Une Forme de Rétro Ingénierie pour Systèmes Multi Agents : explorer l'espace des simulations

François Gaillard, Yoann Kubera, Philippe Mathieu, Sébastien Picault

Une Forme de Rétro Ingénierie pour Systèmes Multi Agents: explorer l'espace des simulations.

F. Gaillard* Y. Kubera* P. Mathieu* S. Picault*

*Equipe Systèmes Multi-Agents et Comportements
Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 Villeneuve d'ascq cédex – FRANCE
mail: prenom.nom@lifl.fr

Résumé

La conception habituelle d'une simulation d'un phénomène passe par sa modélisation et la réalisation d'une implémentation : son étude permet de déterminer si le modèle est correctement construit et peut expliquer le phénomène. Grâce à LEIA, nous renversons ce processus de conception en étudiant une simulation automatiquement générée en parcourant l'espace des simulations possibles de manière à identifier les phénomènes remarquables puis en comprendre les mécanismes sous-jacents. Cet article traite donc de la construction automatisée de modèles et leur implémentation à partir d'une ontologie, constituée d'interactions génériques que nous pouvons affecter à des familles d'agents, rédigée selon la méthodologie IODA. LEIA peut alors parcourir l'espace des simulations en s'appuyant sur des outils de transformation et de simplification de modèle puis identifier les phénomènes particuliers en s'aidant d'une métrique spécifiée en entrée, tout en impliquant l'utilisateur dans ce processus.

Mots-clés : *Interaction, Simulation, Rétro Ingénierie, Observation de modèle*

Abstract

The usual way to design a simulation of a phenomenon is to first build a model and then to implement it. The study of the simulation and its outcomes tells if the model is adequate and can explain the phenomenon. With LEIA, we reverse this process by studying an automatically built simulation by exploring the simulations space in order to identify remarkable phenomena and then understanding the underlying mechanisms. This paper deals with automated construction of models and their implementations from an ontology, consisting of generic interactions that we can assign to families of agents, following the IODA methodology. LEIA can explore the simulations space by using tools for processing and simplifying models, and then identify interesting phenomena by using a specified metric in input. The user is also implied in this process.

Keywords: *Interaction, Simulation, Reverse engineering, Model observation*

1 Introduction

LEIA, pour « LEIA lets you Explore Interactions for your Agents », renverse la vue habituelle de conception de simulations multi agents : nous générons une série de modèles aléatoires et leur implémentation à partir d'une ontologie spécifiée à l'avance, constituée d'interactions génériques que nous pouvons affecter à des familles d'agents. Ensuite, par raffinements successifs et l'utilisation des outils définis dans ce papier, l'utilisateur va pouvoir créer un modèle et étudier ses caractéristiques sous-jacentes. Le navigateur de l'espace des simulations LEIA permet donc d'effectuer un travail de reverse engeneering désigné en tant que « design recovery » par Chikofsky et Cross [2]. L'utilisateur est capable de pleinement comprendre les simulations et les modèles sous-jacents puis d'étudier les relations entre interactions, comme nous l'illustrons dans la figure 1.

De plus, par les variations des modèles, il agit comme un « chatouilleur d'idées » tel que Hofstadter l'envisage dans « Ma Thémagie » [7] : « Des variations sur un thème considérées comme l'essence de la créativité ». LEIA peut évoluer tout seul en choisissant à chaque période de temps la meilleure simulation au sens de la métrique choisie. Il peut aussi laisser ce choix à l'utilisateur, ce qui semble préférable puisque la majorité des phénomènes est indécidable.

LEIA proprose donc une approche contraire au processus classique de conception de la simulation d'un phénomène donné :

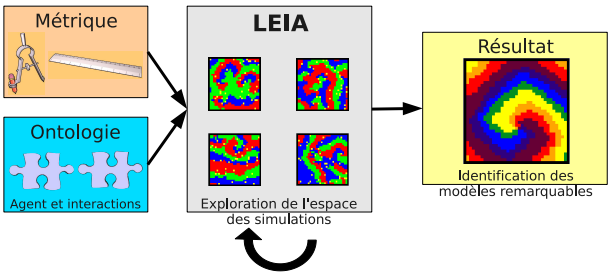

FIG. 1 – Ce schéma illustre le principe de fonctionnement de LEIA : nous spécifions en entrée une ontologie à base d'agent et d'interactions, ainsi que des outils de mesures adaptés au domaine simulé. LEIA permet alors d'explorer l'espace des simulations lié à cette ontologie et associer l'utilisateur au processus de découverte de modèles aux caractéristiques intéressantes.

