

Corso di Laurea in Ingegneria e Scienze Informatiche

Slime-Mold Aggregation

Tesi di laurea in:
PROGRAMMAZIONE AD OGGETTI

Relatore

Prof. Mirko Viroli

Candidato

Lorenzo Tosi

Correlatore

Dott. Gianluca Aguzzi

Abstract

Max 2000 characters, strict.

Optional. Max a few lines.

Acknowledgements

Optional. Max 1 page.

Indice

Abstract	iii
1 Introduzione	1
2 Contesto	3
2.1 Alchemist	3
2.1.1 Il meta-modello	4
2.2 Simulazione di riferimento	7
2.3 Java e la programmazione ad oggetti	7
3 Design	9
3.1 Requisiti	9
3.2 Layer	9
4 Implementazione	11
	13
Bibliografia	13

Elenco delle figure

2.1	Rappresentazione di una reazione in Alchemist	5
2.2	Il modello di alchemist	6
3.1	Struttura PheromoneLayer	10

List of Listings

LIST OF LISTINGS

Capitolo 1

Introduzione

In natura, l'aggregazione delle cellule di muffa mucillaginosa (detti anche funghi mucilluginosi o, in inglese, slime-mold) rappresenta comportamento in cui entità individuali interagiscono tra di loro per formare strutture complesse e funzionali. Lo slime mold, o muffa mucillaginosa è un organismo unicellulare ma, talvolta, può trovarsi anche ad agire come un organismo multicellulare. Pur non essendo un fungo è spesso classificato nella stessa categoria per via delle sue caratteristiche affini a quelle di questi organismi. La particolarità principale della muffa mucillaginosa è che può muoversi come un organismo unicellulare o multicellulare a seconda delle diverse condizioni ambientali in cui si trova. L'habitat principale di questi organismi è il terreno umido, dove di solito si nutrono di foglie morte o di tronchi di alberi in putrefazione. Quando trova una fonte di cibo, lo slime-mold si aggrega, formando una massa citoplasmatica detta plasmodio, composta da un grand numero cellule. Inoltre, questa massa può muoversi, "navigando" attraverso il terreno in cerca di cibo. Quando le risorse alimentari scarseggiano o l'ambiente diventa meno ospitale, lo slime mold può assumere forme diverse: può formare spore, resistenti per sopravvivere in condizioni avverse, oppure aggregarsi insieme ad altri individui simili per formare una struttura multicellulare che si comporta come un'unica entità, condividendone di conseguenza risorse e compiti.

Perché è interessante simularlo?

L'obiettivo di questa tesi è esplorare il fenomeno dell'aggregazione di questi organismi, sviluppando un sistema software che si interfacci ed utilizzi a pieno

tutti gli elementi chiave del simulatore Alchemist. Quest'ultimo, infatti, permette di riprodurre eventi appartenenti a domini estremamente differenti tra loro, come simulazioni chimiche o il comportamento di pedoni in diverse situazioni.

Structure of the Thesis

Lorenzo Tosi: At the end, describe the structure of the paper

Capitolo 2

Contesto

In questo capitolo vengono spiegate le tecnologie adottate

2.1 Alchemist

Alchemist [PMV13] è un simulatore DES (Discrete Event System) che estende il modello computazionale base delle reazioni chimiche in modo tale da favorirne l'applicabilità a situazioni complesse, pur mantenendo elevate prestazioni. In particolare, Alchemist si fonda su una versione ottimizzata dell'algoritmo di Gillespie[Gil77] chiamata Next Reaction Method[GB00], esteso in modo tale da poter lavorare con un ambiente mobile e dinamico dove sia possibile aggiungere o rimuovere reazioni, dati e connessioni topologiche. Le applicazioni già implementate sono varie e comprendono, ad esempio, simulazioni di reazioni biochimiche e movimento di pedoni. Il punto di forza del sistema è il meta-modello estremamente astratto, la cui effettiva realizzazione è demandata alle "incarnazioni", le quali rappresentano l'implementazione vera e propria delle diverse categorie di simulazioni eseguibili all'interno. Attualmente troviamo 4 incarnazioni:

