

Simulation Informatique

La simulation consiste à travers l'exécution d'un ou de programmes de simuler un phénomène complexe (exemple choc d'un véhicule, ...).

Catégorie de simulation

La simulation continue, le système est décrit par des équations différentielles

(au départ sur des calculateurs analogiques)

La simulation discrète – le système évolue en fonction d'événement avec 2 cas :

- asynchrone ou time-slicing
- synchrone ou event-sequencing

La simulation par agents – le système évolue à travers les relations entre les agents

Quelques logiciels de simulation numérique

Fluent : logiciel américain de simulation des écoulements fluides,

Code Mascaret : logiciel libre de simulation numérique d'écoulements unidimensionnels, (problèmes maritime, ...)

Plate-forme Salomé : plate-forme d'intégration libre et open source pour la simulation numérique

...

Quelques logiciels de simulation à événements discrets

Tortuga : framework de simulation à
événement discret en Java

SimPy : framework de simulation à événement
discret en Python

Le paradigme Système Multi-Agent

Bref Historique

L'approche se situe dans le cadre de l'intelligence artificielle distribuée (IAD)

Dans les années 1970 le modèle d'acteur est un modèle mathématique qui définit les acteurs comme les seules entités nécessaires au calcul concurrent.

En réponse à un message, un acteur effectue un traitement, crée d'autres acteurs et envoie d'autres messages.

Le paradigme Système Multi-Agent

Bref Historique

L'approche se situe dans le cadre de l'intelligence artificielle distribuée (IAD)

Dans les années 1975 les architectures Blackboard ont pour objectif la résolution de problèmes. L'idée est d'avoir une base de connaissance commune le « Blackboard – Tableau Noir » qui évolue par différents modules

1975 Hearsay II – Un système de reconnaissance vocale de Carnegie-Mellon

Lee D. Erman and Victor R. Lesser. 1975. A multi-level organization for problem solving using many, diverse, cooperating sources of knowledge. In Proceedings of the 4th international joint conference on Artificial intelligence - Volume 1 (IJCAI'75), Vol. 1. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 483-490

1983 PROTEAN – Recherche de la forme 3D de protéine

Hayes-Roth, B. A blackboard architecture for control. Artificial Intelligence, 1985, 26, 251-321

Le paradigme Système Multi-Agent

Caractéristiques :

- Le système est constitué d'agents et d'un environnement
- L'agent est autonome (au moins partiellement)
- Les agents ont une vue locale du système
- Le système est décentralisé

Le paradigme Système Multi-Agent

J. Ferber 1991 :

- entité réelle ou virtuelle plongée dans un environnement sur lequel elle est capable d'agir
- qui dispose d'une capacité de perception et de représentation partielle de cet environnement
- qui peut communiquer avec d'autres agents

- qui est mue par un ensemble de tendances (objectifs individuels, fonctions de satisfaction, de survie)
- qui possède un comportement autonome tendant à satisfaire ses objectifs, conséquence de ses observations, de sa connaissance, et des interactions qu'elle entretient avec les autres agents
- qui est capable éventuellement de se reproduire

Agents Matériels / Agents Logiciels

Agents Matériels :

- fait de processeurs et de capteurs
- interaction avec le monde réel
- domaine de la robotique

<http://www.ai.mit.edu/projects/genghis/genghis.html>

http://robofoot.polymtl.ca/francais/medias_videos.shtml

http://www.youtube.com/user/AldebaranRobotics#p/u/25/Sx_-witfXU8

Agents Logiciels :

- L'agent est un « code »
- Il est dans un environnement modélisé
- L'environnement propose des moyens de communication
- Souvent il y a une forme de localisation de l'agent pour définir une vue locale

Automates cellulaires

Un automate cellulaire consiste en une grille régulière de « cellules » contenant chacune un « état » dans un ensemble.

