

Possono i mezzi di coordinazione essere utili per
fare MABS (simulazione basata ad agenti)?

Lorenzo Ravaglia

8 maggio 2009

Sommario

All'interno del corso di **Sistemi multi-agente LS** fra gli argomenti trattati sono stati presentati diverse mezzi di coordinazione per la regolazione dell'interazione fra agenti. Questo aspetto è una parte molto importante all'interno della realizzazione di simulazioni che si basano su sistemi formati da un certo numero di agenti che collaborano per raggiungere un obiettivo comune.

All'interno di questo report verrà valutato l'utilità derivante dall'utilizzo di tali mezzi per la realizzazione di simulazioni basate su sistemi multi - agente.

Indice

1	Sistemi MAS	2
1.1	Coordinazione	3
1.2	Spazi di tuple	4
1.2.1	Centri di tuple	5
1.3	TuCSon	6
1.4	Il linguaggio ReSpecT(<i>Reaction Specification Tuple</i>)	8
2	Simulazione e MAS	9
2.1	Modelli	9
2.1.1	ABM - Agent Based Model	11
2.2	Simulazione	12
3	Utilizzo dei mezzi di coordinazione per MABS	13
3.1	Modellazione	13
3.2	Implementazione del modello	14
3.3	Mapping	15
3.4	Esempio	16
4	Conclusioni e possibili sviluppi	19

Capitolo 1

Sistemi MAS

I sistemi multi-agente (MAS) sono un paradigma per la modellazione, la conoscenza e progettazione di sistemi complessi e provvedono a fornire un set di strumenti ad alto livello di astrazione che danno la possibilità di rappresentare un sistema complesso. Un sistema di questo tipo è formato da un certo numero di agenti che interagiscono e agiscono sull'ambiente.

Le parti principali che caratterizzano un sistema multi - agente sono:

- agenti
- società
- ambiente

Gli agenti possono interagire in diverse modalità:

- comunicazione diretta ‘colloquiando’
- comunicazione tramite scambio di messaggi
- modificando lo stato dell'ambiente (stigmergia)
- tramite mezzi di comunicazione (canali, spazi condivisi, ecc.)

All'interno di un MAS è necessario modellare i seguenti aspetti:

- coordinazione: governare e gestire cioè le azioni osservabili, ovvero le interazioni durante l'esecuzione
- organizzazione: coordinazione dal punto di vista statico, verranno definite politiche, permessi e fini, ovvero quali sono le interazioni possibili e quelle non possibili
- sicurezza: coinvolge la parte statica e dinamica; in questa parte viene specificato ciò che non è possibile fare, è duale dell'organizzazione e della coordinazione

Andiamo a concentrare ora l'attenzione sull'aspetto di coordinazione.

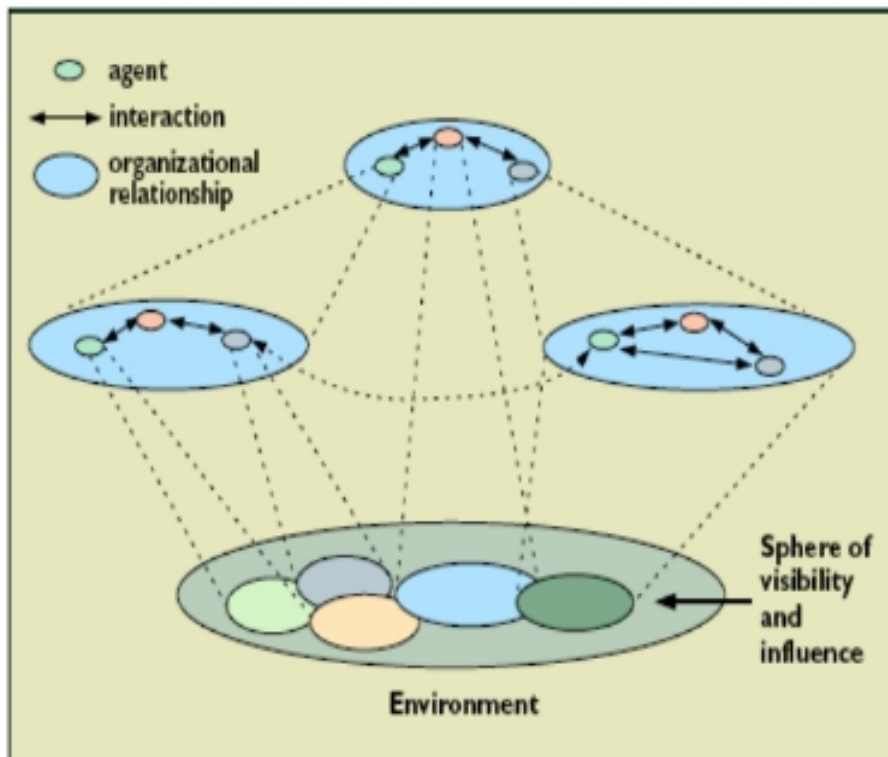


Figura 1.1: Schema di un sistema MAS

1.1 Coordinazione

Per coordinazione si intende la regolazione e gestione di elementi diversi affinché operino tra loro in modo coordinato, ossia integrato ed armonioso, con lo scopo di raggiungere un preciso obiettivo comune, o più d'uno. Consiste nel processo di integrazione e/o creazione di collegamenti fra parti differenti di un'organizzazione per conseguire una serie di obiettivi collettivi.