- Protelis
- SAPERE
- Biochemistry
- Scafì

2.1.1 Il meta-modello

Come accennato in precedenza, il meta-modello è uno dei punti di forza maggiori del simulatore. Per meta-modello si intende un tipo di paradigma che descrive la struttura, le regole e le relazioni che i modelli di dati devono seguire all'interno di un sistema. Rappresenta in modo astratto i concetti e le relazioni all'interno del dominio di interesse e stabilisce i vincoli e le convenzioni che tutti i modelli devono usare. Dunque, tutte le incarnazioni presenti presentano le stesse entità base. Poichè Alchemist è sviluppato partendo da un'ispirazione orientata sulla chimica/biochimica, le entità presentano nomi riconducibili a quei mondi. Infatti troviamo:

- **Molecole** (molecule): il nome di un dato, concettualmente può essere interpretato come il nome di una variabile.
- **Concentrazioni** (concentration): il valore associato alla molecola.
- **Nodi** (node): un “contenitore” di molecole e reazioni.
- **Ambiente** (environment): l'astrazione dello spazio; è un “contenitore” di nodi e svolge i seguenti compiti:
 - Restituire la posizione dei nodi.
 - Restituire la distanza tra due nodi.
 - Supportare il movimento dei nodi, se presente.
- **Regola di collegamento** (linking rule): una funzione relativa allo stato corrente dell'ambiente che associa ad ogni nodo un vicinato.
- **Vicinato** (neighborhood): un entità composta da un nodo centrale e un insieme di nodi vicini.
- **Reazione** (reaction): un qualsiasi evento che provoca un cambiamento dello stato dell'ambiente. È definita da una lista di condizioni, una o più lista di azioni e una distribuzione temporale. La frequenza con la quale avviene una reazione dipende da:

- Un parametro statico “rate”.
 - Il valore di ogni condizione.
 - Una “rate equation”, ovvero una equazione che combina il parametro statico (rate) con i valori delle condizioni, restituendo un “instantaneous rate”.
 - Una distribuzione temporale.
- **Condizione** (condition): una funzione che, dato lo stato attuale dell’ambiente (environment), restituisce un booleano ed un numero. Se il booleano è falso la reazione non può avvenire. In caso contrario, invece, avviene. Il numero può invece influenzare la velocità della reazione a seconda della reazione e della distribuzione temporale.
 - **Azione** (action): un cambiamento nell’ambiente.

