

# Animation diagrammatique de simulations de systèmes dynamiques

Jean-Louis GIAVITTO, Franck DELAPLACE

LaMI<sup>1</sup>, Équipe SPÉCIF, UMR 8042 CNRS, Université d'Évry val d'Essonne, GENOPOLE  
Tour Évry-2, 523 Place des terrasses de l'agora, 91000 Évry Cedex

2 décembre 2003

**Nombre d'étudiants** : 1, ou 1 binôme

**Mots-clés** : visualisation diagrammatique, animation, *flash*, modélisation biologique, langage.

**Public visé** : TER de maîtrise, stage IIE, stage Polytechnique.

## Contexte de l'étude

Le projet MGS développe un langage de programmation original dédié à la modélisation et la simulation de processus biologiques à structure dynamique. Pour ce faire, MGS permet la représentation d'organisations complexes entre des entités variables et hétérogènes, ainsi que leur transformation par des règles locales. Ces travaux trouvent leurs inspirations dans les travaux de J. Von Neuman sur les automates cellulaires, A. Lindenmayer sur les L systèmes, G. Paun sur les P systèmes, G. Berry *et al.* sur la CHAM et la réécriture de multi-ensembles.

La structure de données fondamentale en MGS est la *collection topologique*. Une collection topologique est un ensemble d'éléments organisés par une relation de voisinage. Une *transformation* permet de spécifier de nouvelles fonctions sur les collections par des cas filtrant des *sous-collections*. Ces notions permettent d'unifier dans le même cadre formel les différents modèles de calculs cités plus haut. Pour chacun des modèles il suffit de choisir le bon voisinage pour la collection utilisée. Un point remarquable est l'existence d'un langage de filtres, utilisé pour écrire les règles d'une transformation, qui est commun à tous les types de collection. Ce langage de filtres se fonde sur la notion de voisinage et de chemin.

## Sujet du stage

La modélisation et la simulation de systèmes dynamiques biologiques en MGS produit une grande quantité de données. Cette masse de données *brutes* correspondant au résultat de la simulation ne permet pas une interprétation facile.

Pour permettre une lecture des résultats de la simulation, on peut interpréter graphiquement à l'issue de la *simulation* les données obtenues. Cette approche *post-mortem* permet de juger de la pertinence du *résultat* de la simulation.

On souhaite ici développer un procédé de visualisation des données *animées* afin de rendre compte du processus lui-même. La visualisation pourra avoir lieu au moment de la simulation, ou à l'issue de celle-ci, pour la rejouer par exemple. Il s'agira de développer en *Ocaml* un langage de manipulation de diagrammes et leur évolution temporelle (à la *flash* par exemple<sup>2</sup>).

---

<sup>1</sup>*Contacts* : par courrier électronique : {giavitto, delapla}@ReMoVeMeFIRST.lami.univ-evry.fr. Des informations supplémentaires sont disponibles à partir de la page : <http://mgs.lami.univ-evry.fr>

<sup>2</sup>Voir <http://www.openswf.org/>