celle-ci passe par la modélisation et la réalisation d'une implémentation. L'étude des résultats nous permet de déterminer si le modèle est correctement construit et s'il peut expliquer le phénomène donnée. Sinon, nous revenons à l'implémentation et effectuons des comparaisons entre modèles. Cette approche se retrouve dans un grand nombre de plateformes :

– orientées génie logiciel telles que Swarm [13], Madkit [6] ou Magique [15], qui laissent une grande liberté au concepteur pour créer ses agents, les comportements des agents et l'environnement ;
– orientées ergonomie comme Netlogo [18], conçu pour être employé par des non-informaticiens en reposant sur une forme très simple de programmation.

Cependant, l'ouverture et la généricité se font au détriment d'un cadre précis de la conception des comportements : le risque est alors de mélanger dans l'implémentation des agents le code métier et les connaissances propres au modèle.

La méthodologie IODA[1] [12] repose sur une séparation claire des agents, de leur comportement et du processus de sélection des actions. Dans cette méthodologie, les interactions sont réifiées indépendamment des agents qui les utilisent. De ce fait, nous pouvons établir des bibliothèques d'interactions pour un domaine particulier et adaptables à différentes familles d'agents, augmentant la généricité du travail de modélisation. Une implémentation réalisée au moyen du moteur JEDI [10] offre un support générique à l'utilisation de cette méthodologie.

[1] pour : *Interaction Oriented Design of Agent simulations*

Cette généricité s'illustre parfaitement dans le cadre du générateur JEDI Builder[2] : à partir d'un modèle conçu selon IODA, nous pouvons aisément obtenir une simulation exécutable avec le moteur JEDI.

Nous proposons également des outils de mesure, des métriques, pour qualifier les caractéristiques de chaque modèle réalisé avec IODA, elles peuvent porter sur :

– le modèle, par exemple le nombre d'interactions, les bouclages entre cible,etc ;
– les phénomènes observés, relatifs à l'ontologie ou définis de manière générale, comme :
  – l'activité de la simulation ;
  – les caractéristiques de l'environnement et de sa population ;
  – les variations des populations ;
  – l'évolution de l'usage des interactions au cours de la simulation.

Grâce à la méthodologie IODA, nous avons donc un formalisme qui nous permet de générer automatiquement des modèles et également de spécifier des outils de mesure pour caractériser les résultats. La méthodologie IODA est donc à la base de la construction de LEIA, comme illustré par la figure 2.

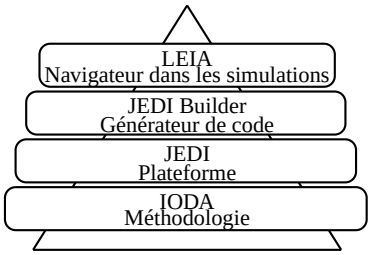

FIG. 2 – Hiérarchie dans la construction du navigateur : LEIA repose sur des simulations générées automatiquement pour la plate-forme JEDI, à partir d'un modèle spécifié dans la méthodologie IODA.

La section 2 présente la méthodologie IODA et son intérêt pour la conception du browser. Nous décrivons dans la section 3 des outils de mesure qualifiant la complexité de systèmes conçus suivant la méthodologie IODA. Enfin, Nous développons ensuite dans la section 4 la conception du browser LEIA.

[2] `www2.lifl.fr/SMAC/projects/ioda/demonstrations/`

2 La méthodologie IODA

La méthodologie de conception de simulation IODA[12] est une méthodologie centrée sur les interactions : elles sont réifiées indépendamment des agents. Les agents de $\mathbb{A}$, ensemble des agents présents dans l'environnement, disposent de primitives de base :

– primitives de perception relatives à l'état global de la simulation (noté $\mathbb{E}$), à l'environnement, à son état interne et à la communication avec les autres agents ;
– primitives d'action qui permettent de modifier l'état de la simulation : son propre état, l'état de l'environnement ou l'état d'autres agents.

Ces primitives permettent de définir le rôle que peut jouer l'agent dans des interactions spécifiques.

