

Documento: il simulatore P2P Simulation.

Questo documento vuole fornire tutte le informazioni necessarie alla comprensione del funzionamento del sistema multiagente P2P Simulation (P2P-s) e sugli aspetti fondanti del progetto.

1. Abstract.

L'intenzione è quella di creare un sistema multiagente che si occupi della simulazione di un sistema Peer-to-Peer di tipo wireless. Vorrei focalizzare il progetto sul movimento e sull'interazione tra i peer, attraverso una reciproca capacità di riconoscimento e scambio di conoscenze. Gli agenti dovranno occuparsi di prendere il risultato di una scansione dello spazio attorno al terminale, in termini di presenza, nome del terminale (identificativo unico) e posizione (es. se ricevo il segnale del terminale A, la tupla di posizione avrà tre argomenti (Nome, Coordinata X, Coordinata Y). Quindi se A sente fisicamente B, avrà B nelle coordinate subito vicine alle sue.)

Questi risultati verranno inseriti nel centro di tuple dello spazio. Su ogni terminale l'agente "residente" può così controllare la situazione attorno a se in termini di mobili presenti; l'aggiornamento viene svolto ad ogni ciclo, quindi con il periodo deciso dal hardware. Se durante una scansione viene a mancare un terminale, questo viene eliminato dal centro di tuple.

Nel momento in cui due peer vengono a "contatto", viene attivata una procedura di scambio delle rispettive conoscenze, comprese quelle acquisite da altri peer durante il proprio percorso.

2. Analisi del sistema.

Il concetto di peer mobile è generico e comprende una serie di operazioni elementari che vanno spiegate e analizzate. Gli obiettivi di questo progetto consentono di fare delle assunzioni a riguardo, infatti volendo analizzare le dinamiche di comportamento degli agenti, risulta inutile specificare se con l'azione del "muovere" o dello "scambiare" si intenda qualcosa di maggiormente specifico. Inoltre, sempre per le stesse motivazioni, un'altra assunzione che non toglie generalità, è quella di considerare l'ambiente in due dimensioni (X,Y).

La rappresentazione dello spazio viene fatta operando una discretizzazione dello spazio stesso, ossia viene immaginato come una griglia le cui dimensioni siano quelle desiderate e in cui in ogni punto della griglia, identificato da due coordinate, può essere presente una possibile posizione di un peer.

Per chiarire il concetto, si veda la figura numero 1.