Di conseguenza, all'interno di qualunque struttura organizzata o gruppo sociale, gli individui incaricati di coordinazione rivestono una grande importanza, poiché le loro scelte possono portare al successo o, al contrario, al mancato raggiungimento degli obiettivi.

All'interno di MAS per coordinazione si intende l'attività volta a gestire e vincolare lo spazio di interazione fra agenti. In generale, un modello di coordinazione tratta con la creazione e distribuzione di agenti, con la comunicazione e mobilità nello spazio, con le attività di sincronizzazione e distribuzione delle attività. E' possibile modellare la coordinazione utilizzando due differenti meta - modelli:

1. meta - modello formato da una parte coordinabile e un medium di

coordinazione

2. meta- modello formato da tre elementi principali: entità, media e leggi di coordinazione

Nel primo caso è il medium di coordinazione che si preoccupa di abilitare le interazioni ammissibili tra le entità rispettando le leggi di coordinazione.

Nel secondo caso il compito viene modulato in tre componenti:

- entità di coordinazione: entità le quali interazione sono regolate dal modello (parte osservabile)
- media di coordinazione: abilita e regola le interazioni fra gli agenti
- leggi di interazione: stabiliscono il comportamento del media di coordinazione, formato da
 - linguaggio di comunicazione: sintassi utilizzata per l'espressione e lo scambio di dati
 - linguaggio di coordinazione: set di primitive di interazione con la loro semantica

1.2 Spazi di tuple

I modelli basati sugli spazi implementano una comunicazione indiretta fra entità attive: lo studio di questi è nato per creare nuovi modelli e linguaggi di coordinazione. Possiamo avere *medium di coordinazione* di due diversi tipi:

- data - driven: gli agenti interagiscono producendo e consumando dati/informazioni nello spazio condiviso
- control - driven: gli agenti interagiscono tramite la generazione e la reazione ad eventi/segnali su parti connesse da canali

Utilizzando il secondo meta - modello visto in precedenza, per gli spazi di tuple avremo i seguenti componenti:

- entità di coordinazione $\implies$ agenti
- media di coordinazione $\implies$ spazio di tuple
- linguaggio di comunicazione $\implies$ tuple
- linguaggio di coordinazione $\implies$ primitive dello spazio di tuple

Per realizzare uno spazio di tuple possono essere utilizzati diversi linguaggi fra cui Java e Linda; quest'ultimo ingloba sia il linguaggio comunicativo che quello di coordinazione. Le caratteristiche principali di Linda sono le seguenti:

- tuple: collezione ordinata in modo eterogeneo di conoscenza
- comunicazione generativa: il mittente e il destinatario possono interagire senza conoscersi; le tuple non sono legate a chi le ha generate e sono accessibili a tutti
- accesso associativo: le tuple vengono accedute tramite il contenuto
- semantica sospensiva: possiamo utilizzare operazioni bloccanti sulle tuple

Dal punto di vista della coordinazione abbiamo però una limitazione forte, perché gli spazi di tuple hanno un comportamento che viene settato solamente una volta, al momento della creazione. L'obiettivo sarebbe ottenere un medium di coordinazione con un comportamento modellabile in modo da contenere al proprio interno la soluzione al problema della coordinazione; in questo modo si avrebbe una divisione del problema in parte sociale ed individuale.

Per raggiungere l'obiettivo sopra specificato, vengono utilizzati come medium di coordinazione i centri di tuple.

1.2.1 Centri di tuple

I centri di tuple sono spazi dati reattivi e condivisi, distribuiti sui nodi dell'infrastruttura, nei quali gli agenti possono inserire, leggere e prelevare informazioni sotto forma di tuple.

I centri di tuple consistono in spazi di tuple migliorati; potremmo, quindi, separare due parti fondamentali:

- spazio di tuple classico
- comportamento programmabile, dove vengono specificate le reazioni da attuare all'avvenire di specifici eventi

La parte del comportamento programmabile è formata da reazioni che vengono attuate al verificarsi di certi eventi di comunicazione, lettura, scrittura o prelevamento di tuple. Utilizzando i centri di tuple possiamo avvalerci di una coordinazione ibrida, ovvero avere un medium che può essere di due tipi:

- information-driven: ciò che il progettista vuole rendere visibile agli agenti
- action-driven: il centro osserva gli eventi e reagisce in base alle regole incapsulate al suo interno

1.3 TuCSoN

TuCSoN (*Tuple Centres Speard over the Network*) è un modello, con annessa infrastruttura e tecnologia, per la gestione della coordinazione di agenti in rete, che impiega *centri di tuple* per risolvere il problema di comunicazione e di coordinazione [1].