Dans la méthodologie IODA, les agents se réduisent à des spécifications simples et sont représentés de manière homogène, ce qui permet d'intégrer n'importe quel agent au modèle de simulation centré interaction : soit $\mathbb{F}$ l'ensemble des familles d'agents d'une simulation donnée, $\forall a \in \mathcal{A}$ et $\forall \mathcal{F} \in \mathbb{F}$, un agent a est une entité autonome et instanciée à partir d'une famille d'agents $\mathcal{F}$, notée $a \prec \mathbb{F}$. Une famille d'agents $\mathcal{F}$ parmi l'ensemble des familles $\mathbb{F}$ est l'abstraction d'un ensemble d'agents partageant tout ou partie de leurs primitives de perception ou d'action. Un agent a dispose également d'un halo de perception $\mathcal{H}_{\mathcal{F}}(a) \subseteq \mathbb{E}$: c'est à dire le sous-ensemble de l'état global de la simulation que l'agent a discerne au travers de ses primitives. Le voisinage $\mathcal{N}(a)$ d'un agent a est l'ensemble des agents présents dans son halo de perception : $\mathcal{N}(a) = \mathbb{A} \backslash \{a\} \bigcap \mathcal{H}_{\mathcal{F}}(a)$.

Une interaction est une séquence de primitives d'action s'appliquant à plusieurs agents, pouvant jouer le rôle de source ou de cible (respectivement notés $\mathcal{S}$ ou $\mathcal{T}$), et est soumise à des conditions d'activation. Les interactions apparaissent totalement dissociées des agents qui les utilisent, augmentant leur réutilisabilité à travers les simulations et applicables à des familles d'agents différentes. On définit la cardinalité d'une interaction $\mathcal{I}$ comme le couple composé du nombre d'agents sources $\mathrm{card}_{\mathcal{S}}(\mathcal{I})$ et du nombre d'agents cibles $card_{\mathcal{T}}(\mathcal{I})$ nécessaires à la réalisation de l'interaction.

Les assignations sont l'ensemble des interactions qu'un agent source peut faire avec des agents cibles. On appelle assignation $a_{\mathcal{S}/\mathbb{T}}$ d'un ensemble d'interactions $(\mathcal{I}_k)_{k\in[1,n]}$ entre une famille d'agents sources $\mathcal{S} \in \mathbb{F}$ et un q-uplet de familles d'agents cibles $(\mathcal{T}_j)_{j\in[1,q]} \subseteq \mathbb{F}^q$ l'ensemble des interactions appartenant aux agents $\mathcal{S}$ qu'ils peuvent effectuer en tant que sources avec les agents de $\mathbb{T}$ en tant que cibles. On définit un ensemble de tuples $(\mathcal{I}_k, p_k, d_k)$, $k \in [1, n]$, appelés éléments d'assignation, avec :

– $\mathcal{I}_k$: l'interaction pouvant être effectuée par toute source $a \prec \mathcal{S}$ et subie par les cibles $\{t_j, j \in [1,q]/t_j \prec \mathcal{T}_j \bigwedge (\forall l \in [1,q], t_j = t_l \Rightarrow j = l)$;
– $\mathrm{card}_{\mathbb{T}}(\mathcal{I}_k) = q$ le nombre de cibles dans $\mathbb{T}$;
– p_k : la priorité donnée à l'interaction $\mathcal{I}_k$;
– d_k : la garde de distance en deçà de laquelle l'interaction est réalisable.

On appelle **matrice d'interaction** la matrice $\mathcal{M} = (a_{\mathcal{S}/\mathbb{T}})_{\mathcal{S}\in\mathbb{F},\mathbb{T}\subseteq\mathbb{F}^{\mathbb{N}}}$ de toutes les assignations entre sources $\mathcal{S}$ et cibles $\mathbb{T}$ possibles (voir pour exemple Fig.4).

La construction d'un modèle dans la méthodologie IODA se fait en 6 étapes :

1. identification des familles d'agents et des interactions, comme constituants d'une matrice d'interaction ;
2. écriture des conditions d'activation et séquences d'actions qui sont primitives de chaque interaction ;
3. identification des primitives d'action et de perception des agents ;
4. spécification de la priorité et de la garde de distance de chaque interaction ;
5. détermination de la dynamique, c'est à dire l'évolution de la matrice d'interaction construite aux étapes précédentes ;
6. détermination des spécificités requises par le modèle.

