

Analisi e sviluppo di un componente Java per la generazione automatica di modelli NetLogo

Candidato: Aurel Pjetri

Relatore: Prof. Enrico Vicario

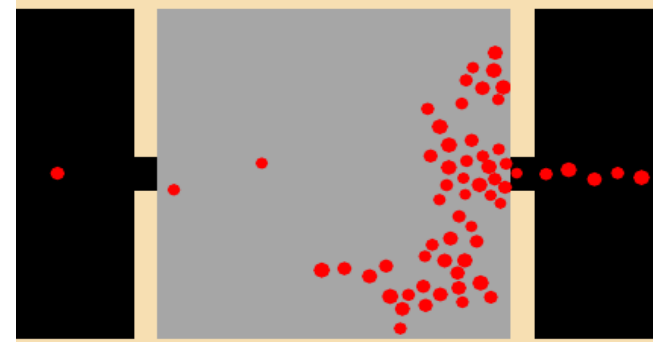
Co-Relatore: Dott. Sandro Mehic

Simulazione delle folle

Gli approcci più diffusi si dividono in due categorie:

- Microscopico:

rappresenta ogni persona come un agente, particella o macchina a stati finiti.



Simulazione di evacuazione tramite *social force model*

- Macroscopico:

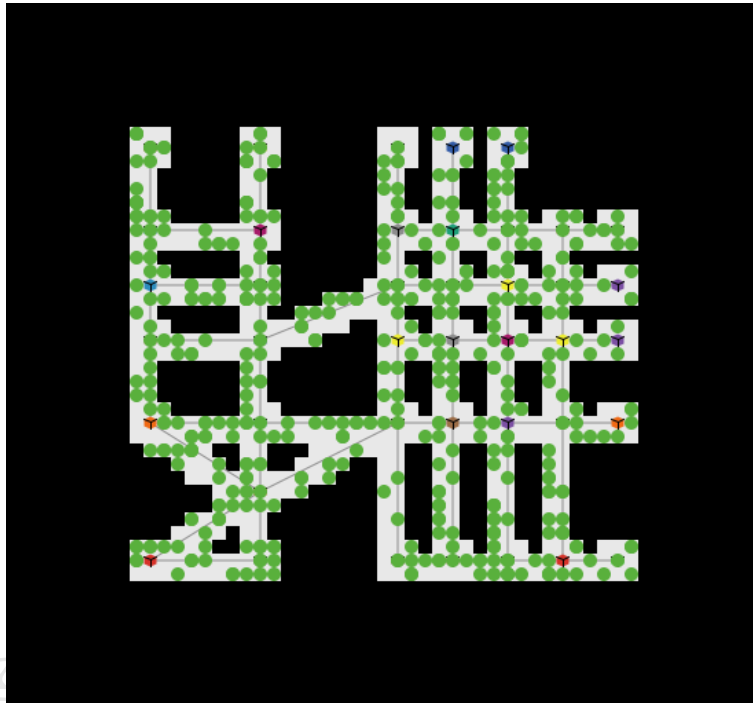
studia caratteristiche macroscopiche come densità media e velocità della folla.



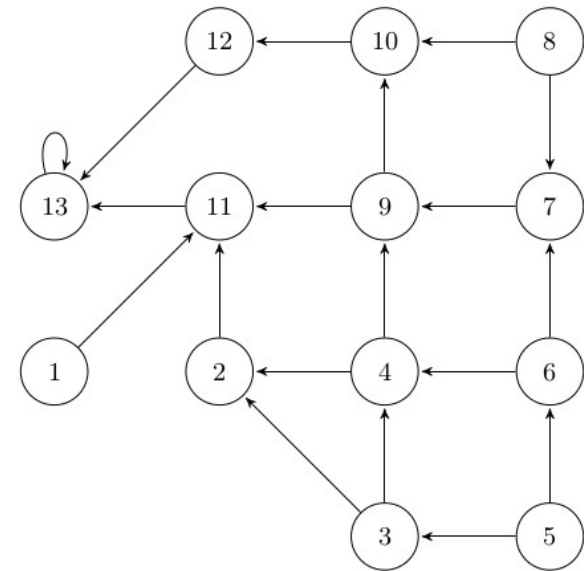
Simulazione tramite approccio *fluid dynamics*

Approccio gerarchico

Combina i due approcci per dare una soluzione analitica indipendente dal numero di agenti.