Per quanto riguarda la configurazione del comportamento del centro di tuple le reazioni vengono espresse tramite il linguaggio ReSpecT (*Reaction Specification Tuple*).

Lo spazio di interazione è formato da molteplici centri di tuple sparsi sui vari nodi. Ogni centro è identificabile univocamente tramite il nodo sul quale risiede e può essere acceduto tramite l'indirizzo Internet completo nella seguente forma:

$$<nomeLocale>@<nodo>$$

I centri di tuple TuCSoN forniscono agli agenti una serie di primitive per la comunicazione fra cui [14]:

- out(T): inserisce la tupla T
- in(TT): preleva la tupla T che fa match con la tupla TT, primitiva bloccante
- rd(TT): legge la tupla T che fa match con la tupla TT, primitiva bloccante
- inp(TT): versione non bloccante della primitiva in()
- rdp(TT): versione non bloccante della primitiva rd()
- set_spec: imposta il comportamento del centro di tuple specificato, la tupla deve essere un programma valido scritto in ReSpecT
- get_spec: ritorna il comportamento corrente del centro di tuple specificato

Tramite TuCSoN è possibile creare diversi centri all'interno dello stesso nodo, ognuno programmabile separatamente e indipendentemente dagli altri.

Con la programmazione dei centri di tuple si possono creare modelli di sincronizzazione e standard di comunicazione incapsulati all'interno del mezzo di comunicazione; in questo modo, i centri di tuple fungono da *artefatti di coordinazione*. Fra i vari modelli implementabili abbiamo i protocolli del tipo Client Server e la gestione della coordinazione tramite semafori, barriere di sincronizzazione e altri modelli [15].

Nelle ultime versioni è stata inserita la gestione della persistenza dei centri, in modo che quest'ultimi siano dinamicamente persistenti. Rendere

persistente un centro di tuple significherà, perciò, salvare il suo contenuto in caso di shutdown del nodo o di eventuali crash del sistema: al riavvio del nodo il centro verrà automaticamente ripristinato.

Per fare in modo che gli agenti possano accedere ad un'infrastruttura TuCSoN è necessario sviluppare prima di tutto un ACC (*Agent Coordination Context* [13]) che consiste in una sorta di interfaccia che un'agente deve avere a disposizione per conoscere ed invocare le primitive sul centro di tuple, questo per favorire l'organizzazione e la sua "vita" all'interno di TuCSoN.

Un ACC descrive l'ambiente d'interazione fornendo all'agente:

- rappresentazione dell'ambiente dove dovrà vivere, interagire e comunicare
- framework per l'interazione all'interno del MAS
- insieme delle interazioni ammissibili
- tool per la configurazione dinamica del contesto

Un ACC deve poter essere configurato ed ispezionabile sia da parte degli umani che da parte degli agenti. La dinamica è descritta da due fasi:

negoziazione: fra agenti ed infrastrutture MAS, per iniziare una nuova sessione di lavoro all'interno dell'organizzazione dove vengono specificati i ruoli da attivare

uso: gli agenti possono utilizzare l'ACC per interagire con l'ambiente di organizzazione tramite le azioni e le percezioni rese possibili dallo stesso

L'ACC è un'astrazione che unifica il concetto di organizzazione e sicurezza. Uno dei modelli di ACC potrebbe essere descritto tramite tre aspetti:

- interfaccia: insieme di operazioni ammissibili messe a disposizione dall'infrastruttura per l'interazione con l'ambiente
- contratto: descrive le relazioni fra agente e ambiente, politiche e protocolli di interazione
- configurazione: descrizione dello stato dell'ACC a run - time

Tutte le caratteristiche viste in precedenza rendono TuCSoN un'infrastruttura idonea per il supporto di sistemi MAS.

1.4 Il linguaggio ReSpecT (*Reaction Specification Tuple*)

Con questo linguaggio possiamo definire il comportamento di un centro di tuple in risposta ad eventi di comunicazione. Un programma ReSpecT ha la forma di un set di reazioni di tipo [3]

$$reaction(E, G, R)$$

dove sono definite le componenti:

- E evento, che innesca la reazione
- G guardia, se all'avvenire dell'evento E questa è verificata viene scatenata la reazione
- R reazione, corpo della reazione