L'état d'une cellule au temps $t+1$ est fonction de l'état au temps t d'un « voisinage »

Les mêmes règles sont appliquées simultanément à toutes les cellules de la grille, produisant une nouvelle « génération » de cellules dépendant entièrement de la génération précédente.

Automates cellulaires

Début des automates cellulaires dans les années 1940 (von Neuman)

Formalisme « mathématique »

De nombreux résultats

(Le jeu de la vie, copieur et constructeur universel, ...)

Un système multi-agent peut souvent être utilisé pour simuler des automates cellulaires

Types d'Agent

- Agent réactif:
 - comportement de type stimulus/réponse
 - pas de mémoire de son histoire, ni de but explicite
 - pas de représentation explicite de l'environnement
- Généralement utilisés en grand nombre d'agents (>100), homogènes à grain fin
- On compte sur l'émergence de comportement

Les agents réactifs

- Exemples :
- alimentation de Fourmis
- Un agent = Une fourmi
 - Suit les phéromones
 - Dépose des phéromones sur leur retour si elles sont chargées de nourriture
 - Les phéromones s'évaporent avec le temps

Les agents proactifs

- Agents
 - Chargés d'un but
 - Généralement associés à une compétence
 - Capables de planifier leurs actions à partir d'une situation présente
- Nécessite
 - Une représentation de l'environnement, (y compris les autres agents)
 - Une communication avec les autres agents
- Exemple : Système Proie/Prédateur

Les agents cognitifs

- représentation explicite de l'environnement et des autres agents
- peut tenir compte de son passé et dispose d'un but explicite
- mode "social" d'organisation (planification, engagement)
- Généralement en petit nombre d'agents (10/20), hétérogènes à gros grain
- Les Interactions entre agents s'établissent en fonction des collaborations nécessaires à la résolution du problème

L'environnement

- Un agent est situé dans un environnement (spatio-temporel)
- Naturellement dynamique
- Évolutif de par les actions des agents
- Un agent à conscience (par sa perception) de sa présence dans celui-ci

L'environnement

- Espace partagé par tous les agents
- A la fois : Introduit des stimulations et/ou contraintes aux agents
- Stimulation pour les agents
 - Découverte de l'environnement : Boîtes de Ghengis
 - Perception de modifications
- Médium de l'interactions
 - Phéromones des fourmis

L'environnement

- Contraintes
 - Contraintes géographiques
 - Accessibilité à certaines informations
 - Connaissance partielle du monde à un instant
 - Possibilité d'interactions
 - Contraintes temporelles

L'environnement

- Environnement continu
 - Dispose d'une métrique (distance)
 - Localisation de l'agent dans l'environnement
- Environnement discontinus (réseaux)
 - Le réseaux peut être modifié
 - Ancrage (présence) de l'agent dans son environnement
 - Accès partiel d'un agent au contenu de l'environnement

- Environnements non structuré
 - Environnement contenant des agents et des objets et informations disponibles du système
 - Abstrait
 - Pas d'ancrage de l'agent
 - Ensemble d'objets qu'un agent peu percevoir et agir sur l'ensemble de l'environnement
 - Pas de modification possible de l'environnement
- Absence d'environnement
 - Agents uniquement communicants

Les interactions

- Système Multi-Agents : à partir de 3 agents (2 agents + un environnement)
- Les interactions sont nécessaire entre les agents
 - Echange de connaissances
 - Transmission de ses croyances
 - Coordination entre agents

- La communication de message se caractérise par
 - Le contenu du message
 - L'intention de l'expéditeur
 - Les conséquences du message dans l'environnement et les agents receveurs
 - Les agents impliqués dans la communication
 - Son mode (direct ou indirect)
- La dimension d'interprétation des messages dépasse la transmission de message au sens de la théorie de l'information

Les interactions/communication par l'environnement

- Communication indirecte :
 - Des traces sont laissées dans l'environnement par l'émetteur du message
 - Ces traces sont perçues par d'autres agents comme une modification de l'environnement
 - Généralement associée à des agents réactif