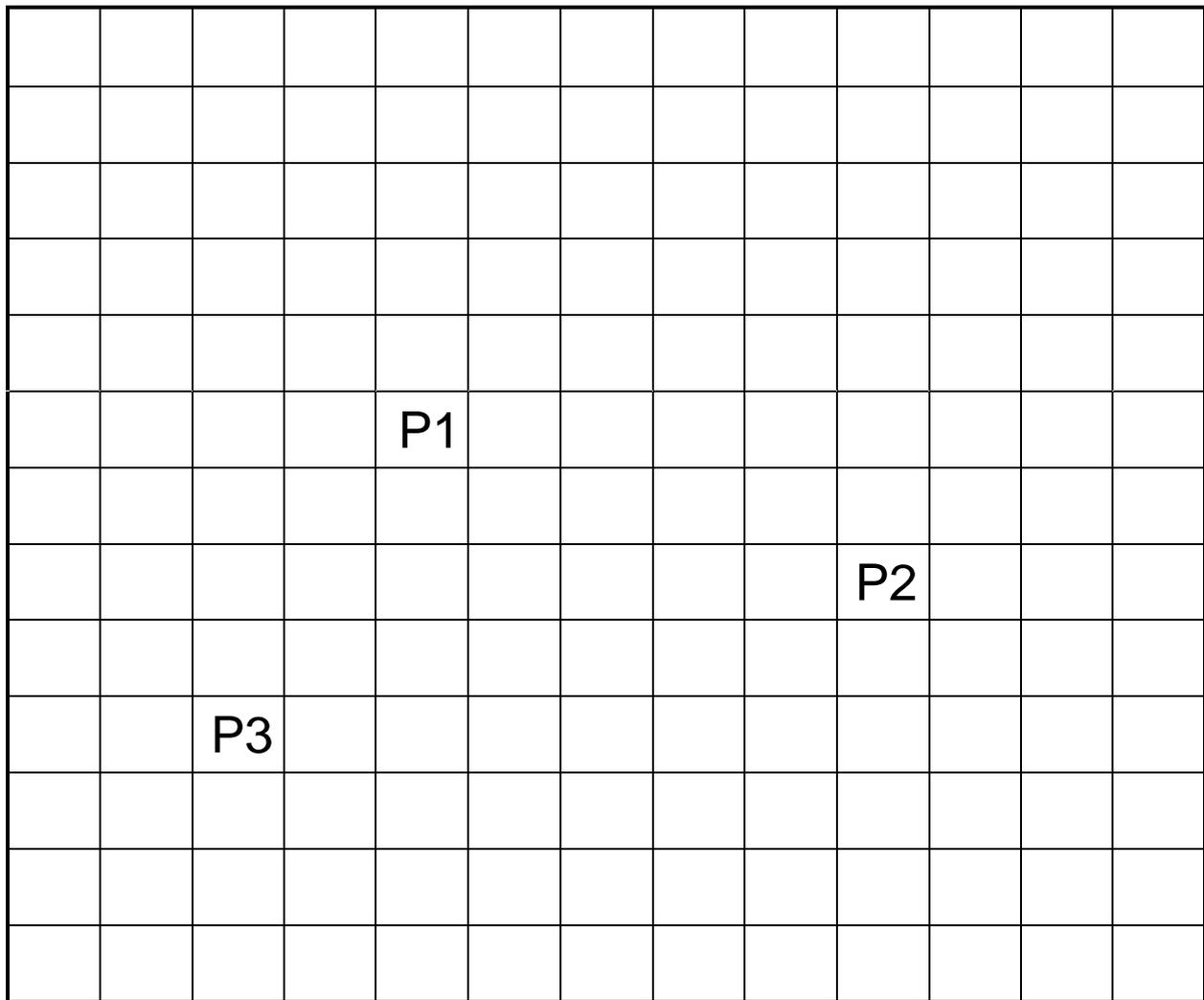


Figura 1: esempio di uno spazio discretizzato con una griglia.

Dalla figura possiamo notare come che uno spazio sia costituito da pareti (immaginarie, ci servono per capire qual è la zona di operabilità del sistema) che lo contornano e una serie di peer che lo occupano.

Quello appena descritto costituisce l'environment, le cui caratteristiche possono variare di volta in volta per dimensioni ed eventuale posizione di partenza dei peer (può essere utile, in simulazione, aggiungere oggetti per ricreare un ambiente particolare, per esempio indoor). Questo è l'ambiente nel quale devono operare gli agenti, e con il quale essi si devono rapportare per raggiungere il loro obiettivo (goal): la mobilità e lo scambio di conoscenze.

Le caratteristiche di ciascun agente devono essere tali da permettere loro di muoversi all'interno dello spazio nel quale si trovano e di interagire con esso, venendo a conoscenza di dove ci si può muovere e dove invece non si può proseguire. Supponiamo quindi che ogni peer sia in grado di trovare il "muro" che delimita lo spazio, agendo in risposta con un cambio della direzione di movimento. Gli agenti devono inoltre essere in grado, nella loro versione definitiva, di avere un percorso configurabile, ottenuto attraverso una serie di "punti obiettivo" da raggiungere all'interno dello spazio.

Supponiamo inoltre che gli agenti siano in grado di "parlare la stessa lingua", ovvero che si utilizzi uno stesso standard di comunicazione: questo ci permette di semplificare le operazioni di interazione tra i peer.

2.1 Lo sviluppo degli agenti e le loro tipologie.

Finora abbiamo analizzato le caratteristiche generali degli agenti e i loro “goal”, cioè i loro obiettivi; nel fare questo, dobbiamo però aggiungere come questo può essere realizzato.

Questo paragrafo ha l’obiettivo di spiegare qual è stato il percorso fatto per arrivare al risultato finale, quali scelte sono state compiute e perché, come si pensa di metterle in opera.

Come prima operazione, è stato necessario studiare l’interazione dell’agente con l’enviroment, quindi come questo si muove e come si comporta nelle situazioni critiche, ovvero il raggiungimento del limite dello spazio stesso. Per questo la prima realizzazione ha visto lo sviluppo di un semplice agente in grado di acquisire l’informazione sulla dimensione dello spazio, necessaria al controllo svolto “on board” sulla propria posizione.

Una volta ottenuto questo primo risultato, siamo passati al comportamento nel caso in cui l’agente raggiunga il limite dello spazio; dopo un’analisi sulle possibili azioni implementabili, si è subito compreso come il semplice cambio di direzione (assunto, per convenzione, verso destra) sia la soluzione più immediata e maggiormente coerente. Infatti la possibilità di un ritorno sui propri passi è sembrata piuttosto priva di significato, in quanto non portava ad una soluzione del problema, ma ad un semplice espediente.

Questa serie di scelte ha portato alla creazione dell’agente *peer0*, base di lavoro stabile per poter poi continuare lo sviluppo.