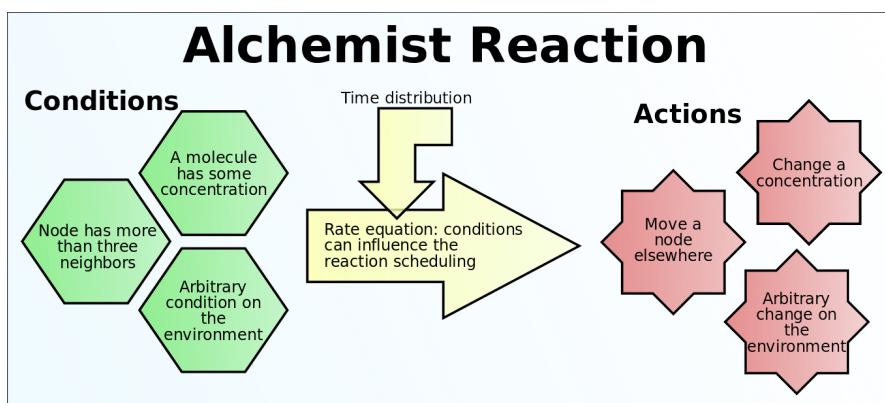


Figura 2.1: Rappresentazione di una reazione in Alchemist

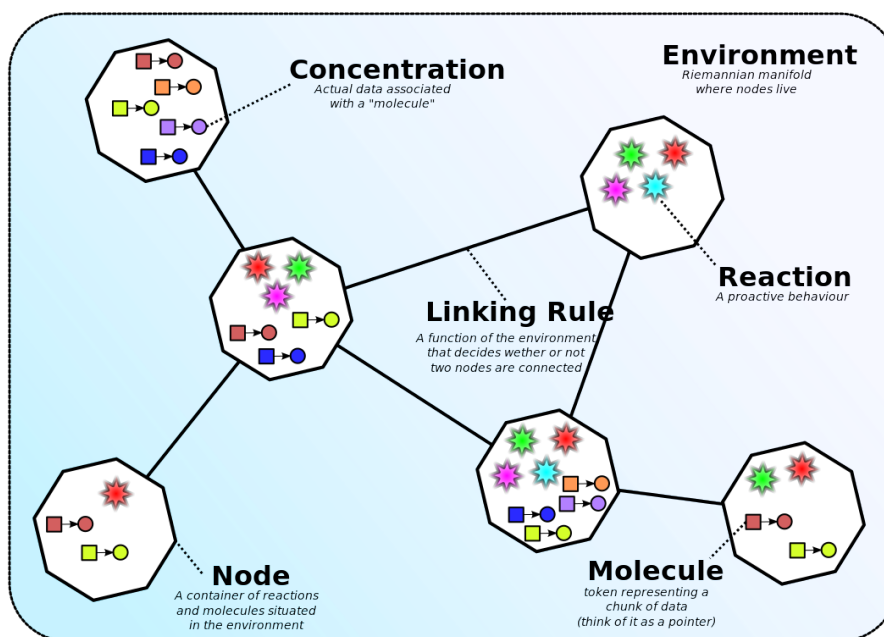


Figura 2.2: Il modello di alchemist

2.2 Simulazione di riferimento

La simulazione utilizzata come riferimento per questo progetto di tesi è presente nella libreria di modelli di NetLogo[Wil97]. Il modello di riferimento è "Slime" [Wil97] e per simulare l'aggregazione di tanti singoli organismi in un gruppo si fa riferimento al comportamento delle tartarughe. Quest'ultime si muovono in uno spazio a griglia e durante il loro movimento rilasciano una particolare molecola chiamata "feromone" che si deposita in una posizione precisa, chiamata "patch". La tartaruga per muoversi "annusa" davanti a se, ovvero percepisce se nelle patch vicine è presente del "feromone". Se questo è abbastanza alto, la tartaruga si sposterà nella posizione "annusata", in caso contrario la tartaruga si muoverà randomicamente. Durante tutto ciò, le patches diffonderanno del "feromone" alle varie posizioni vicine e con il passare del tempo il "feromone" evaporerà dalla griglia.

2.3 Java e la programmazione ad oggetti

Capitolo 3

Design

3.1 Requisiti

Analizzando il modello presente su NetLogo[Wil97], possiamo individuare le caratteristiche e i requisiti che la simulazione dovrà avere:

- Entità "vive", che si muovono e depositino il feromone.
- Un ambiente che gestisca la presenza del feromone; in particolare dovrà:
 - Permettere il depositarsi della sostanza.
 - Diffondere la sostanza.
 - Evaporare la sostanza.
- Le entità dovranno avere un concetto di direzione.
- Il movimento deve seguire delle regole ben precise.