Exemple 1 : Fourmis

Les phéromones sont des traces dans l'environnement qui servent de message

Protocole de communication

- Envoi de messages pair à pair
- Deux agents en communication suivent un protocoles prédéfinis
 - Un protocole peut être modélisé par un réseau de Petri
- Exemple de protocoles existants
 - Résolution de conflits
 - Marchandage (Contract Net)
 - Coordination
 - Allocation de tâches

Actes de langages

- Actes de langages
 - Pour agents cognitifs
 - Utilisation d'une logique de représentation
- La communication : une action intentionnelle caractérisée par un type et plusieurs composantes
- Objectif d'aller vers des agents qui dialoguent

Les organisations

- Organisation de base consiste à prendre en compte des réseaux d'agents pour la réalisation d'une tâche commune
- Les organisations permettent de structurer à la conception
 - Les coordinations entre agents du système
 - Les interactions entre les agents

Types de coopération

- Confrontative cooperation : La tâche est réalisée par des agents différents les résultats sont ensuite fusionnés.
- Augmentative cooperation : La tâche est réalisée par des agents ayant les même compétences, mais travaillant sur des données complémentaires. Le résultat est obtenue comme un ensemble de résultats partiels.

Types de coopération

- Integrative cooperation : La tâche est décomposée en sous tâche dont la réalisation (coordonnée) est effectuée par des agents aux compétences spécifiques. Le résultat est obtenu à la finalisation de l'exécution.

Auto-organisation

- Auto-organisation : Un système est capable d'autoorganisation lorsque le développement des agents permet à un observateur d'observer une structure organisationnelle non programmée
- Les systèmes auto-organisés sont
 - Plus adaptatifs (et donc plus robustes)
 - Générateurs de solutions propres adéquates
- Les SMA permettent de mettre en œuvre ce genre de système et de les étudier

Autonomie

- Autonomie des agents
 - Entre eux, incluant les utilisateurs
 - Exemple des rovers Marsiens (Nécessité d'autonomie entre les réceptions/envois de messages)
- Décentralisation du contrôle
 - Métaphore des feux de circulations
 - Contrôle centrale
 - Contrôle décentralisé : 1 feu par carrefour
- Asynchronisme
- Interactions et organisations

Distribution

- Distribution physique
 - Prendre en compte la nature distribuée des problèmes rencontrés
 - Bénéficiaire de l'efficacité de traitements distribués
- Distribution de la compétence des agents
 - Faire mieux à plusieurs que seul
 - Robustesse
- Distribution des rôles et des buts :
 - Faire mieux grâce à l'hétérogénéité des agents

Hétérogénéité/Homogénéité

- Hétérogénéité : SMA constitué d'agents différents
 - Plutôt pour des agents cognitifs et en petite quantité
 - Avantages pour faciliter l'intégration d'agents non initialement prévu (Ouverture)
- Homogénéité : SMA constitué d'agents identiques
- Plutôt pour les agents réactifs et en grande quantité

Hétérogénéité/Homogénéité

- Hétérogénéité étendue
 - Agents physiques (Robots)
 - Intégration facilité des agents humains comme d'autres agents au sein du SMA

Hétérogénéité/Homogénéité

- Hétérogénéité étendue
 - Agents physiques (Robots)
 - Intégration facilité des agents humains comme d'autres agents au sein du SMA

Utilisateurs

- Les utilisateurs peuvent être intégrés au SMA
 - Considérés comme un nouvel agent
 - Permet d'inclure plusieurs utilisateur
 - Les interfaces servent à intégrer l'utilisateur dans l'environnement défini dans le SMA (Si l'environnement n'est pas physique)

Utilisateurs

- Agents Alter-ego
 - Un agent représente un utilisateur dans un environnement (généralement) virtuel
 - Personnalisation de l'agent qui trace le profil de l'utilisateur pour s'adapter au besoin de celui-ci
- Intelligibilité
 - SMA pour les simulations : les entités du système modélisés correspondent aux agents
 - SMA anthropomorphiques

Connaissances dans les SMA

- Approche blackboard : les données sont partagées et accessibles à tous les agents
- Approche cognitive : les données sont dans les agents eux mêmes
- Les contraintes d'accès aux données :
 - Contraintes d'environnement
 - Nécessité d'être « proche » de la donnée pour y accéder
- Contraintes d'interaction
 - Rétention d'information, mensonge, ...