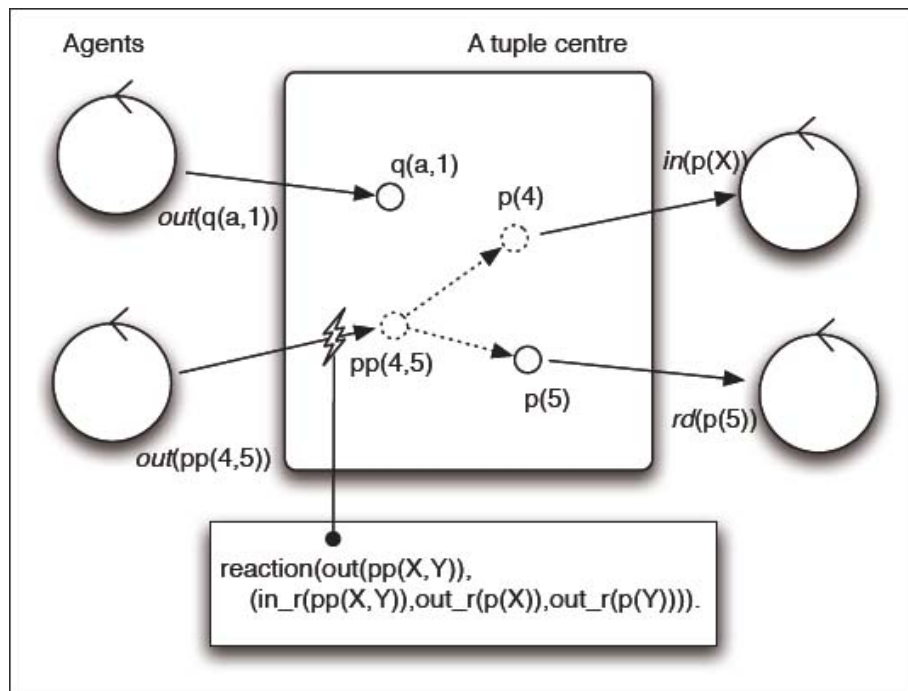


Figura 1.2: Interazione fra agenti tramite centro di tuple

Capitolo 2

Simulazione e MAS

Svariate discipline, nei diversi ambiti scientifici, si sono proposte come obiettivo l'accurata comprensione dei sistemi complessi e la predizione del loro comportamento. Oltre agli strumenti tradizionali di indagine sperimentale, vengono utilizzati allo stesso scopo anche strumenti di modellazione e simulazione, che favoriscono l'analisi del comportamento e delle interazioni fra entità del sistema e, inoltre, permettono di studiare il verificarsi di fenomeni emergenti. Vengono fatti, perciò, esperimenti "virtuali" per predire il comportamento sotto diverse condizioni.

2.1 Modelli

Il modello è una rappresentazione astratta di un sistema utilizzato per investigarne le proprietà e per studiarne le caratteristiche. Solitamente è formato da un insieme di oggetti (o variabili) e dalle loro relazioni. La cosa importante durante la modellazione è l'identificazione degli elementi che compongono il sistema e delle regole che definiscono le proprietà di interesse dello stesso. Il presupposto per un buon modello è la corretta costruzione con documentate e ben definite connessioni fra il modello stesso e gli aspetti del sistema che esso rappresenta.

Per la costruzione di un modello può essere utilizzata la seguente metodologia:

- fase di modellazione dei concetti - *domain model*, durante la quale viene analizzato il sistema reale in tutte le proprie caratteristiche
- fase di specificazione - *design model*: traduzione del modello descritto tramite un modello formale che non fa riferimento a *tool* e piattaforme
- fase implementativa - *computational model*: implementazione del modello per una piattaforma