Microscopico

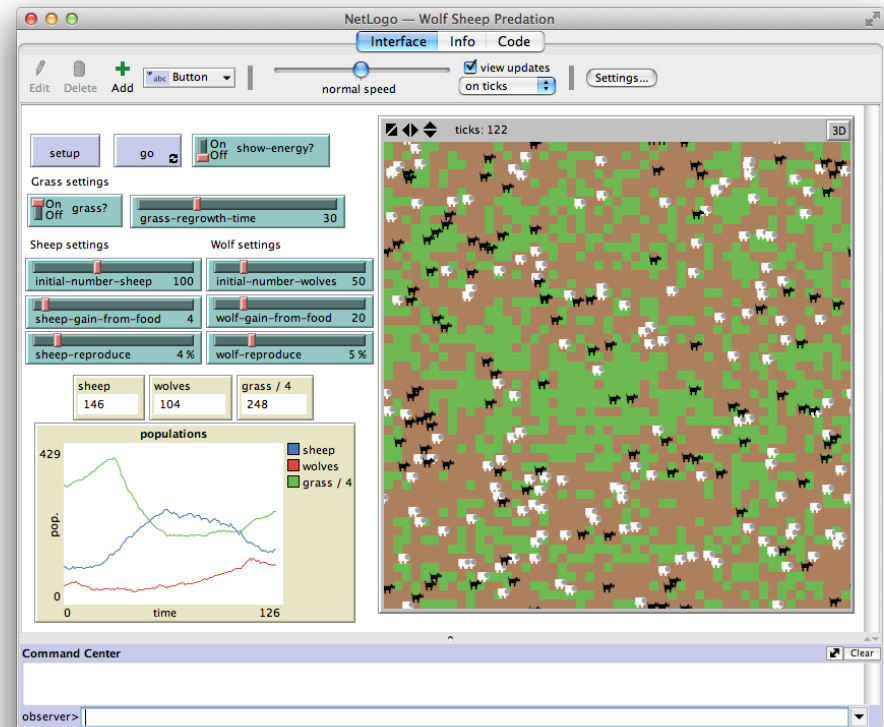


Macroscopico

NetLogo

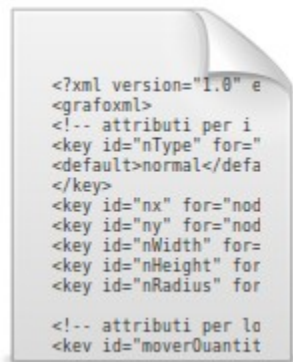
Ambiente di modellazione di sistemi complessi e
linguaggio di programmazione agent-based

- Gli **agenti** hanno uno **stato** proprio e agiscono in modo **indipendente** e **concorrenziale**.
- Il mondo è suddiviso in **patches** interamente programmabili.
- Ampiamente accettato dalla comunità.
- È lo strumento più diffuso per la modellazione agent-based.
- Linguaggio di programmazione **procedurale** poco flessibile e di difficile gestione.

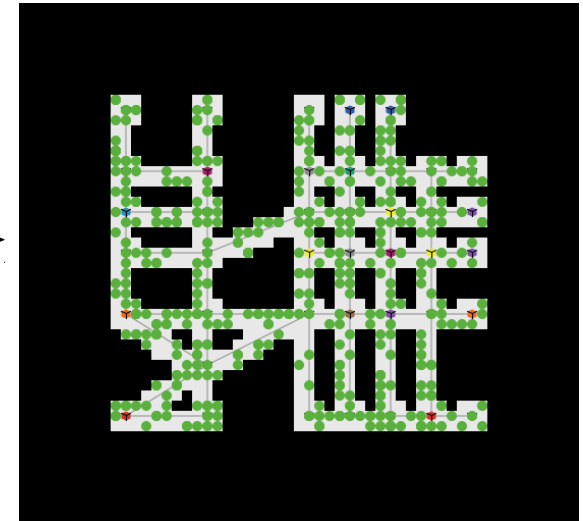
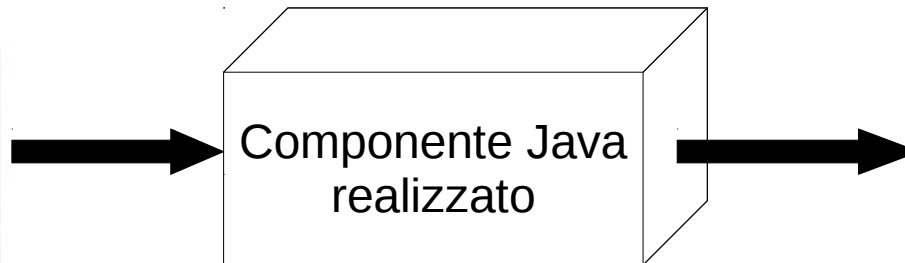