Contrôle du système

- Délégation du contrôle au système
 - Les agents sont autonomes
 - L'utilisateur humain n'intervient que comme une partie du SMA
 - Il n'y a pas de contrôle global des traitements par l'utilisateur
- Comment garantir, un résultat ?
 - Utilisation de mécanisme de résolution de conflit
 - Etudes formelles des systèmes
« coopératifs »

Contrôle du système

- Vers la perte délibérée du contrôle :
 - Observation de phénomènes émergents
 - Conception d'agents adaptatifs, auto-réplicants
- Objectif mettre en place des systèmes adaptatifs
 - Prise en compte du dynamisme du contexte
 - Robustesse et tolérance à la panne

Positionnement

- Approche Objets :
 - Similaire
 - Définition de rôles,
 - Interactions,
 - Organisation
 - Différent
 - Autonomie

Positionnement

- Approche // - Thread :
 - Similaire
 - Distribution des traitements
 - Dynamisme
 - Autonomie
 - Différent
 - Haut niveaux : interaction, organisation
 - Intelligibilité

Positionnement

- Approche Composants :
 - Similaire
 - Distribution des traitements
 - Dynamiques des ressources
 - Autonomies des entités et décentralisation
 - Organisation (coopérations)
 - Différent
 - Situation dans un environnement

Exemples d'applications

- Les système simulés peuvent être très variés
 - Dépliement de protéine
 - Système immunitaire
 - Déplacement de cellules (AgentCell, T. Emonet)
 - Migration de cellules tumorale

AgentCell

<http://emonet.biology.yale.edu/agentcell/>

- Les cellules sont modéliser comme des agents
 - Rouges : Attirées par l'aspartate
 - Vertes : Non attirées par l'aspartate
- Une multitude d'agents sont disposer dans un environnement biologique simulés
 - L'environnement est muni d'un gradient d'aspartate
- Les interactions physiques entre les cellules (chocs) sont pris en compte

Coordination d'Ambulance

- Objectif : Faciliter la coordination de réseaux d'ambulances en situation d'urgence
- Agents
 - Traffic agents : Récupère les informations du « national traffic central », et redistribue les informations (broadcast) aux ambulances
 - Tracing agent : enregistre les actions passées d'une ambulance

Coordination d'Ambulance

- Agents
 - Ambulance team : Représente une ambulance, se charge de
 - négocier ses interventions
 - servir de moniteur à l'ambulance
 - Ambulance coordinator : Gère l'attribution des interventions aux ambulances

B. Lopez, B. Innocenti, S. Aciar, and I. Cuevas, A Multi-Agent System to Support Ambulance Coordination in Time-Critical Patient Treatment, ASAI, 2005

Plateforme de simulation

Plate-forme	Domaine	License	Langage de programmation	Système d'opération	Support d'utilisateurs	FIPA Compliant	Capacité SIG	Capacité 3D
Cormas (Common-pool Resources and Multi-Agent Systems)	Gestion des ressources naturelles, du développement rural et de l'écologie	Libre de modifier, mais de ne pas distribuer la version modifiée	Smalltalk (besoin VisualWorks pour exécuter)	Linux; Macintosh; Windows	Formation, des exemples; forum en ligne; e-mail aux développeurs, la documentation	Inconnu	Inconnu	Inconnu
JADE (Java Agent DEvelopment Framework)	Applications distribuées composée d'entités autonomes	LGPL version 2	Java	Toute Java Plateforme	FAQ; liste de courriels; liste de défauts; tutoriaux, API, la documentation	Oui	Inconnu	Inconnu
Madkit (Multi Agent Development Kit)	Plate-forme multiagent avec une couche de simulation agent	LGPL; GPL	Java	JVM (Java 2)	FAQ; documentation; forum en ligne; exemples; liste de défauts	Inconnu	Inconnu	Inconnu