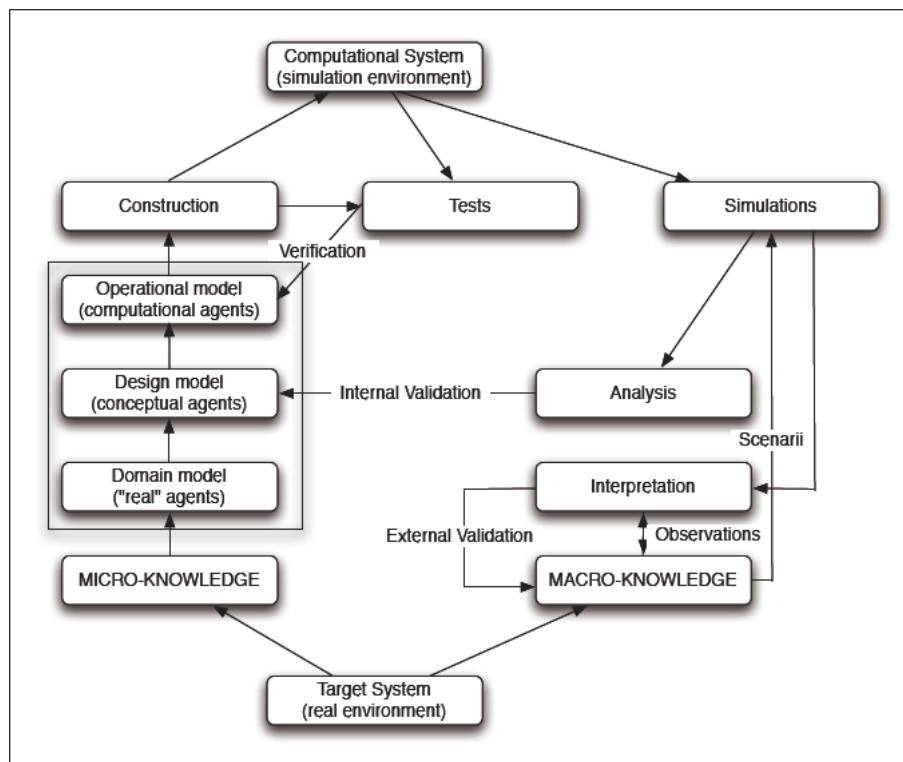


Figura 2.1: Metodologia per la creazione di MABS

Un buon modello deve avere le seguenti proprietà:

- correttezza: imitare gli aspetti di interesse della realtà
- precisione: misurata attraverso la differenza fra i risultati della simulazione e quelli della realtà
- generalità: in grado di descrivere le varianti della realtà ed essere il più scalabile possibile
- affidabilità: la simulazione sul modello deve dare sempre risultati corretti, in qualsiasi condizione; non si dovranno quindi avere risultati completamente sbagliati per una determinata configurazione
- costo: minimizzare il costo necessario per la creazione del modello

Sono presenti modelli tradizionali utilizzati per diversi tipi di simulazione:

- Graph: modello statico e a livello micro
- Boolean Network Dinamic: modello deterministico, sincrono, tempo discreto e a livello micro

- Equazioni differenziali: modello deterministico o stocastico, continuo, e a livello macro

2.1.1 ABM - Agent Based Model

Un ABM è un modello computazionale per la simulazione di azioni ed interazione di individui autonomi [9, 8]. Combina elementi di *game theory*, sistemi complessi, effetti emergenti, MAS, *computational sociology* e *evolutionary programming*. Molto spesso gli ABM sono implementati tramite sistemi MAS. Come già visto, questi ultimi permettono di sviluppare e progettare agenti, società ed ambiente e forniscono metodi per la modellazione di:

- strutture e comportamenti individuali
- interazioni fra entità locali
- struttura e dinamica dell'ambiente

Possiamo quindi definire in ABM le due componenti principali che possono interagire fra loro:

- agenti: vengono definiti sensori e attuatori, comportamento interno (reattivo e proattivo) e lo stato dell'agente
- ambiente: vengono definite topologia e dinamiche interne complesse

I modelli ABM hanno le loro radici storiche nel campo di sistemi CAS (*Complex Adaptive System*), i quali sono stati originariamente motivati dalle ricerche sull'adattività e i fenomeni emergenti all'interno di sistemi biologici.

Quando viene eseguito un modello ABM è possibile studiarne l'evoluzione tramite diversi punti di vista:

- osservazione dell'evoluzione individuale e dell'ambiente
- osservazione delle proprietà globali del sistema come fenomeni emergenti

Utilizzando questa tipologia di modelli si hanno diversi vantaggi:

- formulare comportamenti adattabili da parte degli individui
- rappresentazione di interazioni fra individui
- modellazione di spazi non omogenei
- simulazione con interazioni multiple

2.2 Simulazione

Le simulazioni sono uno strumento sperimentale molto potente e si avvalgono delle grandi capacità di calcolo offerte dall'informatica [11]. La simulazione, infatti, non è altro che la trasposizione di un "modello concettuale" della realtà; quest'ultimo può essere definito come l'insieme di processi che hanno luogo nel sistema valutato e il cui insieme permette di comprendere le logiche di funzionamento del sistema stesso. Attraverso la simulazione vengono effettuate analisi di sistemi semplici e complessi in modo da capirne il comportamento e si ha la possibilità di sottoporli ad azioni difficilmente realizzabili nella realtà o realizzabile ma molto costose.

La simulazione è uno strumento molto importante per i ricercatori, in quanto fornisce risultati che possono essere integrati o utilizzati per la verifica di quelli ottenuti in laboratorio.

Grazie alle simulazioni è possibile conoscere le conseguenze dovute ad un particolare insieme di fattori e all'evoluzione dinamica di un modello. Si tratta della tipica domanda "what if" a cui è possibile dare risposta modificando alcuni parametri del modello.

Utilizzando la simulazione al computer gli utilizzatori sono in grado di ottenere tre obiettivi differenti:

1. proiezione: prevedere gli avvenimenti futuri legati ad un fenomeno
2. sperimentazione: spiegare particolari fenomeni e verificare la qualità del modello utilizzato
3. esplorazione: osservare e studiare diversi aspetti dinamici e l'evoluzione del sistema modellato

La simulazione viene utilizzata per lo studio di sistemi complessi e per la formulazione di nuove teorie nel caso in cui:

- il sistema non può essere osservato
- il tempo di evoluzione del sistema è troppo breve o troppo lungo
- il sistema da simulare non esiste o non ancora
- il sistema è molto complesso

Sebbene la simulazione basata su MAS sia molto vantaggiosa, non sono stati ancora definiti linguaggi formali per descrivere adeguatamente un modello basato su agenti, mentre sono stati sviluppati diversi framework per effettuare simulazioni come, ad esempio, Swarm, RePast, MASON e NetLogo [2][6].