Dans le cadre de l'élaboration de modèles dans le navigateur LEIA, les familles d'agents et d'interactions sont spécifiées à l'avance. Les conditions d'actions et de perception des interactions sont également fixées. La conception du modèle peut alors se limiter à choisir les interactions et les familles d'agents sans avoir besoin de générer du code pour les employer. Nous pouvons ainsi modifier à la volée le modèle d'une simulation sans avoir à arrêter l'application.

3 Les outils de mesure

LEIA peut être utilisé avec n'importe quel outil de mesure. Il n'a pas pour objectif de fixer certains d'entre eux mais plutôt d'offrir des outils pour exploiter au mieux l'étude de l'ontologie en entrée du navigateur. Nous spécifions dans cette section des mesures heuristiques qui ont pour objectif de caractériser la complexité d'un système conçu suivant la méthodologie IODA et simulé grâce au moteur JEDI (Fig.3).

Ces outils présentés dans cette section sont implémentés dans le moteur JEDI [**?**] qui est un moteur séquentiel et dont la représentation du temps est discrète. Une interaction $\mathcal{I}$ n'a qu'une seule source $\mathcal{S}$ ($\mathrm{card}_S(\mathcal{I}) = 1$). A chaque pas de temps, pour chaque source $\mathcal{S}$ potentielle, on sélectionne un couple (interaction $\mathcal{I}$, cible $\mathcal{T}$). L'interaction est choisie suivant l'assignation de priorité la plus élevée parmi toutes les assignations réalisables pour cette source $\mathcal{S}$. La cible $\mathcal{T}$ de cette interaction $\mathcal{I}$ doit également se trouver dans le voisinage de la source $\mathcal{N}(\mathcal{S})$. Pour le couple ainsi sélectionné, nous résolvons alors l'action de l'interaction $\mathcal{I}$ entre la source $\mathcal{S}$ et la cible $\mathcal{T}$. Cette interaction $\mathcal{I}$ est enregistrée en tant « interaction résolue ». A chaque pas de temps, nous pouvons donc avoir un retour quantifié sur l'ensemble des événements de la simulation dont découlent nos outils de mesure.

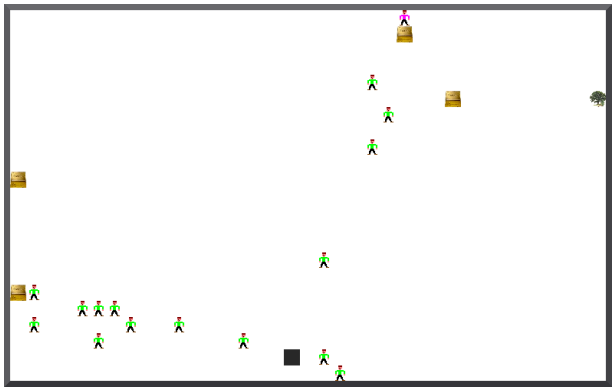

FIG. 3 – Capture d'écran d'une simulation de type « Age Of Empires » fonctionnant avec le moteur JEDI. Des paysans errent dans l'environnement, trouvent de l'or ou du bois puis les ramènent à leur forum. Ils communiquent aux autres paysans l'origine de ces ressources, afin de créer une chaîne de travailleurs partant du forum jusqu'à leur objectif commun.

Ces outils ont pour objectif de révéler les qualités et défauts des modèles simulés. Dans ce qui suit, nous nous plaçons, dans le cadre de LEIA, dans une situation simplifiée où chaque case de l'environnement ne peut être occupée que par un seul agent. La distibution initiale des agents et leurs primitives sont donc implémentées en conséquence.

3.1 Activité dans la simulation

Activité des agents Grâce au moteur JEDI nous pouvons suivre l'activité des agents, en particulier si un agent est en mesure d'effectuer des interactions. Cette mesure appelée « Activité des agents » caractérise l'interactivité de la simulation, c'est à dire la capacité de la simulation à fonctionner et donc évoluer.

