# Donner corps aux interactions (l'interaction enfin concrétisée)

P. Mathieu, S. Picault, J.-C. Routier

▶ **To cite this version:**

P. Mathieu, S. Picault, J.-C. Routier. Donner corps aux interactions (l'interaction enfin concrétisée). 4e Journées Francophones sur les Modèles Formels de l'Interaction (MFI'07), May 2007, Paris, France. pp.333-340. hal-00193333

**HAL Id: hal-00193333**
**https://hal.science/hal-00193333**

Submitted on 3 Dec 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Donner corps aux interactions (l'interaction enfin concrétisée)

P. Mathieu* S. Picault* J.-C. Routier*
mathieu@lifl.fr picault@lifl.fr routier@lifl.fr

*Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq cédex – FRANCE

**Résumé :**
Depuis plusieurs années, nous utilisons IODA comme méthode de description et de réalisation de simulations multi-agents. Cette méthode a pour originalité de concrétiser les interactions de manière à ce qu'elles soient génériques et réutilisables dans différents contextes. Cet article a pour objectif d'identifier les problèmes durs dans ce type de simulation et de montrer comment IODA apporte une aide à leur résolution.

**Mots-clés :** Interactions concrètes, sélection d'action, comportement, simulation

**Abstract:**
Since several years, we use in our team the IODA methodology to describe and realize multi-agent simulations. This method is original and not similar to the others because it makes Interactions becoming concrete and then able to become sufficiently general to be reused in many contexts. The aim of this paper is to identify some hard problems in this kind of simulation and to show how IODA can be helful to solve them.

**Keywords:** Concrete Interactions, selection of actions, behaviour, simulation

## 1 Introduction

Le formalisme que nous proposons ici s'inscrit dans le contexte plus vaste de la simulation par agents où des entités autonomes (les agents) sont dotées d'un *comportement individuel*.

Dans ce cadre, de nombreux modèles d'analyse et d'implémentation ont été conçus afin de représenter les comportements des agents. La notion d'interaction a également été intégrée à de nombreuses méthodologies de conception de SMA, comme par exemple dans l'approche Voyelles [1]. Toutefois l'interaction reste un concept utilisé lors de la phase d'analyse, sans pour autant conduire à une implémentation informatique au cœur de la simulation. Ainsi, les interactions entre agents, même lorsqu'elles sont prises en compte lors de la modélisation, finissent par être codées dans un comportement d'agent *centré sur l'agent*.

Nous soutenons donc qu'il est nécessaire, dans de nombreuses situations de simulation, de définir les interactions d'une façon indépendante des agents, de formaliser très finement la façon de mettre en relation ces interactions et les agents qui peuvent les effectuer ou les subir, et de les coder explicitement dans le simulateur. C'est ce que nous appelons « l'approche centrée interaction » par opposition à l'approche classique dans les simulations distribuées, qui est centrée agent. À l'heure actuelle, aucun simulateur multi-agent n'applique une telle distinction de bout en bout de la tâche de simulation, c'est-à-dire de la modélisation au codage.

Nous avons formalisé et expérimenté le concept d'interaction à travers le projet IODA[1] [3]. Nous présentons d'abord le modèle formel qui permet, au sein de ce projet, de donner une définition opérationnelle de l'interaction. Nous décrivons ensuite la méthode d'analyse de IODA, qui décrit comment les interactions doivent être affectées aux agents, ce qui se traduit immédiatement par une implémenta-

[0]Ce travail est cofinancé par le contrat de plan Etat-Région et les fonds européen FEDER

[1]Pour : *Interaction-Oriented Design of Agent simulations*

tion univoque. Nous montrons à cette occasion le fonctionnement du moteur de simulation, réalisé actuellement dans la plate-forme IODA-light. Cet outil montre que nos concepts n'en restent pas au stade de la seule méthodologie mais permettent de donner une réalité logicielle aux interactions.

## 2 L'approche centrée interaction

Les Systèmes Multi-Agents sont construits sur un schéma d'emboîtement de compétences qui semble aller de soi : l'environnement modélise le monde physique ; il contient des agents qui représentent les entités de ce monde ; eux-mêmes contiennent des architectures de sélection d'action qui gouvernent le choix de comportements, et ceux-ci reposent sur des primitives de perception, de cognition et d'action propres à l'agent.