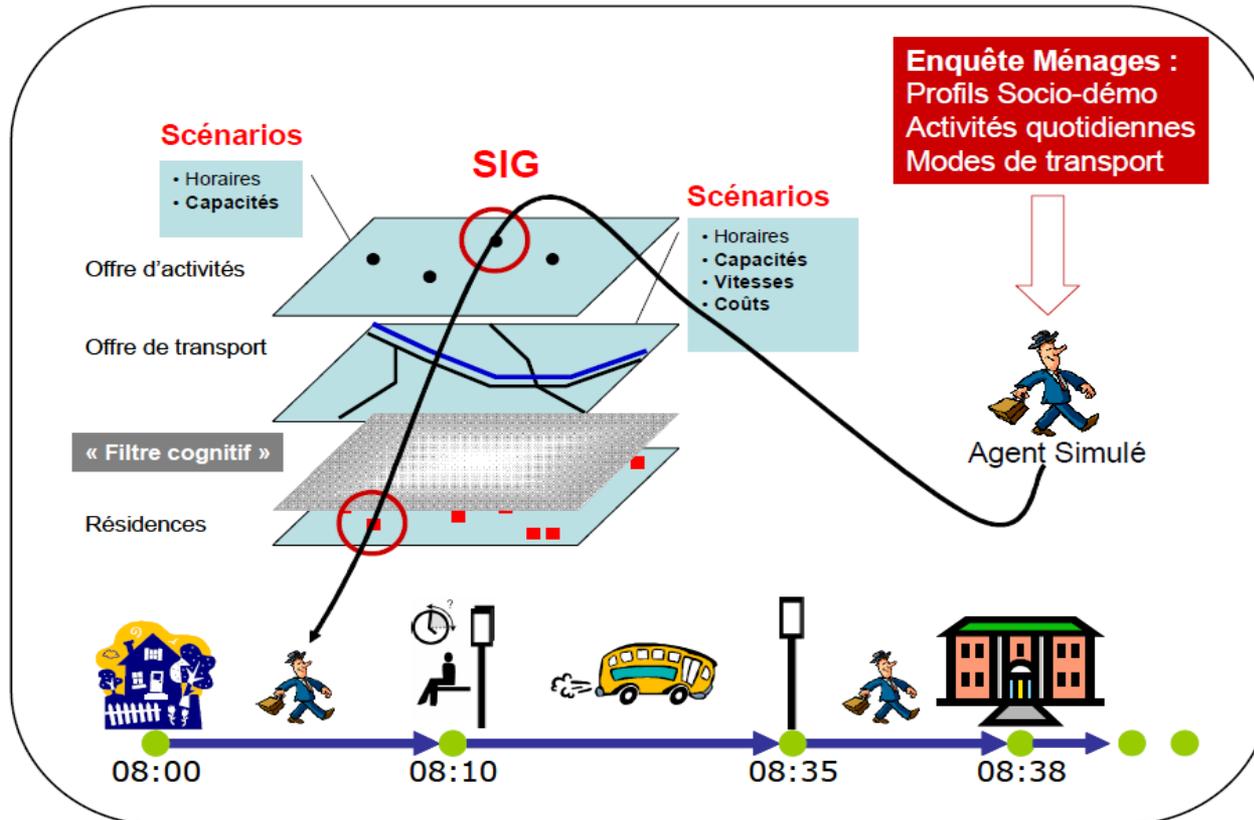
Plateforme de simulation

Plateforme	Domaine	License	Langage de programmation	Système d'opération	Support d'utilisateurs	FIPA Compliant	Capacité SIG	Capacité 3D
MASON (Multi-Agent Simulator Of Neighborhoods)	Complexité sociale, modélisation physique, modélisation abstraite	License académique gratuite (open source)	Java	Toute Java Plateforme	Documentation, tutoriaux, extensions troisième partie; documents de référence; API	Inconnu	Inconnu	Oui
MASS (Multi-Agent Simulation Suit)	simulations distribuées, simulations participatives.	Propriétaire, version gratuite disponible	Java; Exécuter des simulations avec Repast et NetLogo	Tout système d'exploitation avec Java 1.5,	Manuels, tutoriaux, documents de références	Non	Non	A partir de Java, toute visualisation est peut-être ajoutée.
RepastS (REcursive PorousAgentSimulationToolkit Symphony)	Sciences sociales	BSD	Java	Java version 1.4	Documentation; tutoriaux; documents de référence; FAQ; exemples	Inconnu	Oui	Oui

Plateforme de simulation

Plate-forme	Domaine	License	Langage de programmation	Système d'opération	Support d'utilisateurs	FIPA Compliant	Capacité SIG	Capacité 3D
SeSAM (Shell for Simulated Agent Systems)	Recherche, l'enseignement, les ressources, la théorie de graphe	LGPL	Simulation compilées à partir de spécification visuelle	à la Java 5.0 ou supérieur	Tutoriaux; liste de courriels; FAQ; wiki; contacter l'auteur	Plugin disponible	Raster- et Vector-GIS comme représentation spatiale,	Plugin disponible
NetLogo	Sciences sociales et naturelles;	Gratuit, pas open source,	NetLogo	Tout Java Virtual Machine	Documentation; FAQ; tutoriaux; extensions pour la troisième partie; liste de défauts; listes de courriels	Inconnu	Oui	Oui
GAMA	Sciences sociales et naturelles	Logiciel gratuit, LGPL	Java	Tout Java Virtual Machine	Documentation, tutoriaux	Oui	Oui	Oui

MIRO (Modélisation Intra-urbaine des Rythmes quOtidienS) est le modèle Multi-Agent de la Ville en Mouvement, s'intéresse aux déplacements de personnes au cours d'une journée dans les rues de la ville.

