

# Esplorazione di ambienti 2D tramite agenti eterogenei - istruzioni per l'esecuzione

Andrea Monaldini

13 gennaio 2006

## 1 Software necessario

Per l'esecuzione del software è necessario avere a disposizione:

- java 1.5 o superiore
- TuCSoN 1.3.0 o superiore
- tuProlog 1.3.0 o superiore

## 2 Avvio di un nodo TuCSoN e degli strumenti correlati

### 2.1 Avvio di un nodo TuCSoN

Nella release di TuCSoN è presente il file *tucson.jar*. Per avviare un nodo TuCSoN è necessario eseguire:

```
java -cp tucson.jar alice.tucson.runtime.Node
```

### 2.2 Avvio di un Inspector e di un CLIAgent

Al fine di monitorare e interagire con lo stato di un tuple centre e con le reazioni che gli sono state attribuite si consiglia di fare uso dello strumento "Inspector".

Avviare l'Inspector con:

```
java -cp tucson.jar alice.tucson.ide.Inspector
```

L'interfaccia dell'Inspector permette di scegliere quale tuple centre monitorare.

Nel caso sia necessario interagire con il tuple centre tramite un'interfaccia è possibile utilizzare:

```
java -cp tucson.jar alice.tucson.ide.CLIAgent
```

Per maggiori informazioni sull'uso di questi strumenti si rimanda alla documentazione di TuCSon.

### 3 Avvio del sistema di esplorazione

Il software sviluppato mette a disposizione un entry point (classe *explorer.ConfigurableSimulation*) che consente di configurare:

1. dimensione della griglia<sup>1</sup>
2. numero di celle inaccessibili
3. numero di agenti esploratori semplici
4. numero di agenti esploratori avanzati
5. file contenente la teoria prolog con la quale l'agente esploratore avanzato calcola il percorso tra due celle
6. file contenente le specifiche di reazione del tuple centre (ReSpecT)
7. nome del tuple centre da utilizzare (IMPORTANTE: il tuple centre utilizzato per la simulazione deve essere inizialmente vuoto e si deve tenere presente che al termine di ogni esecuzione il centro di tuple sul quale essa si è basata NON è vuoto, ma rimangono tuple necessarie alla comprensione del funzionamento del sistema: se si vuole riutilizzare lo stesso tuple centre si deve provvedere all'eliminazione di tutte le tuple tramite l'Inspector)

---

<sup>1</sup>La ricerca del percorso tra due celle eseguita per mezzo della teoria prolog allegata al progetto soffre, in presenza di un numero elevato di celle (griglie di dimensione superiore a 10x10), di problemi di esplosione della complessità di risoluzione, in quanto da ogni cella partono tre diversi percorsi possibili: è quindi opportuno, per il corretto funzionamento del sistema, mantenere la dimensione della griglia al di sotto delle 10x10 celle.

Quelli precedentemente indicati sono, nel corretto ordine, gli argomenti da passare alla classe *explorer.ConfigurableSimulation* per avviare una simulazione. Il posizionamento di celle inaccessibili, agenti esploratori semplici e avanzati è scelto in modo casuale, così come la direzione iniziale attribuita agli agenti semplici. La griglia generata è caratterizzata dalla presenza di muri su tutte le celle appartenenti al perimetro.

All'interno del file *sw.zip* è presente il file *explorer.jar* che contiene il sistema sviluppato. Nella directory *support\_files* sono inoltre presenti i files *AdvancedExplorerAgent.pl* (teoria prolog per l'agente avanzato) e *specificationTuples.respect* (file per la configurazione delle reazioni del tuple centre).