Capitolo 3

Utilizzo dei mezzi di coordinazione per MABS

In questa parte si andranno ad analizzare i processi di modellazione e simulazione per cercare di capire se è possibile utilizzare i mezzi di coordinazione per effettuare MABS [10].

3.1 Modellazione

Come detto in precedenza per effettuare una simulazione dovremo innanzitutto creare un modello che rappresenti il sistema. Faremo riferimento, in questo caso, ad ABM. Per la costruzione di un ABM è possibile utilizzare la seguente metodologia:

- identificare le tipologie di agenti ed altri oggetti con le loro caratteristiche
- definire l'ambiente all'interno del quale gli agenti vivranno e interagiranno
- specificare i metodi con i quali gli attributi degli agenti vengono modificati, in seguito alla propria interazione con altri agenti o con l'ambiente
- definire metodi che controllano quali agenti interagiscono, quando interagiscono, e come interagiscono durante la simulazione
- implementare il modello dell'agente tramite linguaggi software

Analizzando la metodologia vediamo che i mezzi di coordinazione possono essere utilizzati in diversi step:

- modellazione dell'ambiente: utilizzare un mezzo di coordinazione con un determinato comportamento per modellare l'ambiente della simulazione

- interazione: mezzo di coordinazione che gestisce le interazioni fra gli agenti
- implementazione: utilizzo di tool e tecnologie per la creazione di mezzi di coordinazione

Prendendo come riferimento il meta-modello di coordinazione possiamo così mappare le varie componenti di un ABM:

- entità di coordinazione $\implies$ agenti
- media di coordinazione $\implies$ rappresentato dal mezzo di coordinazione
- leggi di interazione $\implies$ incapsulate all'interno del mezzo di coordinazione

Possiamo quindi utilizzare i mezzi di coordinazione per la creazione di ABM. Dopo aver deciso il tipo di agenti da utilizzare ci si può concentrare sull'implementazione del mezzo di coordinazione, facendo particolare attenzione alle leggi che lo governano e che successivamente andranno a modellare le interazioni derivanti direttamente dal modello.

3.2 Implementazione del modello

Per implementare un modello ABM possiamo utilizzare TuCSoN come infrastruttura partendo dalla descrizione del modello (*design model*); dobbiamo quindi preoccuparci dell'implementazione dell'ambiente e della gestione delle interazioni.

L'ambiente può essere rappresentato come un singolo centro di tuple o distribuito fra diversi centri. Essi sono collegati fra di loro in modo che ogni agente abbia la possibilità di muoversi da una parte all'altra all'interno di tutto l'ambiente, così da poterlo esplorare nella sua interezza [12].

Per caratterizzare l'ambiente è necessario specificare:

1. le varie tipologie di tuple che devono essere inserite all'interno del centro e quale parte del modello rappresentano
2. le leggi che governano le interazioni fra agenti ed ambiente
3. le leggi che gestiscono le interazioni fra i vari agenti
4. stabilire quali sono i ruoli dei vari agenti in modo da definire quali azioni possono effettuare sull'ambiente

Strutturando quindi in uno specifico modo le tuple, siamo in grado di modellare l'ambiente seguendo le specifiche del modello, e di gestire le varie interazioni utilizzando una delle proprietà dei centri di tuple. Questa ci permetterà di mostrare agli agenti tuple presenti all'interno del centro che, in

verità, saranno diverse da quelle effettivamente presenti nello stesso. Quest'ultima proprietà può essere utilizzata anche per gestire i vari permessi collegati ai ruoli, sarebbe ad esempio possibile nascondere un determinato insieme di tuple agli agenti che non hanno un ruolo abbastanza alto.

Per quanto riguarda il comportamento dell'ambiente, è modellabile tramite la programmazione del comportamento del centro di tuple con ReSpecT.

A questo punto abbiamo un modello ABM che utilizza centri di tuple per la rappresentazione dell'ambiente, che fungono anche da mezzo di comunicazione e coordinazione. Ora non rimane che calibrare il sistema, identificare ed attuare l'esperimento ed, infine, analizzare i risultati della simulazione.

3.3 Mapping

Tramite una delle possibili metodologie per la progettazione, lo sviluppo e l'effettuazione di simulazioni utilizzando modelli ABM, andiamo ora a verificare se l'infrastruttura TuCSoN è adatta per effettuare simulazioni su MAS.