Obiettivo

Automatizzare il processo di scrittura del codice NetLogo
che esegue le simulazioni

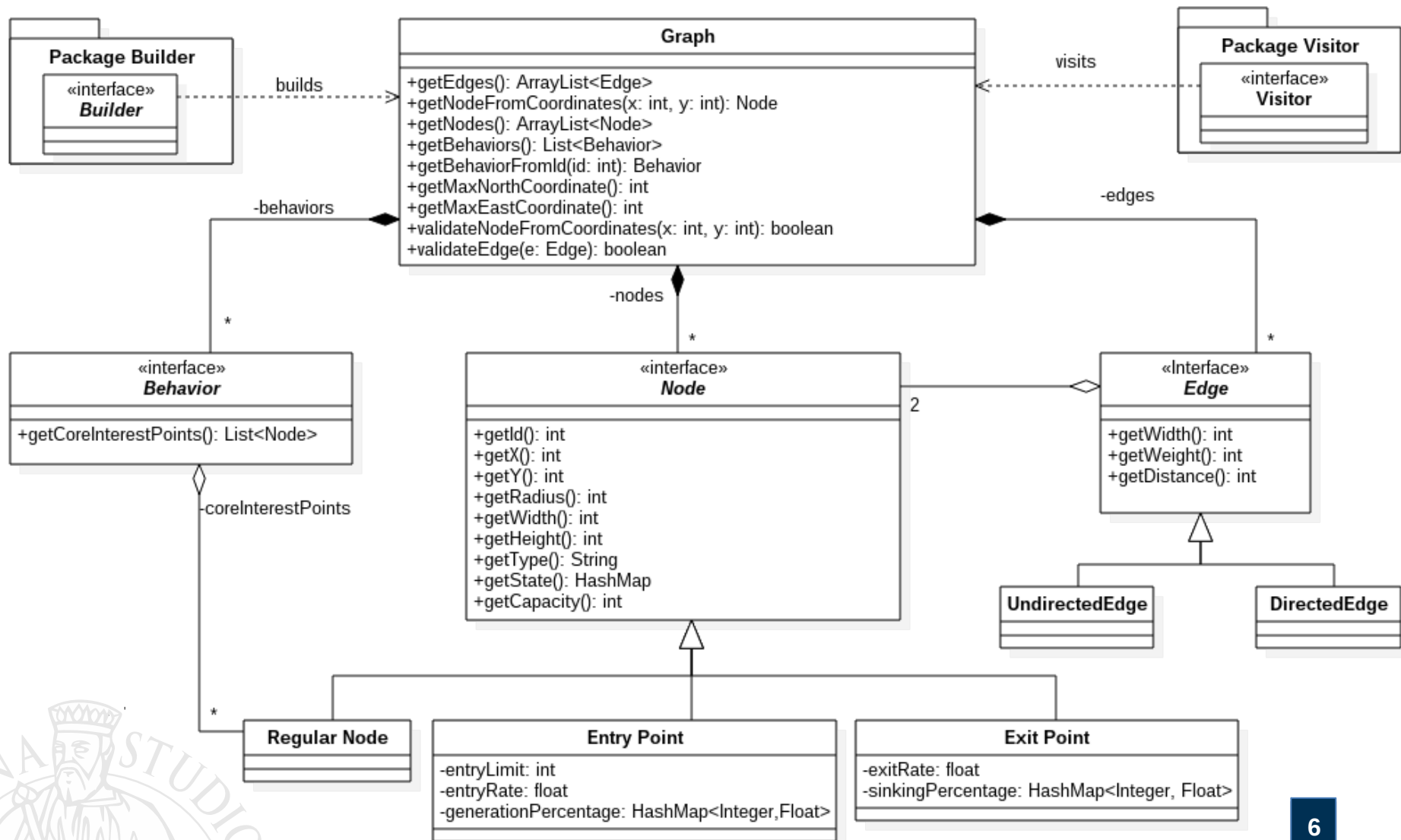


Documento
XML



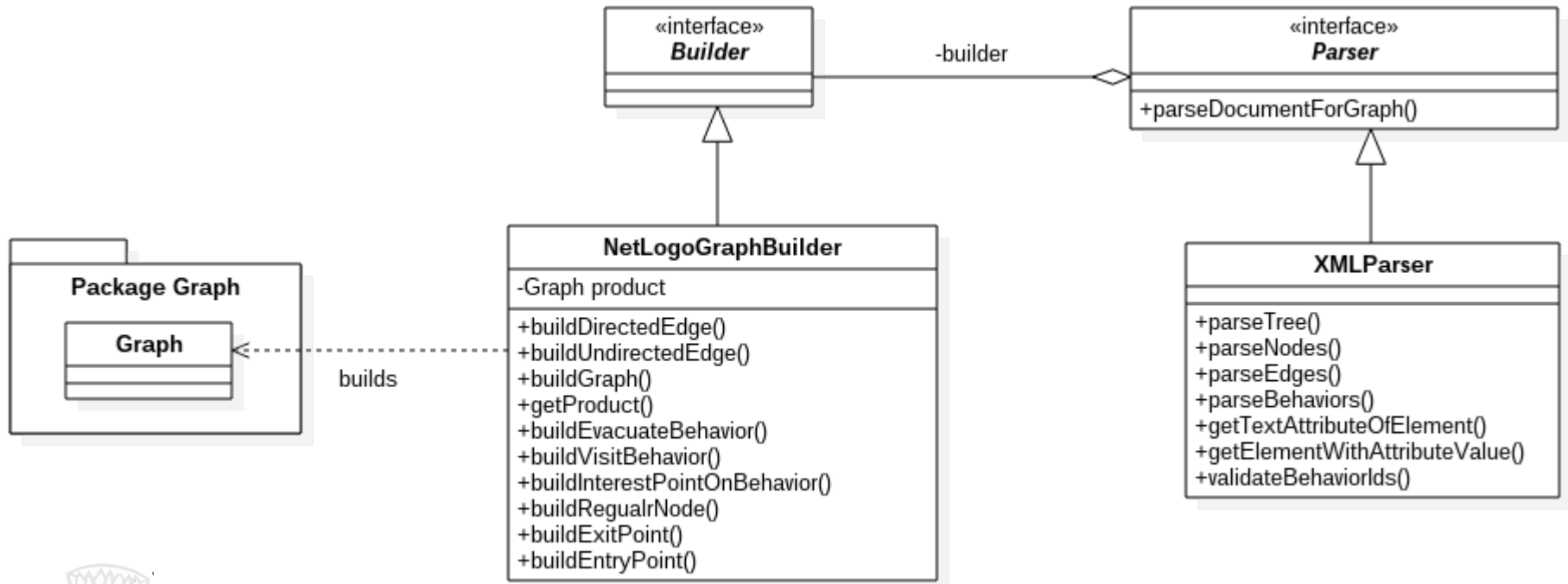
Modello NetLogo