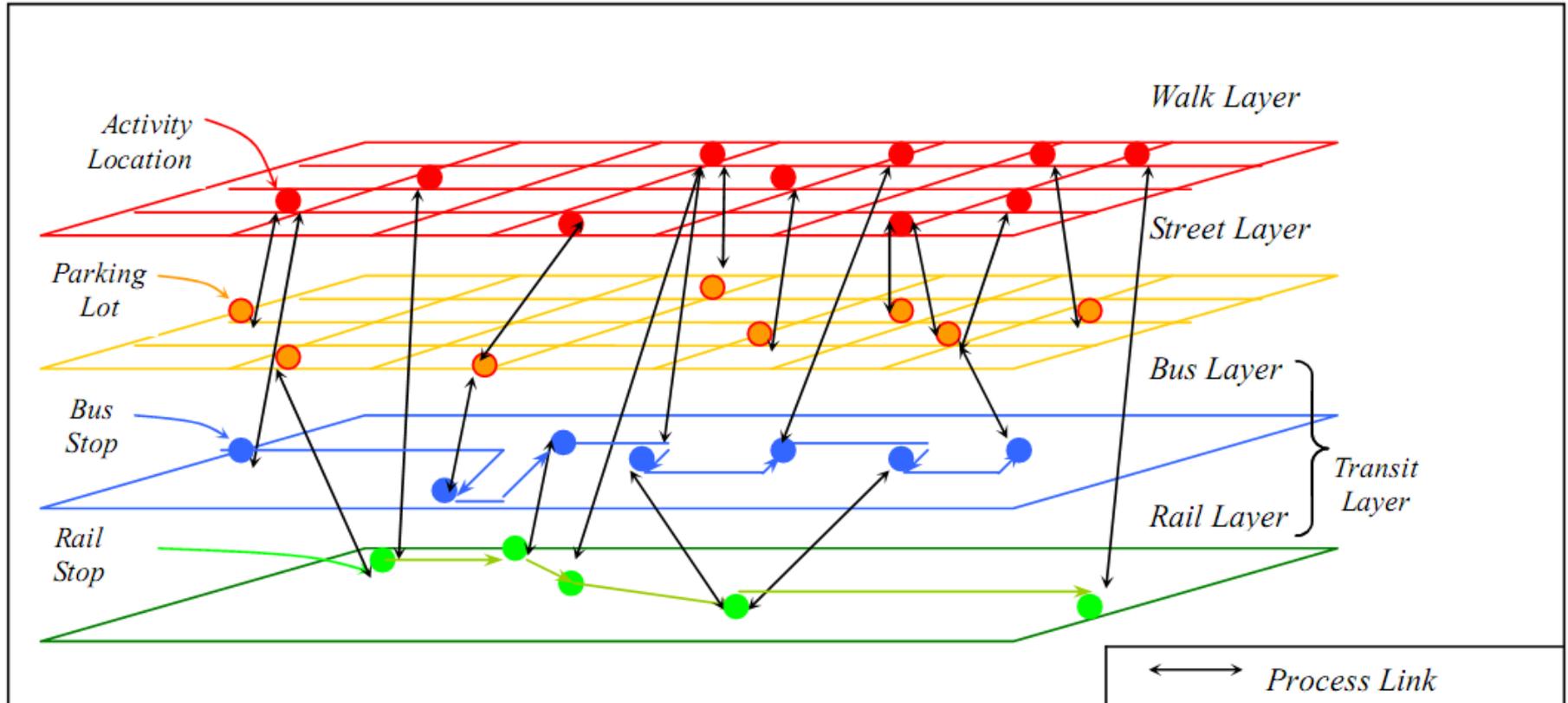


TRAnSIMS

<http://code.google.com/p/transims/>

- TRansportation Analysis and SIMulation System, développé par Los Alamos National Laboratory – Etats Unis
- TRANSIMS simule le mouvement multimodal des voyageurs (voiture, camion, piéton, vélo).
- Evaluer les impacts des changements de stratégies trafic ou des caractéristiques démographiques sur la distribution modale, l'heure de départ et les comportements en général des utilisateurs du réseau transport

TRAnSIMS



GéOpenSim

- Un système open source pour étudier l'évolution de l'espace urbain.
- Une représentation vectorielle et multi-niveaux de l'information géographique apte à décrire les tissus urbains,
- Des mécanismes pour simuler des évolutions du tissu urbain en se basant une représentation d'agents géographiques vectoriels

NetLogo

<http://ccl.northwestern.edu/netlogo/>

- NetLogo est un environnement pour développer des systèmes multi-agent
- Son développement a commencé en 1999 par Uri Wilensky
- C'est un héritier de StarLogo (1989-1990) sur Connection Machine
- 1994 une version sur Macintosh MacStarLogo
- 1997 StarLogoT

Logo

- Langage Informatique
- Le nom vient Logos « parole, discours, intelligence »
- Marvin Minsky et Seymour Papert
- Logo est un langage de programmation orientée objet réflexif
- Il est connu pour sa tortue graphique

HubNet

<http://ccl.northwestern.edu/netlogo/hubnet.html>

- HubNet est une évolution qui permet de fonctionner en client/serveur
- Le système permet une interaction avec un ensemble d'utilisateur

NetLogo

Il y a 4 types d'Agents :

- Turtles. Turtles sont des agents qui se déplacent dans le « monde »
- Patches: Case des grilles de déplacement
- Links: Les Links sont des agents qui relient deux « Turtles »
- The observer: Les « Observer » sont des observateurs sans localisation

NetLogo

Instructions :

- Il y a deux types d'instruction
 - Les procédures écrites par l'utilisateur
 - Les primitives proposées par NetLogo

```
to setup                ;; commentaires
  clear-all            ;; intialisation du monde
  create-turtles 10     ;; faire 10 tortues
end                      ;;
```