La metodologia è composta dai seguenti step [7]:

Step1 identificare l'obiettivo della simulazione

Step2 identificare lo scopo, le assunzioni e i parametri del modello giustificandoli

Step3 pianificare gli esperimenti, ovvero specificare la lista di simulazioni da effettuare

Step4 caratterizzare il sistema, questo step si sviluppa a sua volta in:

1. identificazione degli agenti da definire:
 - quali sono le entità (agenti) attivi che prendono parte al processo che viene studiato
 - qual'è il loro ruolo
 - quali sono le interazioni fra diversi agenti
 - quali sono le interazioni fra agente e ambiente
 - quali sono i risultati delle interazioni
 - quali entità sono prodotte dall'aggregazione di entità semplici
2. caratterizzazione dell'ambiente, definendo l'aspetto spaziale (mapa) e le sue possibili funzioni
3. definizione del comportamento degli agenti

Step5 specificare il modello multi - agente, ovvero implementare il software specificando i seguenti componenti:

1. definizione delle classi dell'agente e delle sue strutture (*behavior - oriented* o *goal - oriented*)
2. definizione degli elementi dell'ambiente e delle loro relazioni con gli agenti
3. definizione delle strutture di organizzazione e interazione
4. definizione delle procedure computazionali per ottimizzare le prestazioni

Utilizzando la programmazione dei centri di tuple possono essere definite:

- le interazioni fra gli agenti e fra agenti ed ambiente rendendo disponibili o meno alla vista degli stessi un certo insieme di tuple
- i risultati delle interazioni, decidendo quali e quante tuple vengono prodotte come conseguenza di una certa interazione
- gli elementi dell'ambiente e la loro relazione con gli agenti tramite la scelta della struttura (definizione sintattica e semantica) delle tuple
- le procedure computazionali per ottimizzare le prestazioni

Tramite la distribuzione dei vari centri di tuple del sistema è possibile definire:

- la caratteristica spaziale dell'ambiente decidendo se utilizzare un unico centro di tuple totalmente visibile a tutti gli agenti oppure se utilizzarne diversi in modo da poter realizzare la topologia del sistema
- la struttura organizzativa e d'interazione del sistema

3.4 Esempio

Prendiamo ora il modello utilizzato all'interno della simulazione delle cellule staminali ematopoietiche per constatare la possibilità di inserire mezzi di coordinazione, nel nostro caso specifici centri di tuple, per la modellazione. Verrà preso come riferimento il modello ideato da Mark d'Inverno e dai suoi collaboratori al Goldsmiths College (University of London) [4][5].

Ambiente

L'ambiente è diviso in regioni ed ognuna di esse può contenere cellule e sostanze chimiche. Le sostanze chimiche possono essere di due tipi:

- quelle prodotte dalle cellule staminali
- quelle prodotte dalle cellule stromali

ed avere una concentrazione variabile che va da 0 a 255.

La diffusione di sostanze chimiche è stata modellata in modo che ad ogni step la concentrazione di sostanze chimiche in ogni regione venga calcolata tramite la media delle concentrazioni della regione e di quelle vicine.

Per modellare questo ambiente sarebbe possibile:

- impiegare un centro di tuple per ogni regione, che potrebbe essere situata in un nodo
- definire sintassi e semantica delle tuple che rappresentano le sostanze chimiche
- inglobare la regola di diffusione delle sostanza all'interno del comportamento di ogni centro

Agenti

Gli agenti del sistema sono le cellule che sono divisi in due tipi:

- cellule staminali (possono avere tre stati: staminali, progenitore, clone)
- cellule stromali

Entrambi i tipi di cellule depositano sostanze chimiche nella regione in cui si trovano quando non sono legate a cellule dell'altro tipo, in caso contrario si possono verificare diverse situazioni:

1. Se una cellula è vicina a cellule dell'altro tipo assumiamo che sia prossima all'accoppiamento; se una cellula staminale si lega con una cellula stromale la prima è in uno stato di riposo e la seconda non può secernere sostanze chimiche.
2. Una cellula quando attua una secrezione cerca di riempire tutta la regione in cui si trova con la massima concentrazione della particolare sostanza chimica.
3. La secrezione di una cellula stromale consiste in una ulteriore cellula dello stesso tipo che non si può legare con cellule staminali.
4. La secrezione di una cellula staminale consiste in una cellula di identico tipo che non si può legare con cellule stromali.