Logica di dominio



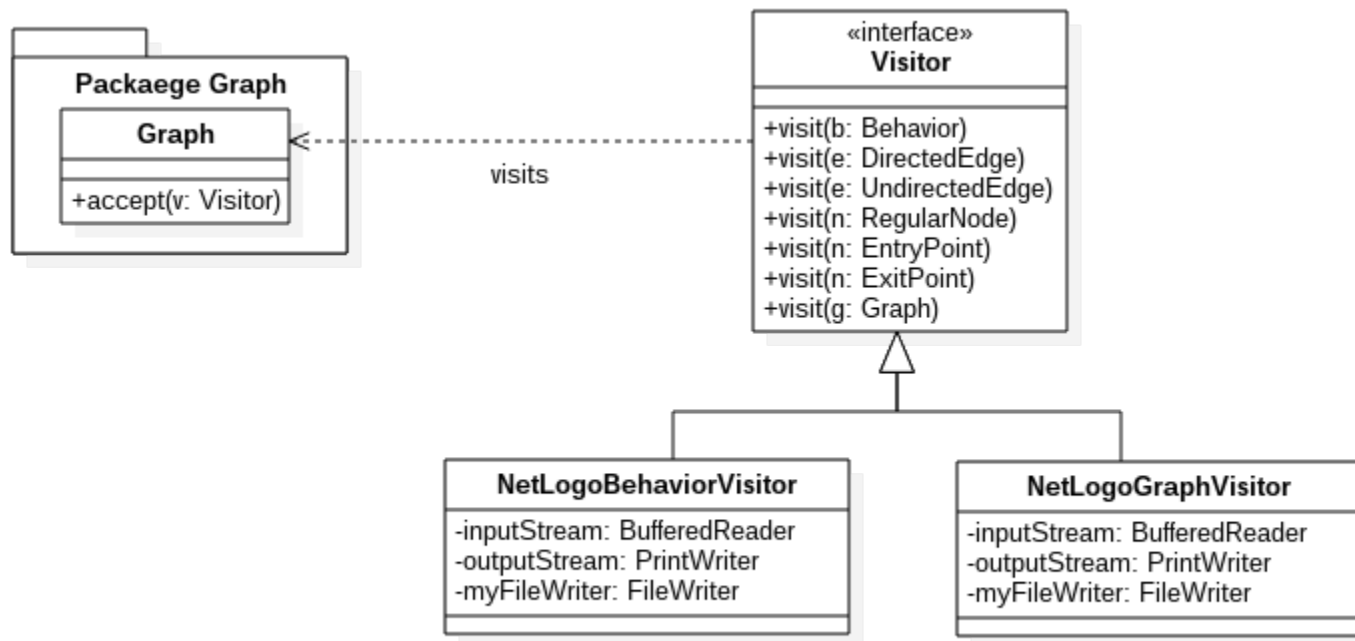
Builder

- **XMLParser** analizza il documento ed estrae le informazioni di interesse.
- **NetLogoGraphBuilder** costruisce e compone gli oggetti della struttura.