Or, si l'interaction reste cantonnée à la phase d'analyse, c'est en grande partie parce qu'il est difficile de réifier cette notion au moyen de comportements « encapsulés » dans les agents. Nous prenons donc le contre-pied de cette hiérarchisation : dans le modèle que nous proposons, nous cherchons donc à donner un poids opérationnel égal tant aux entités du système qu'aux activités auxquelles elles prennent part.

### 2.1 Le modèle formel de l'interaction dans IODA

Notre modèle d'interaction s'appuie sur des primitives de base qui fixent le niveau de granularité le plus petit qui puisse être représenté dans une simulation donnée. Nous distinguons des primitives de perception (stimuli, communication, croyance, ...) et des primitives d'action (déplacement, modification interne, destruction, création, ...).

Les interactions sont définies comme des ensembles de primitives qui impliquent simultanément plusieurs agents et qui constituent un bloc sémantique dans une simulation donnée [2]. Par exemple *manger* ou *ouvrir* ne sont pas de simples actions atomiques, mais correspondent à des ensembles structurés d'actions mettant en jeu deux agents différents et qui ne peuvent être effectuées qu'à certaines conditions, peu dépendantes des spécificités des agents concernés.

**DÉFINITION 1 (INTERACTION)**
*Une **interaction** est une séquence d'actions primitives, s'appliquant à plusieurs agents, déclenchées par des perceptions spécifiques et soumises à certaines conditions d'exécution.*
*Ces perceptions et ces actions primitives peuvent être réalisées selon des modalités variables par les agents, mais leur enchaînement logique est décrit de façon générale par leur structuration sous la forme d'une interaction (cf. fig. 1).*

**DÉFINITION 2 (SOURCES/CIBLES)**
*Les agents qui prennent part à une interaction ne jouent généralement pas le même rôle. On distingue dont entre des agents **sources** qui peuvent **effectuer** l'interaction, et des agents **cibles** qui peuvent la **subir**. Pour avoir lieu, une interaction doit mettre en relation des agents sources et des agents cibles.*
*N.B. : Dans les interactions symétriques, on peut évidemment intervertir sources et cibles.*

Une interaction, en tant qu'expression abstraite d'un comportement, doit formuler les conditions portant sur la réalisation de la séquence d'actions primitives. Nous avons distingué deux composantes dans les tests exprimant les pré-requis nécessaires à une interaction : le *déclencheur*, qui exprime une motivation pour les agents à effectuer l'interaction et la *condition* proprement dite, qui exprime les pré-requis

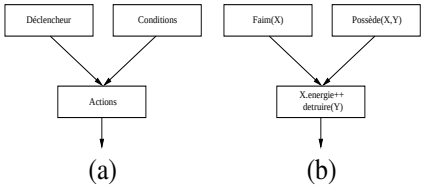


FIG. 1 – (a) Représentation générale d'une interaction. L'interaction est constituée d'un ensemble de perceptions (déclencheur d'une part, conditions d'exécution d'autre part) qui, lorsqu'elles sont réunies, permettent la réalisation d'une séquence d'actions. (b) Exemple : l'interaction *manger(X,Y)* se décrit de façon générale à partir d'une motivation interne (la faim) qui sert de déclencheur, de conditions d'exécution (posséder l'objet à manger) et de la séquence d'actions résultant de l'interaction : augmentation de l'énergie de l'agent source $X$ (celui qui effectue *manger*) et destruction de l'agent cible $Y$ (celui qui subit *manger*).

matériels ou logiques pour pouvoir effectuer la séquence d'action (cf fig.1).

Ainsi formulée, cette notion d'interaction apparaît complètement dissociée des agents qui les utiliseront. Une même interaction peut alors être réutilisée dans différentes simulations et appliquée à des agents sources ou cibles différents : ainsi, l'interaction *ouvrir* a évidemment la même sémantique, et répond à la même fonctionnalité, qu'il s'agisse d'une simulation d'évacuation de bâtiment en cas d'urgence ou d'une chasse au trésor dans un jeu vidéo. Concevoir une simulation consiste donc d'abord à établir les primitives de base qui pourront être utilisées, à ensuite les agréger dans des interactions, et enfin affecter celles-ci aux agents, et ce aussi bien lors de l'analyse du problème qu'au moment de l'implémentation de la simulation. Nous décrivons au § 3.1 comment ces interactions peuvent être affectées aux agents pour les doter d'un comportement et au § 4 comment le moteur de la simulation choisit les interactions à réaliser au sein du système.