NetLogo

Instructions :

- Les instructions peuvent avoir des paramètres

```
to-report absolute-value [number] ;; valeur absolue
  ifelse number >= 0                ;; de number
    [ report number ]                ;;
    [ report 0 - number ]            ;;
end                                   ;;
```

NetLogo

Instructions :

- Une « command » exécute une action

```
to go
```

```
  ask turtles
```

```
    [ forward 1           ;; toutes les tortues avance d'un  
      right random 360 ];; et tourne de manière aléatoire
```

```
end
```

NetLogo

Variables :

- Variables globales

globals [number-of-trees]

- Variables turtle, patch et link

- NetLogo

turtles, (color, xcor, ycor, heading, ...)

link, (color, end1, end2, thickness, ...)

patch, (pcolor, pxcor, pycor, ...)

- Utilisateur (pour définir var1, var2, var3)

turtles-own[var1], patches-own[var2], links-own[var3]

NetLogo

Variables :

- Lecture des variables

`ask turtle 5 [show color] ;; affiche la couleur de la tortue 5`
`show [color] of turtle 5 ;; idem`

- Affectation de variables

`ask turtle 5 [set color blue] ;; couleur bleu pour tortue 5`
`ask turtles [set pcolor red] ;; permet de mettre en rouge la
;; position de la tortue`

NetLogo

Ask :

- Permet de spécifier une commande pour turtles, patches, links (touts les éléments) ou pour un élément turtle, patch, link
- patch-at

```
ask turtle 0           ;; demande à la tortue 0
[ ask patch-at 1 0    ;; demande à la case à son est
  [ set pcolor red ] ] ;; de devenir rouge
```

NetLogo

Variables locales :

```
to swap-colors [turtle1 turtle2] ;; turtle1 et turtle2 paramètres
  let temp ([color] of turtle1)   ;; couleur de turtle1
  ask turtle1 [ set color ([color] of turtle2) ]
  ask turtle2 [ set color temp ]
end
```

NetLogo

Squelette :

globals [...] ;; variables globales

turtles-own [...] ;; variables des tortues

patches-own [...] ;; variables des patches-cases

links-own [...] ;; variables de links-liens

...

to setup ;; intialisation

clear-all ... setup-patches ... setup-turtles ... setup-graphs ...
end

...

```
to go ;; Evaluation
  conduct-observer-procedure ...
  ask turtles [conduct-turtle-procedure] ...
  ask patches [conduct-patch-procedure] ...
  tick
  update-graphs ;; visualisation (update-plots, update-view)
end
...
```

```
to update-plots
  set-current-plot "my-plot"
  set-current-plot-pen "my-pen"
  plot statistics ...
  set-current-plot "my-XY-plot"
  plotxy ticks (count turtles) ...
  set-current-plot "my-histogram"
  histogram [age] of turtles ...
end

...

to-report statistics
... report the-result-of-some-formula
end
```

DaisyWorld

James Lovelock a proposé une abstraction pour représenter la rétroaction

- On a une terre avec modèle simple d'albédo (loi de Stefan-Boltzmann)
- Des pâquerettes blanches et noires en compétition changent l'albédo

<http://zool33.uni-graz.at/schmickl/models/daisyworld.html>

Jeux de Simulation

- 1989 SimCity
- 1990 SimEarth
- 1991 SimAnt
- 1992 SimLife
- 1993 SimFarm ...
- 2000 The Sims
- 2003 SimCity 4
- 2004 The Sims 2
- 2008 Spore ...

Mars Simulation Project

<http://mars-sim.sourceforge.net/>

- C'est un projet open-source pour une simulation d'une installation sur Mars
- Il comporte de nombreux éléments pour simuler l'installation d'humains sur Mars.

Bibliographie

NETLOGO 4.0 – QUICK GUIDE

Luis R. Izquierdo (<http://luis.izquierdo.name>)