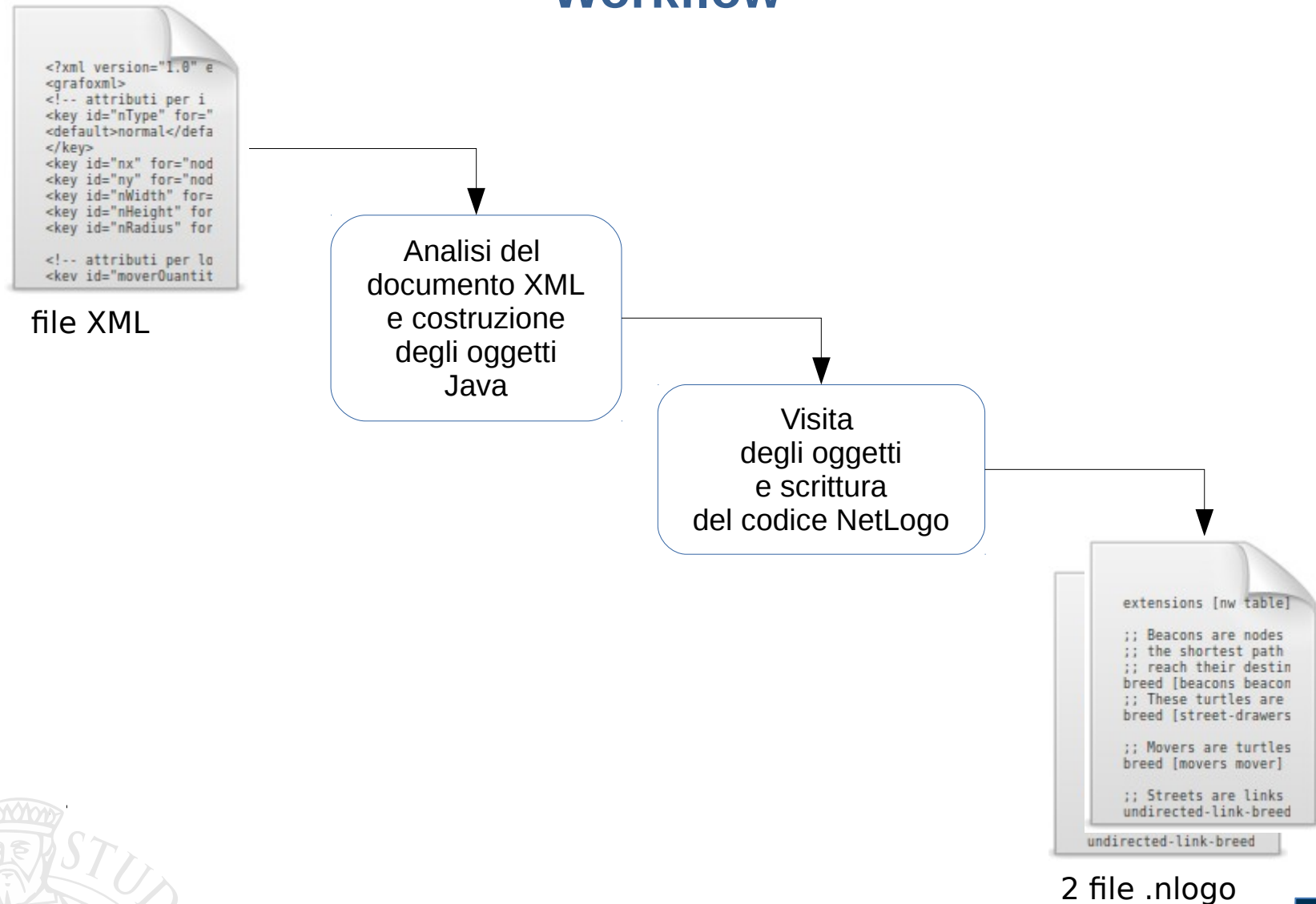


Visitor

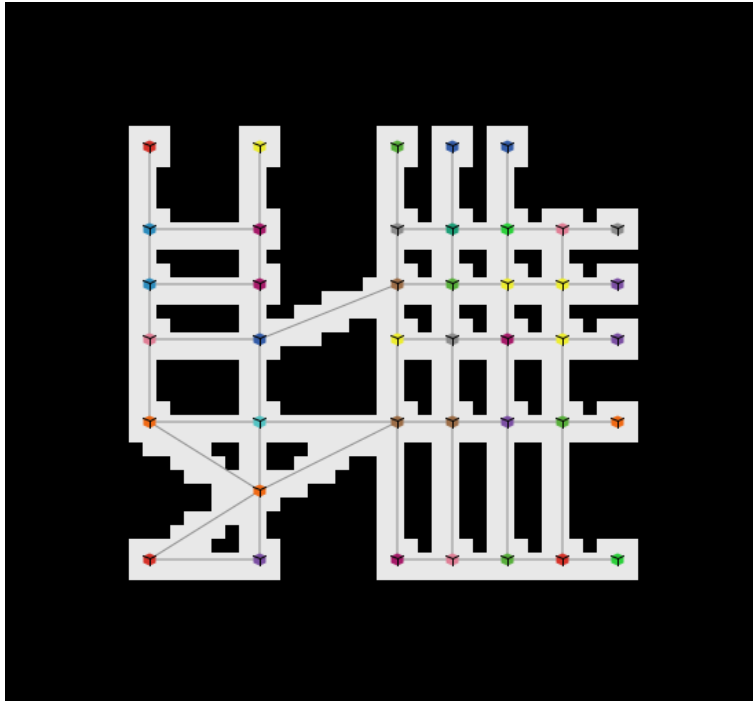
- Usano le interfacce degli oggetti della struttura per estrarre le informazioni.
- Scrivono il codice NetLogo.