## 2.2 Structure des agents dans IODA

Dans IODA, les agents se réduisent à des spécifications simples qui permettent d'intégrer n'importe quel agent à notre modèle de simulation centré interactions.

**DÉFINITION 3**
*Un **agent** est une entité autonome dont les caractéristiques minimales sont les suivantes :*
- *il est doté d'un état ;*
- *il dispose de primitives de perception et d'action ;*
- *perception et action ne s'étendent pas à tout l'environnement : elle se restreignent à un **halo** $\mathcal{H}$ propre à l'agent qui est une fonction retournant un sous ensemble de l'environnement ; en particulier l'agent ne perçoit pas tous les autres agents mais seulement ses **voisins**, c'est-à-dire ceux présents dans son halo de perception ;*
- *il se voit affecter la liste des interactions qu'il peut effectuer ou subir, chacune avec un niveau de priorité et une garde de distance.*

**DÉFINITION 4**
*Le **voisinage** $V$ d'un agent $x$ est l'ensemble des agents perçus par $x$, i.e. présents dans son halo de perception :*
$V(x) = \{y | y \in \mathcal{H}(x)\}$

## 2.3 Cardinalité des interactions

Un cas trivial d'interaction fait appel à une source et une cible, par exemple *manger* ou *ouvrir*. Toutefois, la complexité des problèmes à simuler nécessite de prendre en compte d'autres situations que nous allons examiner pour montrer qu'on peut les reformuler sous une forme normale qui ne requiert qu'une source.

**DÉFINITION 5**
*On appelle **cardinalité** d'une interaction le couple composé du nombre de sources*

*et du nombre de cibles nécessaires pour la réalisation de l'interaction.*

**Une source, une cible (cardinalité 1/1)** En première approche, une interaction peut être effectuée par une source sur une cible. La source est un agent qui *peut effectuer* l'interaction, et la cible un agent qui *peut la subir*, ces deux informations étant établies lors de l'analyse puis traduites dans une matrice d'interaction (cf. § 3.1).

**Une source, aucune cible (1/0)** Par extension on peut concevoir des interactions sans cibles définies. Celles-ci correspondent à deux situations : L'interaction réflexive (un agent qui agit sur lui-même), l'interaction avec l'environnement (un agent agit sur l'état du monde sans que d'autres agents soient impliqués).

**Une source, plusieurs cibles (1/n)** IODA permet également de spécifier des interactions s'appliquant à plusieurs cibles simultanément (toujours à partir d'une source unique) pour représenter des activités nécessitant une coordination entre agents, soit de même classe, soit de classes différentes. Cela s'exprime en indiquant dans la matrice d'interaction les cardinalités appropriées (cf. § 3.1).

**Plusieurs sources, une cible (n/1)** Le cas symétrique impliquant plusieurs sources et une cible n'est pas utilisé dans IODA, dans la mesure où il peut se ramener systématiquement au précédent en exprimant l'interaction à la voix passive. Par exemple, pour faire transporter un meuble par plusieurs déménageurs, on n'utilisera pas l'interaction *transporter* qui requerrait l'action simultanée de plusieurs sources (les déménageurs) sur une même cible (le meuble), mais plutôt l'interaction *être transporté* qui, elle, ne fait appel qu'à une source (le meuble) interagissant simultanément avec plusieurs cibles.

**Plusieurs sources, plusieurs cibles (n/p)** Dans IODA l'interaction simultanée entre plusieurs sources et plusieurs cibles est ramenée systématiquement sous la forme $1/(n-1+p)$, ou à plusieurs interactions 1/n. Cette solution, bien que critiquable, donne satisfaction dans tous les cas que nous avons traités. Cela reste néanmoins un problème dur.