Définition 3.1 ***Activité***
*Avec $\mathbb{I}(t)$ l'ensemble des « interactions résolues » au pas de temps t, soient et $\mathbb{A}(t)$ l'ensemble des agents dans une simulation donnée, l'**activité** des agents est donnée par :*
$Activite(t) = \mathrm{card}(\mathbb{I}(t)) / \mathrm{card}(\mathbb{A}(t))$

Le score fourni est donc le ratio entre le nombre d'agents qui sont sources d'une interaction et la totalité des agents de la simulation. Il faut noter que, dans le paramétrage par défaut du moteur JEDI, la source $\mathcal{S}$ et la cible $\mathcal{T}$ d'une interaction résolue sont désactivés[**?**] : la cible $\mathcal{T}$ ne peut alors plus participer à une autre interaction au cours du même pas de temps, que ce soit en tant que cible ou source. Plus cette mesure sera faible, plus les agents seront passifs, leur état n'évoluant qu'au gré des quelques interactions qui ont été résolues (en considérant que leur état interne n'évolue pas de lui même). D'un point de vue interaction, cette mesure permet donc de révéler des modèles très complexes, comme la vente ou l'achat d'objets, mais qui ne génèrent pas de modifications de l'environnement et de sa représentation.

3.2 Environnement

Modifications de l'environnement Le moteur JEDI fournit un environnement assimilable à un ensemble de cellules que peuvent occuper les agents (les agents disposent de coordonnées réelles [**?**]). L'environnement est représenté graphiquement dans la simulation comme une grille en 2 dimensions, fabriqué à partir des cellules dans lesquelles sont représentés les

agents. Il dispose d'un ensemble de primitives, telles que *Placer un agent* ou *Retirer un agent* de l'environnement, qui nécessitent un renouvellement de sa représentation lors de leur utilisation. Nous proposons donc de mesurer, au pas de temps t, le nombre d'appels à ces primitives de l'environnement, noté $\mathcal{E}(t)$, par rapport au nombre d'agents dans la simulation.

Définition 3.2 ***Modifications***
Au pas de temps t, le nombre de ***modifications*** *des agents est donné par :*
$Modifications(t) = \mathcal{E}(t) \,/\, \mathrm{card}(\mathbb{A}(t))$

Nous obtenons un indicateur de l'évolution de la représentation de l'environnement entre 2 pas de temps. Cette mesure n'est pas toujours bornée : elle n'est comprise entre 0 et 1 que si les interactions résolues ne font appel qu'à une seule primitive de l'environnement nécessitant son rafraîchissement.

Stabilité de l'environnement Cet indicateur est une mesure des points stables de la simulation, effectuée grâce à l'évolution de l'occupation des cases de l'environnement par rapport à chaque famille d'agents. Un grand écart type pour certaines cases montre que celles-ci sont occupées de manière récurrente par la famille d'agents considérée : on peut alors en conclure qu'il s'agit d'un point stable dans la simulation.

Définition 3.3 ***Stabilité***
Au pas de temps t, soient :
- $N_{\mathcal{F}}(t) = card(\mathbb{A}_{\mathcal{F}}(t))$ *le nombre d'agents de la famille $\mathcal{F}$;*
- p *le nombre de cases de l'environnement ;*
- $\mathcal{O}_{c,\mathcal{F}}(t)$ *la présence cumulée d'agents de la famille $\mathcal{F}$ depuis le début de la simulation dans la case c ;*
- $\mathcal{M}_{\mathcal{F}}(t) = \frac{N_{\mathcal{F}}(t) \times t}{p}$ *la moyenne d'occupation des cases de l'environnement.*

L'écart-type d'occupation des cases est :
$\sigma_{\mathcal{F}}(t) = \sqrt{\frac{1}{p} \times \sum_{c=1}^{p} (\mathcal{O}_{c,\mathcal{F}}(t) - \mathcal{M}_{\mathcal{F}}(t))^2}$.

Imaginons un modèle dans lequel il n'y a absoluement aucun mouvement : depuis le départ, chaque agent occupe une case différente, ne peut pas se déplacer dans une autre case et la population Nb est absoluement la même depuis le début de la simulation. Au pas de temps t, l'écart-type le plus défavorable est :

$$\mathrm{def}_{\mathcal{F}}(t) = \sqrt{\frac{1}{p} \times (\sum_{c=0}^{p-Nb} (0 - \mathcal{M}_{\mathcal{F}}(t))^2 + \sum_{c=p-Nb}^{p} (t - \mathcal{M}_{\mathcal{F}}(t))^2)}$$

La ***stabilité*** *est alors définie comme :*
$Stabilite_{\mathcal{F}}(t) = \sigma_{\mathcal{F}}(t) / \sigma\mathrm{def}_{\mathcal{F}}(t)$

Si à l'instant t, la population Nb est à son maximum et, que de sucroît, l'occupation des cellules stagne, l'écart-type va converger vers l'écart-type défavorable. Une simple mesure de la stabilité des agents aurait consisté à déterminer la proportion d'agents restés au même endroit entre 2 pas de temps. Notre mesure tient compte ici de la présence cumulée d'agents depuis le début de la simulation : elle nous permet de révéler des zones de convergence des agents dans l'environnement.