3.2 Layer

Si pensi alla simulazione come se fosse un mondo, una città, complessa e ricca di informazioni. E' di interesse capire il livello di inquinamento di questa città, qualcosa di invisibile all'occhio umano ma comunque presente nell'ambiente, oppure la temperatura nelle varie aree cittadine. Si ha bisogno di "inserire" nel nostro

ambiente degli "strati" invisibili che hanno il compito di raccogliere queste informazioni. E' possibile in Alchemist definire questi "strati" di dati, chiamati Layer. Nel contesto di questo progetto è stato necessario l'utilizzo di un Layer che avesse la funzione di "rete di raccolta" dei feromoni che, nella simulazione di riferimento 2.2, venivano rilasciati dalle tartarughe nelle varie posizioni. Un aspetto di particolare rilevanza è stato il processo decisionale relativo alla gestione delle posizioni collegate al deposito del feromone. In alchemist non è presente di "spazio", le posizioni sono puntiformi

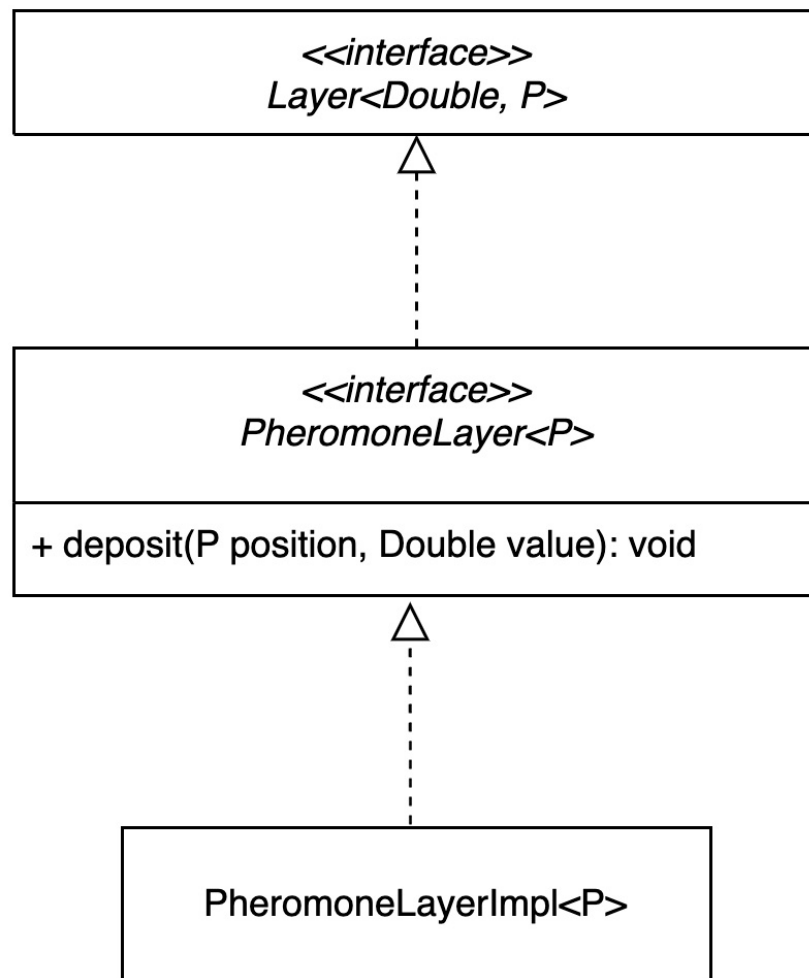


Figura 3.1: Struttura PheromoneLayer

Capitolo 4

Implementazione

Bibliografia

- [GB00] MA Gibson and J Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *Journal of Physical Chemistry A*, 104(9):1876–1889, 2000.
- [Gil77] DT Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
- [PMV13] D Pianini, S Montagna, and M Viroli. Chemical-oriented simulation of computational systems with alchemist. *Journal of Simulation*, 7(3):202–215, August 2013.
- [Wil97] Uri Wilensky. Netlogo slime model. <http://ccl.northwestern.edu/netlogo/models/Slime>, 1997.
- [Wil99] Uri Wilensky. Netlogo. <http://ccl.northwestern.edu/netlogo/>, 1999.