

Calcul amorphe en MGS

Julien COHEN

LaMI¹, 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 binôme

Mots-clés : amorphous computing, modélisation biologique, langage.

Public visé : TER de maîtrise.

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

Le terme *amorphous computing* désigne un domaine récent de recherche visant à effectuer des calculs à partir de systèmes *a priori* inorganisés (amorphes). Par exemple un ensemble de cellules coopère afin de former un organisme multi-cellulaire sous la direction d'un programme génétique partagé par l'ensemble des cellules. Un essaim d'abeilles collabore pour construire une ruche. Les humains se réunissent pour construire des villages, des villes et des pays. Ces exemples amènent des questions fondamentales sur l'organisation des systèmes de calcul :

- Comment obtenir un comportement cohérent en faisant coopérer un grand nombre d'entités non fiables, dont l'interconnexion est inconnue, irrégulière et variable au cours du temps ?
- Quels langages de programmation et quelles méthodes adopter pour amener des myriades d'éléments programmables à coopérer afin d'atteindre un but précis ?²

De nombreux exemples d'applications ont émergés de ce domaine (matériaux programmables, structures auto-réparatrices, réseaux auto-organisés, *etc.*). Le but de ce stage consiste à implémenter des exemples significatifs en tirant parti des facilités amenées par le langage MGS.

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

²Amorphous computing : <http://www.swiss.ai.mit.edu/projects/amorphous/>