

oRis : un langage pour simulations multi-agents

F.Harrouet⁽¹⁾, R.Cozien⁽²⁾, P.Reignier⁽¹⁾, J.Tisseau⁽¹⁾

- (1) Laboratoire d'Informatique Industrielle
Ecole Nationale d'Ingénieurs de Brest
C.P. n° 15, 29608 BREST Cedex
e-mail: [harrouet,reignier,tisseau]@enib.fr
- (2) Ecoles de Saint-Cyr Coëtquidan
Direction Générale de l'Enseignement et de la Recherche
56381 GUER Cedex

19 Février 1997

Le but de notre travail est de proposer un outil pour mettre au point, de manière assez simple, des simulations multi-agents. Il s'agit d'un langage orienté objet, interprété, dynamique dont la syntaxe est assez proche de celle de C++. Les simulations ont lieu pour l'instant sur une interface 2D qui permet d'avoir facilement un résultat visible. Ceci a été fait en essayant d'être le plus neutre possible, c'est à dire en ne supposant rien sur le sujet de la simulation réalisée.

Ecrire une simulation avec *oRis* consiste à décrire des modèles d'agents et à en instancier afin d'initialiser la simulation. La relation *modèle/agent* est semblable à *classe/instance*, les modèles servent à décrire les attributs et méthodes des agents correspondants. La notion d'agent est donc proche de celle d'objet si ce n'est qu'un agent a ses propres buts, qu'il n'est pas piloté de l'extérieur mais se comporte comme il l'entend. Dans *oRis*, cette différence réside dans la présence ou l'absence d'une méthode **reaction** (l'équivalent d'une fonction **main** par objet). Celle-ci n'est pas appelée explicitement par l'utilisateur mais automatiquement par le simulateur. Ainsi un cycle de simulation consiste à désigner chaque agent pourvu d'une **reaction** afin qu'il l'exécute (qu'il "vive"). La désignation peut se faire de manière ordonnée (ordre fixe) ou aléatoire sans remise afin d'éviter d'introduire des priorités entre agents.

Ce langage est interprété afin de permettre la modification du code en cours d'exécution. Il est ainsi possible de créer de nouveaux modèles ou de surdéfinir des méthodes avec exactement le même langage que celui utilisé dans le fichier script initial. Le nouveau code à prendre en compte peut provenir d'un autre fichier mais peut surtout être inscrit dans une chaîne de caractères qu'un ou plusieurs agents auront construite comme bon leur aura semblé. Ceci doit rendre possible une certaine forme d'apprentissage permettant aux agents de réécrire leur comportement au cours de leur "vie".

L'interpréteur est actuellement opérationnel mais l'interface 2D est en cours de finalisation. Une fois terminée, nous serons en mesure de porter les simulations écrites avec des versions antérieures du simulateur et surtout de tester les apports de cette nouvelle version.