Le cellule hanno due capacità:

- rilevare la concentrazione di sostanze chimiche all'interno della loro regione
- rilevare quale tipo di cellule si trovano vicino a lei

Sono presenti in questo modello tre regole fondamentali per i movimenti delle cellule:

- le cellule progenitore sono respinte dalle cellule stromali
- durante la divisione delle cellule staminali queste sono respinte dalle sostanze chimiche staminali
- quando le cellule staminali sono in fase di crescita vengono attaccate dalle cellule stromali

La descrizione degli agenti introduce nuove specifiche da modellare per quanto riguarda ambiente ed interazione:

- dato che gli agenti devono essere in grado di rilevare la concentrazione di sostanze chimiche della regione in cui si trovano dobbiamo rendere disponibile questo attributo e lo possiamo fare producendo, come risultato del calcolo della diffusione delle sostanze chimiche, una tupla unica per ogni regione che viene aggiornata ogni volta
- quando una cellula secerne sostanze chimiche inserisce nuove tuple di “tipo” sostanza chimica all’interno della regione con la massima concentrazione possibile e visto il tentativo di riempire tutta la regione dovrà essere settato per ognuna di queste un quantitativo massimo di sostanze chimiche, rappresentato dal numero totale di tuple di tipo sostanza chimica

Un ulteriore aspetto da modellare è il fatto che ogni cellula deve conoscere quali e quanti sono i suoi vicini. Per supportare questo, tramite centri di tuple, potremmo fare in modo che ogni cellula inserisca all’interno della regione in cui si trova la propria posizione, magari sotto forma di coordinate, in modo che il centro possa consumarla e verificare quanti e quali sono i vicini di ogni singola cellula. A questo punto il centro ha la possibilità di inserire tuple dedicate per ogni cellula, definendone il vicinato.

Attuando queste specifiche sarebbe possibile modellare l’ambiente, l’interazione, la coordinazione e alcuni comportamenti degli agenti utilizzando, come mezzi di coordinazione, i centri di tuple.

Capitolo 4

Conclusioni e possibili sviluppi

In questo approfondimento si è cercato di analizzare l'utilità dei mezzi di coordinazione per la modellazione e la simulazione di sistemi complessi. Grazie a questi mezzi è possibile modellare:

- la topologia dell'ambiente
- il comportamento dell'ambiente
- le interazioni fra entità attive del sistema, modellate tramite agenti, e l'ambiente

Si evince, quindi, che i mezzi di coordinazione, ed in particolare i centri di tuple, possono rivelarsi veramente utili per effettuare simulazioni basate su sistemi multi - agente (MAS). Potrebbero però inserire problemi di complessità, come nel caso in cui la topologia dell'ambiente fosse molto complessa; questo implicherebbe infatti l'utilizzo di un elevato numero di centri di tuple con annesse programmazioni di comportamento.

Per quanto riguarda gli sviluppi futuri si potrebbe procedere alla creazione di un tool che permetta di creare modelli ABM in cui la parte che riguarda l'ambiente e le interazioni siano basati su centri di tuple, integrando al suo interno:

- modellazione di agenti
- modellazione dell'ambiente tramite centri di tuple
- regolazione delle interazioni e del comportamento dell'ambiente tramite la programmazione dei centri di tuple

Bibliografia

- [1] A portal site for TuCSon
<http://apice.unibo.it/xwiki/bin/view/TuCSon/>.
- [2] P. Angelini. Simulazione software basata su agenti, January 2006. Thesis.
- [3] M. Casadei and A. Omicini. *ReSpecT Guide*, 2008.
- [4] M. d’Inverno and J. Prophet. Multidisciplinary investigation into adult stem cell behavior. In C. Priami, E. Merelli, P. P. Gonzalez, and A. Omicini, editors, *Transactions on Computational Systems Biology III*, volume 3737, pages 49–64, 2005.
- [5] M. d’Inverno and R. Saunders. Agent-based modelling of stem cell self-organisation in a niche. In *Engineering Self-Organising Systems*, pages 52–68, 2004.
- [6] F. Klügl and E. Norling. Agent-based Simulation: Social Science Simulation and Beyond. Technical report, The Eighth European Agent Systems Summer School (EASSS 2006), Annecy, July 2006.
- [7] F. Klügl, C. Oechslein, F. Puppe, and A. Dornhaus. Multi-agent modelling in comparison to standard modelling. *Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, pages 105–110, 2002.
- [8] C. Macal and M. North. Tutorial on agent-based modeling and simulation part 2: How to model with agents. *Winter Simulation Conference*, 0:73–83, 2006.
- [9] C. M. Macal and M. J. North. Agent-based modeling and simulation: Abms examples. In *WSC ’08: Proceedings of the 40th Conference on Winter Simulation*, pages 101–112. Winter Simulation Conference, 2008.
- [10] S. Montagna. Modelling and simulating in systems biology: An approach based on multi-agent systems, January 2006. Master Thesis.

- [11] S. Montagna and A. Omicini. Simulation & multi-agent systems an introduction, 2008-2009. Presentation.
- [12] S. Montagna, A. Ricci, and A. Omicini. A&a for modelling and engineering simulations in systems biology. *IJAOSE*, 2(2):222–245, 2008.
- [13] E. Nardini and A. Omicini. Coordination in multi-agent systems, November 2008. Presentation.
- [14] A. Ricci and E. Denti. *TuCSon Guide*, 2006.
- [15] A. Vallorani and O. Riganelli. Luce + tucson.