

# CogLogo: une implémentation de MetaCiv pour NetLogo

François Suro  
suro@lirmm.fr

Jacques Ferber  
ferber@lirmm.fr

Tiberiu Stratulat  
stratulat@lirmm.fr

LIRMM, Université de Montpellier, CNRS, Montpellier, France

## Abstract

*CogLogo est une extension pour NetLogo qui implémente les principes de MetaCiv. Le but est de proposer un framework pour la modélisation du système de décision d'un agent social intégrant un mécanisme de renforcement.*

**Mots-clés :** ABS, Agent Based Simulation, multi-agent systems, MASQ, cognitons, MetaCiv, CogLogo, NetLogo

## 1 Introduction

MetaCiv est un framework générique de SMA reposant sur le méta-modèle MASQ, pour l'implémentation de systèmes sociaux complexes.

Cette architecture est constituée de deux éléments fondamentaux : les cognitons et les plans.

1. **Les cognitons** sont des "unités cognitives élémentaires" [1] dynamiques, pouvant être ajoutées ou retirées de l'esprit de l'agent. Ils peuvent représenter des croyances, des savoir-faire ou des percepts. L'ensemble des cognitons présents dans l'esprit d'un agent à un instant donné constitue son état mental.
2. **Les plans** représentent les différentes activités que peut réaliser un agent. Chaque plan est lui-même constitué d'une succession d'actions, plus ou moins complexes. Afin de guider la prise de décision de l'agent, à chaque plan est associé un poids qui sera déterminé par les cognitons présents dans l'esprit de l'agent. De manière générale, un agent choisira d'appliquer un plan ayant le poids le plus élevé possible. Néanmoins, l'architecture de MetaCiv ne fixe pas le mécanisme précis du processus conatif, et il est possible d'en envisager plusieurs en fonction des objectifs du SMA. Par exemple, on peut choisir systématiquement le plan de poids le plus fort, effectuer un tirage aléatoire qui favorise les poids forts, etc.

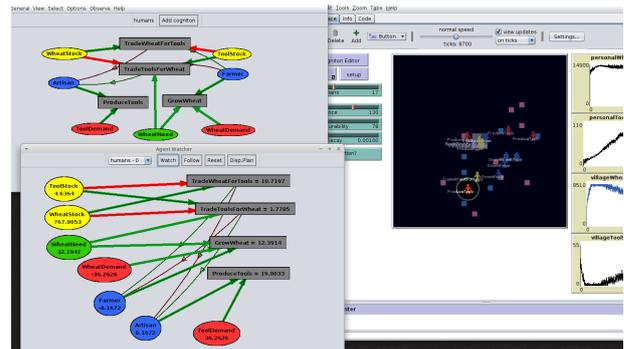


Fig. 1 – en haut à gauche, l'éditeur de schéma cognitif, à bas à gauche la fenêtre d'observation d'un agent montrant le poids des cognitons et la valeur calculée des plans.

## 2 L'extension CogLogo

CogLogo [1] est une extension NetLogo [2] écrite en Java qui implémente l'architecture à base de cogniton de MetaCiv [3]. CogLogo fournit une **interface graphique pour définir un schéma cognitif**, le rapport cogniton-plan à travers les liens d'influence. Le schéma cognitif prend en charge le processus de décision qui aboutit au choix du plan à exécuter. L'implémentation du Plan se fait sous forme d'une procédure NetLogo. CogLogo fournit aussi un **mécanisme de renforcement automatique**, sous forme de liens de feedback, un **choix de règles pour le processus de décision**, poids maximum ou stochastique, ainsi que la possibilité d'**observer l'évolution de l'esprit de chaque agent en temps réel**. Coglogo est un projet open source disponible à cette adresse :

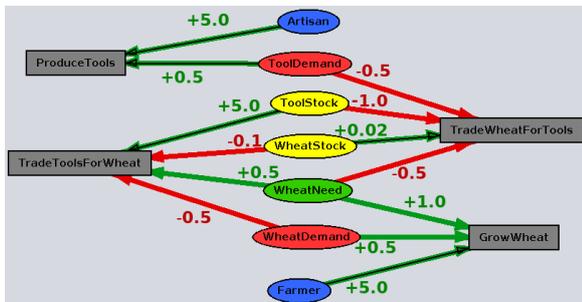
<https://github.com/suroFr/CogLogo>

## 3 Mise en place d'une simulation avec CogLogo

Un des exemples de simulation fournit avec CogLogo est un modèle Artisan-Agriculteur. Dans ce modèle les agents doivent produire de la nourriture pour survivre, ils peuvent aussi produire des outils à durabilité limitée qui multiplient le

rendement de leur activité de production de nourriture. Les agents ont la possibilité d'échanger de la nourriture contre des outils ou des outils contre de la nourriture. On souhaite observer l'émergence d'une spécialisation des agents en artisans et agriculteurs.

