Calcul amorphe en MGS

Olivier Michel, Julien Cohen

LaMI[1], Équipe SPÉCIF, UMR 8042 CNRS, Université d'Évry val d'Essonne, GENOPOLE
Tour Evry-2, 523 Place des terrasses de l'agora, 91000 Évry Cedex

7 janvier 2005

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 systèmes dynamiques complexes (en particulier en biologie). `MGS` permet la représentation d'organisations sophistiquées entre des entités variables et hétérogènes, ainsi que leur transformation par des règles locales (interactions). Ces travaux se fondent sur des notions de topologie alégbrique et permettent des modèles de calculs variés comme les L-systèmes, le calcul chimique ou bien les automates cellulaires.

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.

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 courier électronique : `jcohen @ReMoVeMeFIRST.lami.univ-evry.fr`. Des informations supplémentaires sont disponibles à partir de la page : `http://mgs.lami.univ-evry.fr`

[2] Amorphous computing : `http://www.swiss.ai.mit.edu/projects/amorphous/`