**DÉFINITION 6 (FORME NORMALE)**
*Une interaction est écrite **sous forme normale** lorsqu'elle fait intervenir exactement une source.*
*Toute interaction peut s'écrire sous forme normale :*
- *les interactions de cardinalité 1/0, 1/1 ou 1/n sont déjà sous forme normale ;*
- *une interaction $I$ de cardinalité n/1 peut s'exprimer à la voix passive (« être $I$-é ») pour se ramener à une cardinalité 1/n ;*
- *dans une interaction de cardinalité n/p, on peut choisir comme source unique l'un des $n$ agents sources et reléguer les autres parmi les cibles, soit se ramener à une cardinalité 1/(n-1+p).*

Dans le reste de cet article, nous considérerons que toutes les interactions sont écrites sous forme normale.

## 3 La méthode d'analyse IODA

Attaquer un problème de simulation suppose d'identifier à la fois les entités qui, selon le modèle du domaine ciblé, sont supposées interagir les unes avec les autres pour produire le phénomène étudié, et ces interactions elles-mêmes. Dans un modèle de simulation centré agents, l'identification se focalise sur les entités, que l'on dote ensuite de comportements destinés à produire les interactions voulues. Les fonctions abstraites associées aux interactions sont ainsi perdues et « engluées » dans la spécificité des agents. Nous suggérons au contraire de mener de front l'ana-

lyse des agents et des interactions, de façon à garder une vue abstraite des fonctionnalités assurées par les agents.

La méthodologie IODA propose ainsi trois étapes pour la conception d'une simulation centrée interactions :

1. Identifier *d'abord* les interactions (fonctionnalités abstraites, processus élémentaires). Cela conduit à dresser une matrice entre sources et cibles potentielles (cf. tab. 1) dans laquelle on fait apparaître ensuite les interactions génériques.
2. Écrire les déclencheurs, conditions et actions de ces interactions.
3. Identifier les caractéristiques des agents concernés (attributs), ainsi que les primitives de perception et d'action (par exemple `faim`, `détruire`...), d'après les déclencheurs, conditions et actions constituant les interactions auxquelles ces agents devront participer.
4. Spécifier, pour toute affectation d'une interaction à des agents source et cible(s), la priorité relative de cette interaction et sa garde de distance. Cela conduit à raffiner la matrice précédente.
5. Déterminer enfin la dynamique du système, c'est-à-dire la façon dont, au fil des interactions et pendant la simulation, évolue cette matrice (et par conséquent, les possibilités d'interaction des agents). Voir [3] pour plus de détails.

### 3.1 La matrice d'affectation des interactions aux agents

Une fois définies les interactions susceptibles d'être réalisées au cours d'une simulation, il est en général facile de déterminer quels agents seront cibles ou sources. Il reste néanmoins deux points à préciser :

- La condition de distance entre la source et la cible pour que l'interaction puisse se produire. En effet, les agents n'interagissent potentiellement qu'avec des agents suffisamment « proches », qu'il s'agisse d'une distance spatiale dans l'environnement ou d'une mesure de proximité dans un espace d'états.
- La priorité que prend une interaction donnée lorsqu'elle est affectée à un agent donné, par rapport aux autres interactions qu'il est susceptible d'effectuer. Pour qu'un comportement rationnel puisse résulter des interactions entre agents, il faut en effet hiérarchiser ces interactions les unes par rapport aux autres, et ce d'une façon qui dépend assez étroitement des caractéristiques fonctionnelles des agents sources de cette interaction.

C'est lors de cette phase d'affectation des interactions génériques à des agents concrets, et lors de la définition des priorités et des gardes de distance associées à chaque interaction pour un agent donné, que l'on peut affiner les comportements produits au cours de la simulation. Au reste, rien n'exige que cette affectation reste inchangée au cours du déroulement de la simulation.

**DÉFINITION 7 (ASSIGNATION)**
*On appelle* ***assignation*** *des interactions $I_1, I_2 \dots I_n$ entre une source $\mathcal{S}$ et une cible (ou un groupe de cibles) $\mathcal{T}$, un* ***ensemble*** *de 4-uplets de la forme $(I_k, c_k, p_k, d_k)$ avec :*

- *$I_k$ : l'interaction pouvant être effectuée par $\mathcal{S}$ et subie par $\mathcal{T}$*
- *$c_k$ : la cardinalité de l'interaction (nombre de cibles de type $\mathcal{T}$ attendues)*
- *$p_k$ : la priorité donnée à l'interaction $I_k$ de ce 4-uplet par rapport à toutes celles que la source peut effectuer*
- *$d_k$ : la garde de distance $\mathcal{S}/\mathcal{T}$ en deçà de laquelle l'interaction est réalisable (facultative si la garde se conforme au halo de perception de la source, ou si $c_k = 0$).*

