri sistemi ad agenti**
  - tuProlog + Java come base del laboratorio e dei progetti
  - forse linguaggi specifici, come 3APL