Nous fournissons également 2 indicateurs sur l'environnement : le mélange et la cohésion. Le mélange correspond à la moyenne du pourcentage d'agents d'autres familles dans le voisinage de chaque agent. De même, la cohésion correspond à la moyenne du pourcentage d'agent de la même famille dans le voisinage de chaque agent. L'idée générale est l'idée communément admise du pourcentage de similarité dans le voisinage de Moore, défini comme les huit cellules entourant une cellule centrale centrée sur l'agent considéré.

Définition 3.4 ***Mélange et Cohésion***
Au pas de temps t, soient :
- $\mathcal{N}(x)$ *le voisinage d'un agent x ;*
- $\mathcal{F}(x)$ *la famille dont l'agent est une instance ;*
- $Diff(x) = card(\{a \in \mathcal{N}(x) | \mathcal{F}(a) \neq \mathcal{F}(x)\})$ *le nombre d'agent dans le voisinage d'un agent x dont la famille est différente de la sienne ;*
- $Pareil(x) = card(\{a \in \mathcal{N}(x) | \mathcal{F}(a) = \mathcal{F}(x)\})$ *le nombre d'agent dans le voisinage d'un agent x dont la famille est la même que la sienne ;*
- $CasesP(x)$ *le nombre de cases possibles dans le voisinage d'un agent x ;*
- $Nb = card(\mathbb{A})$ *le nombre d'agents ;*
- p *le nombre de cases de l'environnement.*

Le ***mélange*** *est défini comme :* $Melange = (1/p) \times \sum_{x=1}^{Nb} (Diff(x)/CasesP(x))$ *;*
la ***cohésion*** *est définie comme :* $Cohesion = (1/p) \times \sum_{x=1}^{Nb} (Pareil(x)/CasesP(x))$.

Nous utilisons également une définition classique de la densité.

Définition 3.5 ***Densité***
Au pas de temps t, soient $Nb(t) = \mathrm{card}(\mathbb{A}(t))$ le nombre d'agents et p le nombre de cases de

l'environnement, la ***densité*** *de population est définie comme :* $Densite = \frac{Nb(t)}{p}$.

3.3 Etude de l'évolution des populations et de la résolution des interactions

Usage des interactions Grâce à la méthodologie IODA, la séparation des interactions et des agents nous permet d'enregistrer facilement les « interactions résolues » de chaque entité. L'analyse de ces enregistrements nous révèle le comportement des agents et l'usage global des interactions, tout particulièrement l'apparition de cycle dans leur usage voire un ordre entre elles, comme présentés dans les exemples suivants.

Source \ Cible	Envir.	Végétal	Sauterelle
Végétal	-	-	-
Sauterelle	-	Brouter 1	Dévorer 0

FIG. 4 – Matrice d'interaction d'un modèle simple de recherche de nourriture : *Brouter* a une priorité supérieure à *Dévorer*

Prenons l'exemple de deux familles d'agent *Végétal* et *Sauterelle* ; et 2 interactions *Brouter* et *Dévorer*. Dans Fig.4, les *Sauterelles* vont *Brouter* en priorité les végétaux puis, lorsque les *Végétaux* auront disparu, les *Sauterelles* vont se *Dévorer* entre elles. L'interaction *Dévorer* ne sera effectuée que lorsque il n'y aura plus de végétaux. Imaginons que dans cet exemple, de nouveaux *Végétaux* puissent pousser en cours de simulation en nombre suffisant pour nourrir les sauterelles présentes : l'interaction *Dévorer* ne pourra alors jamais s'opérer, d'où une étape de simplification envisageable dans ce modèle. Bien que nous n'ayons pas connaissance a priori de l'évolution de la simulation, nous pouvons détecter de manière heuristique les interactions qui n'apportent rien au modèle.