*L'assignation $a_{\mathcal{S},\mathcal{T}}$ décrit donc l'ensemble des interactions que $\mathcal{S}$ et $\mathcal{T}$ peuvent réali-*

*ser conjointement.*
*N.B.1 : lorsque $c_k > 1$ et que deux cibles au moins appartiennent à des catégories différentes, on notera plutôt $\mathcal{T} = (\mathcal{T}_1 \dots \mathcal{T}_n)$.*
*N.B.2 : la garde de distance $d_k$ peut être arbitrairement grande, mais elle est en pratique bornée par le halo $\mathcal{H}$ de l'agent.*

| S / T | ∅ | F | P | S | … |
|---|---|---|---|---|---|
| F | (Create-Soldier ; 0 ;1 ;—) (Create-Peasant ; 0 ;0 ;—) | | (Give-Task ; 1 ;2 ;—) | (Give-Task ; 1 ;3 ;—) | … |
| L | | | | (Give-Path ; 1 ;0 ;5) | … |
| M | | | | | … |
| S | (Become-Chief ; 0 ;2 ;—) (Move ; 0 ;0 ;—) | (Protect ; 1 ;1 ;2) | | | … |
| E | (Become-Chief ; 0 ;4 ;—) (Move ; 0 ;0 ;—) | (Destroy ; 1 ;1 ;2) | (Fight ; 1 ;2 ;5) | (Fight ; 1 ;3 ;5) | … |
| P | (Move ; 0 ;0 ;—) | (PutGold ; 1 ;1 ;0) | | | … |

TAB. 1 – Extrait de la matrice des interactions réalisables dans la simulation « Age of Empires » testée sur notre plate-forme. On trouve en ligne les sources et en colonne les cibles. Agents : **F**, forum (crée des paysans et des soldats) ; **M**, mines (fournissent des ressources) ; **P**, paysans (exploitent les mines et déposent l'or au forum) ; **S**, soldats (défendent paysans et forum contre les soldats ennemis) ; **E**, ennemis (soldats d'invasion) ; **L**, limites (bornes entre lesquelles patrouillent les soldats). Chaque case $a_{ij}$ de la matrice contient une *assignation*, i.e. une liste donnant les interactions qui peuvent être effectuées par la source $i$ sur la cible $j$, avec une cardinalité (nombre de cibles), un niveau de priorité (relatif à l'ensemble des interactions que $i$ peut effectuer) et une garde de distance qui peut être vide.

**DÉFINITION 8**
*On appelle* ***matrice d'interaction*** *la matrice $M = (a_{i,j})$ de toutes les assignations ($a_{i,j} = \{(I_1, c_1, p_1, d_1), \dots (I_n, c_n, p_n, d_n)\}$) entre sources et cibles dans la simulation. Ces sources et cibles peuvent être aussi bien des agents individuels que des catégories abstraites (classes, groupes, équipes, etc). Par conséquent, dans toute simulation qui comporte des interactions sans cibles (i.e. de cardinalité 1/0), il existe dans la matrice d'interaction une colonne $(a_{i,\varnothing})$.*

La forme générale de cette matrice est donnée sur un exemple dans le tableau 1.

### 3.2 Critères d'éligibilité d'une interaction

**DÉFINITION 9 (ÉLIGIBILITÉ)**
*Pour un agent $x$, une interaction $I$ est dite* ***éligible*** *si $x$ peut être source de $I$ et s'il existe dans le voisinage $V$ de $x$ des agents pouvant être cibles de $I$, en respectant les gardes de distances. N.B. L'éligibilité porte sur des critères syntaxiques (possibilité d'être source ou cible d'après la matrice d'interaction), et non sémantiques : une interaction éligible n'est réalisable en pratique que si les déclencheurs et les conditions sont vérifiés.*

Le moteur de simulation repose principalement sur l'évaluation des critères d'éligibilité des interactions susceptibles d'être effectuées par chaque agent du système. On peut formuler le critère d'éligibilité pour 3 cas, selon la cardinalité de l'interaction considérée (on désigne ci-dessous la matrice d'interaction par $M$) :