Le schéma cognitif des agents est le suivant :



De plus nous utilisons le système de renforcement (*feedback links*) qui augmente le poids des cognitions *Artisan* et *Farmer* si l'agent réussit une transaction d'outils contre blé et blé contre outils respectivement (ce qui caractérise un professionnel est la capacité à commercialiser son produit, et non à produire).

Ci-dessous quelques fonctions qui permettent à CogLogo de s'interfacer avec NetLogo :

```

1 ;; choose-next-plan renvoie une chaîne de
  ;; caractères, le nom d'une fonction, qui
  ;; peut être appelée par la primitive
  ;; "run"
2 run coglogo:choose-next-plan
3
4 ;; set-cogniton-value change la valeur de
  ;; cogniton fournit en premier paramètre
  ;; à la valeur fournie en second
  ;; paramètre.
5 coglogo:set-cogniton-value "ToolStock"
  toolStock

```

## 4 Résultats de la simulation

Voici le résultat d'une exécution de la simulation avec 25 agents sur une durée de 500 000 ticks

Les agents en rouge produisent des outils, en bleu de la nourriture. Les agents en jaune sont en train de réaliser un échange au marché (village).

La couleur de l'habitation d'un agent représente la moyenne de l'activité conduite par l'agent pendant les 25 000 derniers ticks : En rouge, production d'outils, en bleu production de nourriture, la teinte intermédiaire de violet indique la prédominance d'une activité ou l'autre.

Dans cette exécution de la simulation le manque

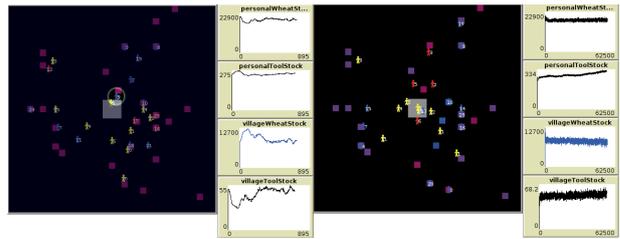


Fig. 2 – à gauche : état initial, à droite : tick 27 000

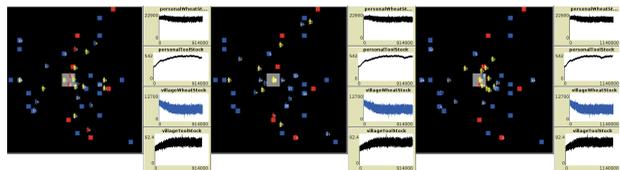


Fig. 3 – de gauche à droite : tick 400 000, 450 000 et 500 000

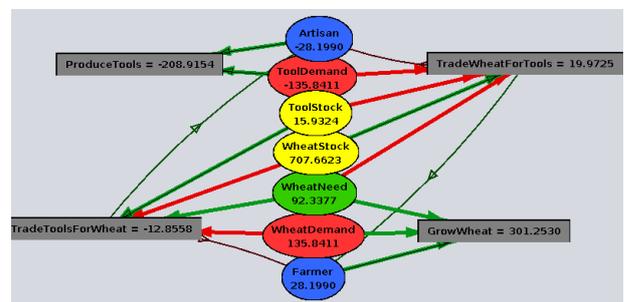


Fig. 4 – état interne d'un agent spécialisé en agriculteur

d'outils a été comblé par une spécialisation progressive de certains agents en artisans, mais dans d'autres exécutions, il est arrivé que trop d'agents se spécialisent en artisans, causant une surproduction d'outils. Dans ce cas, à cause du manque de succès dans l'échange d'outils contre de la nourriture, les agents les moins spécialisés parmi les artisans ont progressivement changé leur activité vers l'agriculture, causant un retour à l'équilibre du marché.

## Références

- [1] Suro, F., *CogLogo*, SMILE : Système Multi-agent, Interaction, Langage, Évolution., Laboratoire d'informatique, de robotique et de microélectronique de Montpellier., France, 2017, <https://github.com/suroFr/CogLogo>
- [2] Wilensky, U., *NetLogo*, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., 1999, <http://ccl.northwestern.edu/netlogo/>
- [3] Ferber, Jacques and Nigon, Julien and Maillé, Gautier and Stratulat, Tiberiu., *MetaCiv : un framework multi-agent basé sur MASQ pour modéliser des sociétés humaines*, JFSMA : Journées Francophones sur les Systèmes Multi-Agents, 2014, <https://hal-lirmm.ccsd.cnrs.fr/lirmm-01463384>