

La fault-tolerance nei sistemi multi-agente

Emanuele Panzavolta

Approfondimento per il corso di Sistemi Multi-Agente

5 maggio 2009

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Outline

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Il contesto

Sistemi multi-agente: scenari

- paradigma adatto per sistemi
 - complessi
 - pervasivi
 - dinamici
- sistemi critici
 - vincoli di *dependability*
 - necessaria un'adeguata *fault-tolerance*

Fault-tolerance

- a diversi livelli
 - full fault-tolerance
 - graceful degradation
 - fail-safe
- revisione dei processi di software system-engineering
 - la fault-tolerance non va *aggiunta* a posteriori

Outline

- 1 Introduzione
- 2 I fault nei MAS**
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

I fault nei MAS

Fault: definizione

- *Qualunque avvenimento o situazione che non fa parte del funzionamento normale di un sistema*

Obiettivo

- Architetture per MAS fault-tolerant, a fronte di
 - infrastrutture inaffidabili
 - guasti hardware
 - crash software
 - errori di comunicazione tra nodi remoti
 - agenti malevoli
 - emergenza di comportamenti indesiderati

Criteri generali di progettazione

Best-practise

- modularità e isolamento
- ridondanza
- determinismo

Proprietà

- i MAS sono, al tempo stesso, intrinsecamente
 - fault-tolerant
 - autonomia degli agenti → isolamento
 - non fault-tolerant
 - autonomia degli agenti → non determinismo

Outline

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Approccio #1: agent-centric

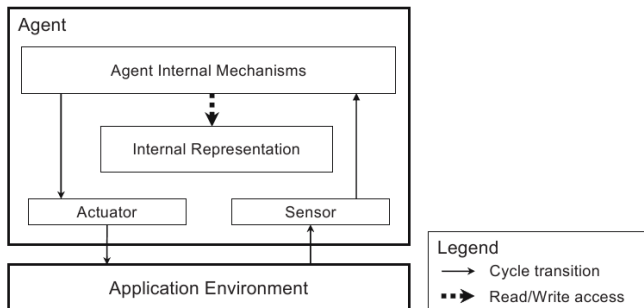
Revisione dei concetti nell'ambito MAS

- Concetto di eccezione / fault
 - *Un fault si ha quando un agente percepisce un evento come inatteso*
 - I fault hanno significato solo all'interno degli agenti

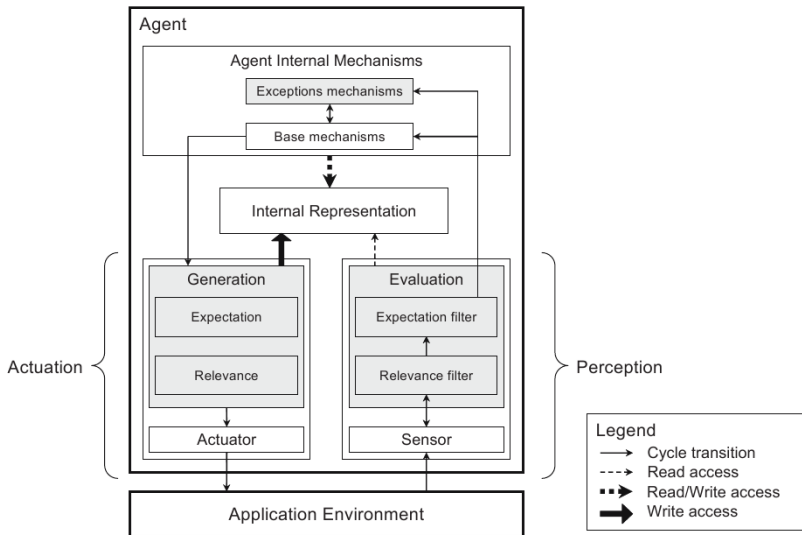
Procedure di exception-handling

- è opportuno che siano affidate agli agenti stessi (*problem-solving agents*)
 - nuova architettura di agente

Architettura tradizionale di agente



Architettura di agente con exception-handling



Outline

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali**
 - Agent-centric
 - **System-centric**
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Approccio #2: system-centric (1/3)

Il problema del non-determinismo

- autonomia degli agenti → non-determinismo
 - incompatibile con una vera fault-tolerance
- soluzione
 - limitare la libertà degli agenti
 - per salvaguardare l'integrità di sistema
 - introduzione di *sentinelle*

Approccio #2: system-centric (2/3)

Sentinelle

- sono agenti, addetti al controllo del sistema
 - osservazione
 - monitoraggio delle interazioni tra agenti
 - osservazione dello stato interno degli agenti
 - azione
 - proporre piani alternativi agli agenti
 - isolare agenti operanti in modo non corretto
- costituiscono un servizio di exception-handling
 - separato dai problem-solving agents

Approccio #2: system-centric (3/3)

Requisiti per gli agenti

- ① non sono più totalmente opachi
- ② dovrebbero supportare due linguaggi
 - query-language
 - con cui comunicare alle sentinelle il proprio stato
 - action-language
 - con cui le sentinelle indicano agli agenti le azioni di *fault-recovery* da effettuare
 - il supporto ai due linguaggi è preferibile, ma non obbligatorio
 - bene per sistemi aperti
 - migliore supporto ai linguaggi → migliore exception-handling

Outline

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Fault-fisici: la replicazione

Fault fisici

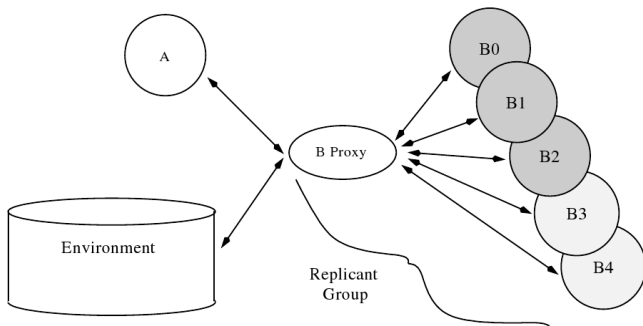
- crash hardware, bug software, etc.
- tipo di fault maggiormente studiato
 - soluzione tipica: ridondanza

Feature desiderate

- ① replicazione trasparente
- ② dinamicità delle politiche di replicazione
- ③ ottimizzazione delle risorse computazionali
 - ridondanza → overhead
 - varie modalità di replicazione
 - *cold-standby*
 - *hot-standby*
 - *active*

Architettura a proxy

- Gruppi di replicazione, gestiti in modo trasparente tramite proxy



Outline

- 1 Introduzione
- 2 I fault nei MAS
- 3 I due approcci principali
 - Agent-centric
 - System-centric
- 4 Fault fisici: la replicazione
- 5 Conclusioni

Conclusioni

- Settore di ricerca poco attivo
 - pochi lavori negli ultimi anni

fault-tolerance in generale

- nessuno dei due approcci convince pienamente
- Agent-centric
 - non adatto a sistemi aperti
 - scarsa modularità
- System-centric
 - perdita di autonomia degli agenti
 - le sentinelle possono essere un *single point of failure*

un caso specifico: tolleranza ai fault fisici

- ci sono proposte concrete
 - framework DARX (Dynamic Agent Replication eXtension)