– pour une interaction de cardinalité 1/0 (pas de cible) : la possibilité d'effectuer l'interaction $I$ ne dépend que de l'agent source $x$

**Critère d'éligibilité (1/0) :**
$\textit{éligible}(x, I, \varnothing) \iff \exists a_{\mathcal{S}} \in M$
tel que $x \in \mathcal{S}$ et $(I, 0, p, d) \in a_{\mathcal{S}}$

– pour une interaction de cardinalité 1/1 (une seule cible) : la possibilité d'effectuer $I$ dépend de la source $x$ et du choix d'une cible potentielle $y$ dans le voisinage $V(x)$ appartenant à une cible spécifiée dans la matrice d'interaction

**Critère d'éligibilité (1/1) :**
$\forall y \in V(x)$, *éligible*$(x, I, y) \iff$
$\exists a_{\mathcal{S},\mathcal{T}} \in M$ tel que $x \in \mathcal{S}, y \in \mathcal{T}$,
$(I, 1, p, d) \in a_{\mathcal{S},\mathcal{T}}$ et *dist*$(x, y) \leq d$

– pour une interaction de cardinalité 1/n (plusieurs cibles) : outre la source $x$, il faut considérer les parties de $V(x)$ de cardinal $n$ (i.e. les arrangements de $n$ cibles $y_i$ possibles, chacune devant a priori appartenir à une catégorie $\mathcal{T}_i$ spécifiée dans la matrice d'interaction)

**Critère d'éligibilité (1/n) :**
$\forall (y_i)_{i\in[1,n]} \in \mathscr{P}^n(V(x))$, *éligible*$(x, I, (y_i))$
$\iff \exists a_{\mathcal{S},(\mathcal{T}_1 \ldots \mathcal{T}_n)} \in M$ tel que
$x \in \mathcal{S}, \quad \forall i \in [1, n]\ y_i \in \mathcal{T}_i$,
$(I, n, p, d) \in a_{\mathcal{S},(\mathcal{T}_1 \ldots \mathcal{T}_n)}$ et *dist*$(x, y_i) \leq d$

Il reste à définir concrètement, dans le moteur, l'algorithme de choix des interactions parmi toutes celles qui peuvent être effectuées à un moment donné dans tout le système multi-agents. Nous en proposons une réalisation au § 4.

# 4 De la méthodologie à l'implémentation

La plateforme que nous avons baptisé « IODA-light » vise à donner une implémentation exacte (sans heuristique) de la méthodologie IODA, ce qui en fait une des seules qui permette de passer de façon univoque de l'analyse au code. Elle nous permet de prototyper des modèles centrés interactions et d'en étudier les propriétés. Elle est disponible sur

`http://www.lifl.fr/SMAC/projects/ioda`

**Restriction de cardinalité.** En raison des considérations de complexité mentionnées ci-dessus pour les interactions de cardinalité 1/n, elle est destinée à ne traiter que des interactions de cardinalité 1/0 ou 1/1, en faisant l'hypothèse qu'on peut décomposer assez souvent une interaction exprimée en cardinalité 1/n par une succession d'interactions de cardinalité 1/1 voire par l'utilisation de macro-agents.

**Le moteur de sélection d'interaction.** Lors du déroulement de la simulation, le rôle du moteur de sélection d'interaction consiste, pour chaque agent source, à choisir de façon équitable une interaction réalisable parmi toutes celles également réalisables et de même priorité, et à l'exécuter.

**DÉFINITION 10**
*Une interaction $I$ de cardinalité $n$ est* ***réalisable par*** *$x$ sur les cibles $T = \{y_1 \ldots y_n\}$ (noté $\Re(x, I, T)$) si elle est éligible pour ces cibles et que son déclencheur et sa condition sont tous deux vérifiés pour la source et les cibles.*
$\Re(x, I, T) \iff$ *éligible*$(x, I, T) \wedge$ *déclencheur*$(x, I, T) \wedge$ *condition*$(x, I, T)$
*N.B. : dans le contexte de IODA-light, on a $n \leq 1$ dont $T$ se réduit soit à l'ensemble vide (cardinalité 1/0), soit à une cible $y$ (cardinalité 1/1).*