Per avviare una simulazione con una griglia di dimensione 8x8, 5 celle inaccessibili, 2 agenti semplici e 2 avanzati, utilizzando i files della teoria prolog e di di specifiche sopra citati, basando il sistema sul tuple centre "prova" si deve eseguire il comando (nell'ipotesi che ci si trovi nella stessa directory del file *explorer.jar*, *2p.jar* e *tucson.jar*) :

```
java -cp explorer.jar:2p.jar:tucson.jar explorer.ConfigurableSimulation 8 5 2 2
./support_files/AdvancedExplorerAgent.pl
./support_files/specificationTuples.respect prova
```

Una possibile griglia generata dal precedente comando è presente in figura 1.

Al fine di eseguire test sul sistema è stata creata una classe (*explorer.SimpleSimulation*) che avvia una simulazione configurando in modo deterministico le posizioni delle celle inaccessibili e degli agenti (figura 2).

Per avviare il sistema con questa configurazione utilizzare il comando:

```
java -cp explorer.jar:2p.jar:tucson.jar explorer.SimpleSimulation
```

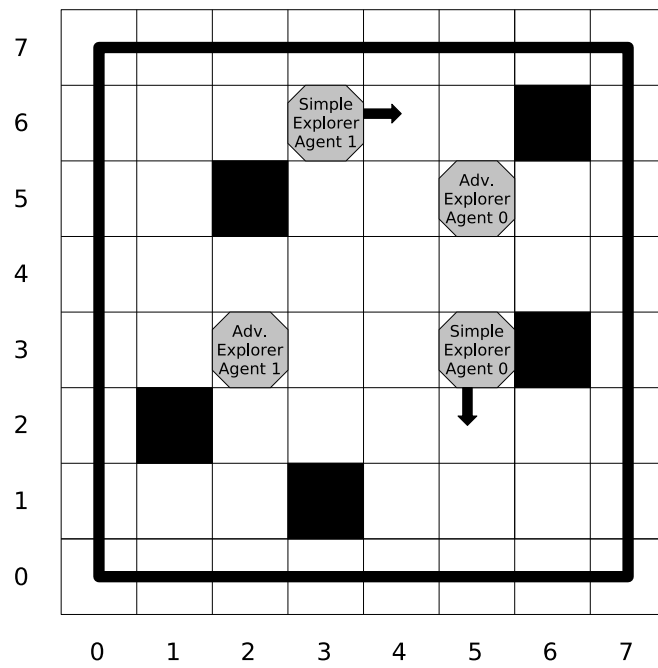


Figura 1: Esempio di griglia generata automaticamente con dimensione 8x8, 5 celle inaccessibili, 2 agenti semplici e 2 avanzati.

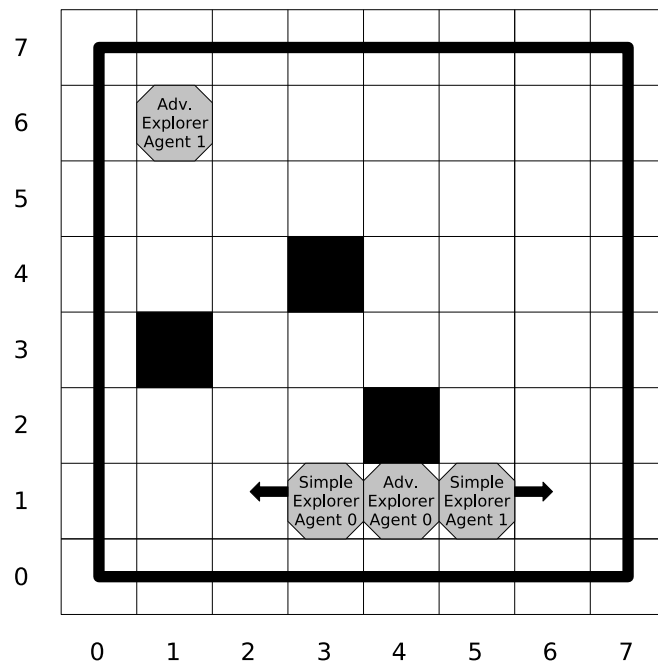


Figura 2: Configurazione iniziale impostata dalla classe `explorer.SimpleSimulation`