Il passaggio da questo semplice agente ad uno più complesso che implementasse il secondo goal (cioè l’arricchimento delle proprie conoscenze) si è svolto analizzando il secondo problema: come permettere la coordinazione tra due agenti in fase di scambio. Si è scelto di iniziare da una prima assunzione: che uno dei due peer sia fermo nella sua posizione. Questa assunzione, infatti, è fondamentale per semplificare il problema in questa fase dello sviluppo. Così facendo, ci si deve occupare solo del movimento di un agente, eliminando temporaneamente il problema nato dalla possibile (anzi, quasi certa) mancanza di sincronizzazione degli agenti.

Supponiamo allora che un agente sia in grado di interagire con l’altro una volta arrivato ad una distanza di un’unità di griglia. Per meglio spiegare, si veda la figura 2.

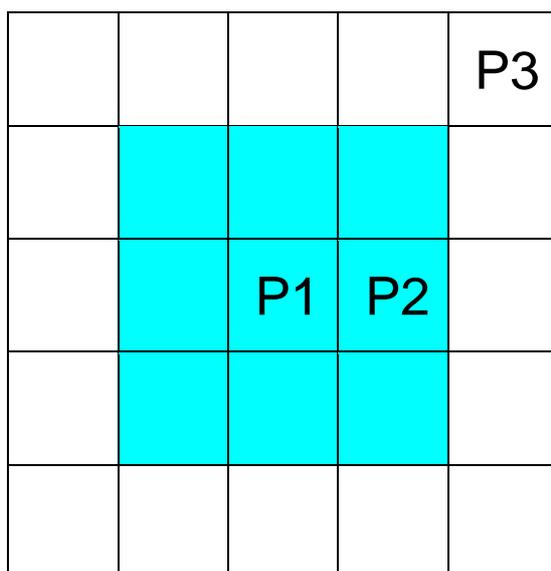


Figura 2: posizione del peer 1 con relativa area di visibilità (in azzurro).

Si nota subito come il peer 2 sia all'interno di quest'area, diventando quindi un possibile obiettivo della comunicazione. Al contrario, il peer 3 si trova fuori da quest'area, rendendo impossibile la comunicazione.

Arrivati a questo punto, si è ottenuto il *peer_1*, rappresentato dall'omonimo agente.

Terminata la prima fase, passiamo al miglioramento del comportamento degli agenti, aggiungendo la possibilità da parte dell'agente stesso di seguire un percorso desiderato e, infine, di coordinarsi con un altro in fase di comunicazione.

Il movimento lungo un percorso (configurabile dall'utente) viene realizzato attraverso una serie di obiettivi in termini di punti nello spazio; cos' facendo, l'agente si muove prima lungo uno degli assi, poi lungo l'altro, in modo da raggiungere le coordinate desiderate. Fatto ciò, si cambia obiettivo. L'unica scelta di un certo spessore è quella sul trattamento degli obiettivi già raggiunti; per creare un agente in grado di continuare il suo movimento fino all'arrivo di un segnale di stop, scegliamo di riposizionare gli obiettivi raggiunti in fondo alla lista degli obiettivi, in modo da tornare in prima posizione una volta finito il ciclo completo.

Il coordinamento tra agenti in fase di comunicazione risulta invece più complicato: il rispettivo movimento non coordinato può causare la mancanza di rilevazione di presenza di un peer, con conseguente fallimento dello scambio di conoscenze. Si sceglie allora di organizzare la flow-chart in modo da posizionare il movimento dell'agente come ultima operazione; questa scelta consente di avere diversi momenti per controllare l'intorno del peer. In questo modo, lo spostamento sarà possibile solo una volta assicuratisi che non ci siano altri agenti intorno a se.

Questa prima forma di coordinamento (forse impropria) non risulta però sufficiente a garantire la corretta interazione tra i peer; un ulteriore momento di coordinazione deve essere previsto una volta che gli agenti abbiano riscontrato la relativa presenza. Questo è possibile attraverso la scrittura di particolari tuple e l'attesa della loro lettura; cos' facendo, i due peer opereranno in sincronia, garantendo la corretta comunicazione. Questo agente sarà quello in grado di rappresentare il peer maggiormente efficiente, che chiameremo *peer_x_adv*.

2.2 Intelligenza e risvolti sociali.

Se per intelligenza intendiamo il comportamento di un sistema, che realizza un compito che richiederebbe l'intelligenza umana per essere portato a termine, non c'è dubbio che effettuare la coordinazione tra due enti e realizzare una comunicazione richieda intelligenza. Ne consegue che se si manifesta il raggiungimento del goal che gli agenti hanno come obiettivo essi devono considerarsi a tutti gli affetti intelligenti.