**DÉFINITION 11**
*On appelle* ***potentiel d'interaction de niveau*** *$p$ de l'agent $x$, noté $\mathcal{P}_p(x)$, l'ensemble des couples formés par les interactions de priorité $p$ réalisables par $x$ et les cibles sur lesquelles elles peuvent être effectuées :*
$\mathcal{P}_p(x) = \{(I, T = \{y_1 \ldots y_n\}) \mid n =$ *card*$_x(I) \wedge p =$ *prio*$_x(I) \wedge \Re(x, I, T)\}$
*où* *card*$_x(I)$ *et* *prio*$_x(I)$ *désignent respectivement la cardinalité et la priorité de $I$ dans l'assignation correspondante pour la source $x$.*
*N.B. : dans le contexte de IODA-light, on a $n \leq 1$.*

On peut alors décrire l'algorithme qui permet au moteur de sélection d'interaction de veiller à ce que chaque agent puisse effectuer ou subir au plus une interaction par

pas de temps.
À chaque pas de temps :

1. Mettre à jour l'état de l'environnement.
2. Rendre tous les agents **activables** (i.e. leur permettre d'effectuer ou de subir une interaction).
3. Pour chaque agent activable $x$ :
   (a) Percevoir les caractéristiques de l'environnement dans le halo $\mathcal{H}(x)$ ;
   (b) Percevoir les agents voisins $V(x)$, puis retirer de $V(x)$ les agents qui ne sont plus activables (i.e. qui ont déjà participé à une interaction) ;
   (c) Mettre à jour l'état interne de l'agent $x$ ;
   (d) Déterminer les interactions **éligibles** ;
   (e) Initialiser le niveau de priorité $p$ au niveau maximal pour $x$ ;
   (f) Calculer $\mathcal{P}_p(x)$ ; si $\mathcal{P}_p(x) = \varnothing$, décrémenter $p$ et recommencer ;
   (g) Si on arrive à $\mathcal{P}_0(x) = \varnothing$, alors l'agent ne peut être source d'aucune interaction et son pas de temps s'achève (mais $x$ reste activable)
   (h) Sinon (i.e. dès qu'on a un niveau de priorité $p$ pour lequel $\mathcal{P}_p(x) \neq \varnothing$), choisir au hasard un couple $(I^*, T^*) \in \mathcal{P}_p(x)$ ;
   (i) Effectuer les actions de $I^*$ avec $x$ comme source et $T^*$ comme cible puis **désactiver** $x$ et les agents de $T^*$.

Cet algorithme garantit le choix équitable des interactions du niveau de priorité le plus élevé pour chaque agent ; en outre, la désactivation (étape i) évite qu'un agent ne prenne part plusieurs fois à une interaction au cours du même pas de temps.

# 5 Conclusion

La complexité de plus en plus grande des simulations à agents situés et la montée à l'échelle de ces applications nécessite d'avoir un guide méthodologique allant de la phase d'analyse du problème au code informatique. Lors de précédents articles nous avons proposé une approche centrée Interactions, nommée IODA, qui a l'avantage de concrétiser les interactions entre les agents offant ainsi une facilité de conception et une réutilisatibilité du code fortement améliorée par rapport à une approche classique. Après avoir résumé cette approche, cet article présente d'une part la différence entre IODA et l'approche traditionnelle sur un exemple concret et d'autre part les problèmes difficiles à résoudre au sens des Interactions et des classes de complexité algorithmiques associées. Dans un second temps, le paquetage Ioda-Light, qui fournit les classes nécessaires à l'application de cette méthode est décrit. Ce paquetage, initialement proposé à des fins pédagogiques, est une restriction de IODA aux interactions 1 :1 . Le passage de IODA et ses tableaux au squelette du code issu de IODA-ligth est quasi automatique, au point qu'une de nos perspectives à court terme est de fournir un outil de développement graphique pour réaliser ce type de simulation.

# Références

[1] Y. Demazeau. From Interactions to Collective Behaviour in Agent-Based Systems. In *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo, 1995.

[2] P. Mathieu, S. Picault, and J.-C. Routier. Simulation de comportements pour agents rationnels situés. In *Actes de la conférence Modèles Formels pour l'Interaction (MFI'03)*, pages 277–282, Lille, 2003.

[3] Philippe Mathieu and Sébastien Picault. Towards and interaction-based design of behaviors. In Marie-Pierre Gleizes, editor, *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS'05)*, 2005.