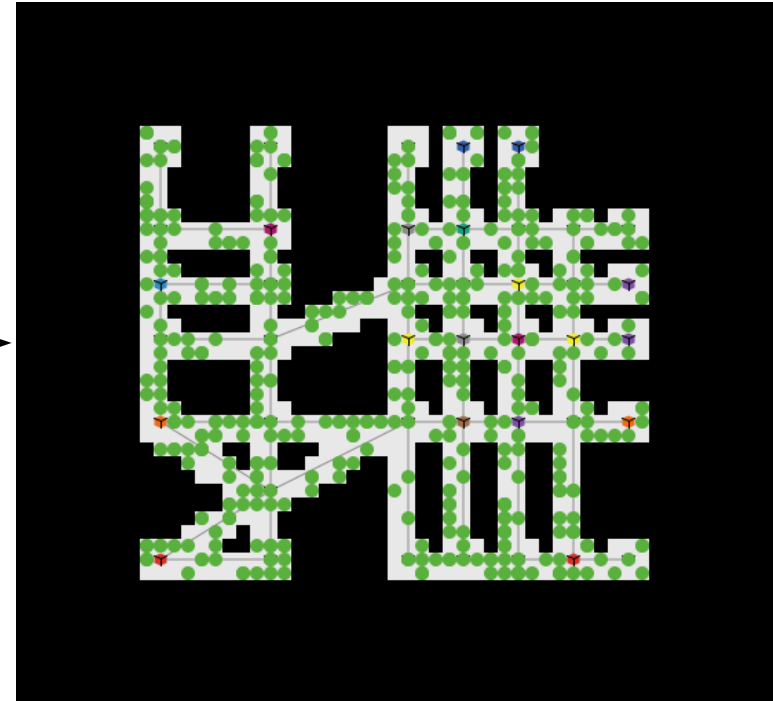
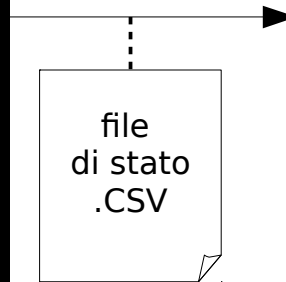
Workflow



Modelli NetLogo



Modellazione dell'ambiente



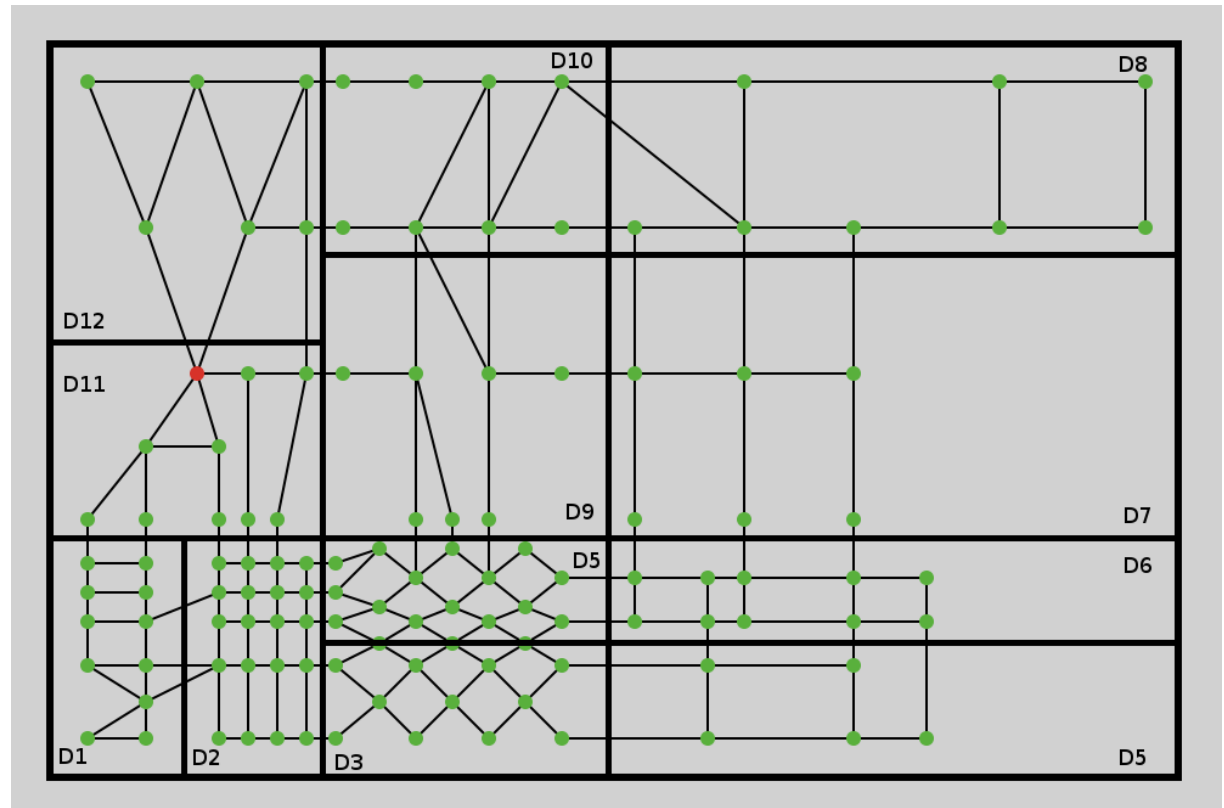
Modellazione del movimento degli attori e raccolta dei dati