Dai risultati ottenuti si può allora affermare che tutte e tre le tipologie di agenti manifestano un comportamento intelligente, pur effettuando procedimenti diversi in termini di difficoltà. Risulta però indubbio che dei tre comportamenti se ne possa delineare una classifica per livello di intelligenza, si osservi a questo proposito il rapporto di ciascuna tipologia di agente con le seguenti caratteristiche.

- § *Proattività*. Ogni agente del MAS si può considerare proattivo in quanto è in grado autonomamente di selezionare la prossima zona da occupare riconoscendo la presenza di ostacoli e valutando, quindi, che il movimento sia corretto.
- § *Reattività*. Anche questa caratteristica è presente in ogni agente, in quanto tutti sono in grado di fronteggiare i cambiamenti che possono esserci tra uno spazio ed un altro e anche quelli che possono verificarsi all'interno dello spazio stesso, come ad esempio la variazione delle dimensioni.

- § *Abilità Sociale e Cooperazione*. Questo aspetto è quello che differenzia maggiormente il peer_x_adv dagli altri. Come si è detto più volte i peer devono *cooperare* per il raggiungimento del goal comune, cioè la diffusione della conoscenza, ma non si è specificato cosa questo significhi. Ogni agente, infatti, cerca di raggiungere sia l'obiettivo del raggiungimento del punto desiderato, sia l'arricchimento delle sue conoscenze, ottenuto ogni qualvolta viene incontrato un altro agente. Se si guarda però in modo più organico la totalità degli agenti, vediamo come il singolo obiettivo dell'accumulo di conoscenza è chiaramente parte di un goal più grande, rappresentato dall'accrescimento complessivo della conoscenza da parte di tutto il MAS.

3. Progetto.

La realizzazione del progetto è stata effettuata utilizzando Tucson e TuProlog. Tutti gli agenti sono stati realizzati in Prolog per la possibilità che offre di operare con predicati logici e ragionamenti inferenziali, mentre gli artefatti di coordinazione sono stati realizzati in Tucson.

La creazione dello spazio consiste nella creazione di un centro di tuple di nome "*Spazio*" nel quale vengono immesse le tuple necessarie alla descrizione delle dimensioni dello spazio stesso. Il controllo sulla consistenza delle tuple (consistenza delle dimensioni, che vengono prese maggiori di 0) viene fatta in Respect. Una volta creato lo spazio gli agenti possono cominciare ad operarvi e questo viene simulato con le query che gli agenti fanno al centro di tuple e le tuple che loro stessi aggiungono, ad esempio viene richiesta la presenza o meno di un altro agente e viene immessa la tupla contenente l'indicazione della propria posizione.

Nel momento in cui un agente peer va in esecuzione, esso crea un altro centro di tuple nominato "*Peer_x_adv*". Questo centro di tuple può essere visto come un centro di raccolta delle informazioni, utilizzato dagli agenti per monitorare il proprio stato, quello dell'ambiente. Inoltre su questi tuple centre vengono inserite quelle tuple che consentono di coordinare lo scambio delle conoscenze tra agenti.

4. Conclusioni.

Dopo l'implementazione del MAS e il suo testing, le numerose prove effettuate ci permettono di dire che:

- § Gli agenti peer_x non sono in grado (per costruzione) di percorrere in modo efficiente lo spazio di lavoro; questo porta ad una situazione di scarso aumento della conoscenza sia individuale che complessiva, con conseguente incapacità di raggiungere pienamente il goal. Per consentire di migliorare la situazione, occorrerebbe aumentare il numero di peer e predisporre alcuni di loro in una posizione fissa, tale da aumentare le possibilità di incontro.
- § Gli agenti peer_x_adv raggiungono sempre gli obiettivi indicati dall'analisi. Sono in grado quindi di percorrere efficacemente lo spazio di movimento, ricreando le traiettorie desiderate. Questo fa sì che sia altamente più probabile l'incontro tra agenti, richiedendo inoltre un tempo di esecuzione minore. L'aumento della probabilità d'incontro porta inevitabilmente all'aumento delle conoscenze individuali e di tutto il sistema.

Alla luce di questi risultati è chiaro che ci sono margini di miglioramento nelle prestazioni, soprattutto per quanto riguarda l'ottimizzazione dei percorsi (che risulta ancora abbastanza lento). Si è comunque riusciti nell'intento che ci si era prefissati, cioè di realizzare un sistema multi agente

che simulasse le dinamiche di lavoro di un sistema peer-to-peer e soprattutto si sono toccati aspetti molto interessanti della programmazione ad agenti.