Esperimenti

Simuliamo l'evacuazione su un modello che rappresenta un tessuto urbano

Per gli scenari A e D:

- Due modalità di simulazione: con **densità costante** e **transitoria**.
- Tre stati iniziali diversi: **alta**, **media** e **bassa densità** di **affollamento**.



Risultati

- Nello scenario con maggiore frammentazione il tempo di simulazione è minore
- In caso di modifiche possiamo simulare solo le regioni interessate
- Nel modello monolitico impiego sempre lo stesso tempo

Scenario	Densità costante	Transitorio
Monolitico		9:19:40
A	21:33:36	0:21:51
D	09:09:02	0:23:13

Tempi totali di simulazione

Regione	Densità costante	Transitorio
A1	3:16:25	0:03:06
A2	2:13:11	0:02:40
A3	5:12:48	0:03:40
A4	6:20:39	0:06:08
A5	2:36:25	0:03:29
A6	1:46:20	0:02:46

Tempi di simulazione scenario A

Regione	Densità costante	Transitorio
D1	0:27:26	0:01:56
D2	0:29:15	0:02:03
D3	0:49:46	0:01:58
D4	0:29:48	0:02:00
D5	1:12:14	0:01:51
D6	1:04:18	0:01:40
D7	1:14:06	0:01:58
D8	1:04:41	0:02:07
D9	1:08:36	0:01:58
D10	0:53:40	0:02:06
D11	0:06:25	0:01:43
D12	0:08:47	0:01:48

Tempi di simulazione scenario D

Risultati

- Con la modalità a densità costante si raccolgono molti più dati
- I dati sono più realistici rispetto al modello transitorio

Scenario	Densità costante	Transitorio
Monolitico		9949
A	395650	15998
D	408295	15862

Numero totale di dati raccolti

Scenario	Densità costante	Transitorio
Monolitico		9:19:40
A	21:33:36	0:21:51
D	4:49:27	0:23:13

Tempi totali di simulazione

Regione	Densità costante	Transitorio
A1	51325	1693
A2	55661	1528
A3	69095	2423
A4	71533	4057
A5	62510	2582
A6	85526	3715

Numero di dati estratti dallo scenario A

Regione	Densità costante	Transitorio
D1	23463	803
D2	27834	803
D3	46190	803
D4	35894	803
D5	46454	1100
D6	43967	1100
D7	25052	1900
D8	25902	1900
D9	35990	1450
D10	32826	1450
D11	43770	1450
D12	20953	2300

Numero di dati estratti dallo scenario D

Conclusioni e sviluppi

Risultati conseguiti:

- Componente Java che automatizza la scrittura del codice NetLogo.
- Utilizzo del componente nelle attività di ricerca del Software Science and Technology Laboratory (STLAB), Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze.

Sviluppi abilitati dai risultati conseguiti:

- Estensione della logica di rappresentazione dei comportamenti degli agenti includendo anche aspetti **sociali** e studio dell'impatto che questo ha sulla correttezza delle simulazioni.
- Implementazione di un'interfaccia che faciliti l'utilizzo.
- Implementazione di nuovi parsers per formati alternativi all'XML.