Dynamique des interactions Dans un phénomène de rétroaction, le résultat du phénomène considéré agit en retour sur lui même. De telles rétroactions peuvent apparaître dans la résolution des interactions dans nos simulations.

Prenons un exemple simple (cf. Fig.5 gauche) : 3 agents sont placés autour d'une table. Au début de la simulation, un objet *Fleur* est donné à l'un des *Perso*. En cours de simulation, si un agent a une *Fleur* en main, il effectue l'interaction *Donner*, en l'occurence la *Fleur*,

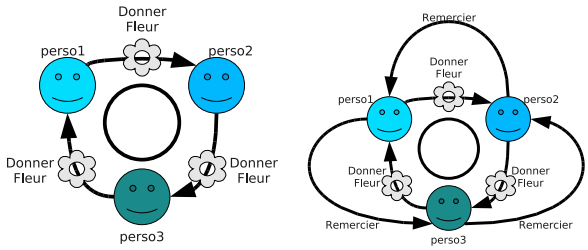

FIG. 5 – Expérience simple de transmission d'objet entre 3 individus : à gauche sans remerciement ; à droite avec remerciement.

à l'un de ses voisins. En considérant un des agents, l'interaction *Donner* l'objet *Fleur* ne sera effectuée qu'une fois tous les 3 pas de temps, le temps que l'objet *Fleur* fasse le tour de la table : le phénomène est périodique.

La mise en place d'un enregistrement des « interactions résolues » pour chaque agent dans le moteur JEDI permet de détecter l'usage cyclique d'interactions qui font parties d'un éventuel phénomène de rétroaction.

Reprenons l'exemple précédent que nous modifions (cf. Fig.5 droite) :
après avoir reçu l'objet *Fleur*, l'agent effectue d'abord l'interaction *Remercier* dont la cible est son voisin donateur, puis effectue au pas de temps suivant l'interaction *Donner*, en l'occurence la fleur, à un autre voisin. Si on étudie les interactions d'un agent, on constate que les interactions *Remercier* et *Donner Objet* sont effectuées tous les 6 pas de temps (2 pas de temps pour chaque personne). *Donner Objet* ne peut être fait qu'après *Remercier* : il y aura donc une phase de $\frac{2\pi}{6}$ entre les deux interactions. Si l'on suit les interactions de manière générale, *Remercier* et *Donner Objet* sont effectuées alternativement une fois à chaque pas de temps.

L'ordre entre *Remercier* et *Donner Objet* est visible, avec un phénomène périodique de période 2 pas de temps. Dans ce cas, nous avons une connaissance de bas niveau sur le scénario modélisé : nous proposons donc l'utilisation d'outils d'analyse fréquentielle afin de déterminer les phénomènes périodiques dans l'usage (fréquence) et l'ordre (phase) des interactions.

TFD En voyant l'usage des interactions comme un signal discret, nous pouvons utiliser la définition classique de la Transformation de Fourier Discrète (TFD) qui nous permet d'étudier les interactions dans l'espace des

fréquences sans contrainte dans le choix des fréquences recherchées.

Définition 3.6 ***TFD***
Soient $\mathrm{s}_{\mathcal{I}}(n)$ *l'évolution de l'utilisation d'une interaction* $\mathcal{I}$ *et* Npt *le nombre de pas de temps utilisés par la TFD, la* ***TFD*** *est définie comme :*
$\mathcal{S}_{\mathcal{I}}(k) = \sum_{0}^{\mathrm{Npt}-1} \mathrm{s}_{\mathcal{I}}(n) \times e^{-2ikn/\mathrm{Npt}}$

Fréquences remarquables Nous proposons la recherche de maxima locaux dans le spectre de fréquence obtenu par transformé de Fourier de l'évolution des interactions.

Définition 3.7 ***Fréquences remarquables***
Soit $\mathcal{S}_{\mathcal{I}}(k)$ *la TFD de* $\mathrm{s}_{\mathcal{I}}(n)$*, l'ensemble* Freq *des* ***fréquences remarquables*** *est :*
$\mathrm{Freq} = \{k / \mathcal{S}_{\mathcal{I}}(k-1) < \mathcal{S}_{\mathcal{I}}(k)$ *et* $\mathcal{S}_{\mathcal{I}}(k) > \mathcal{S}_{\mathcal{I}}(k+1)\}$

Si l'échantillonnage est correctement choisi, nous pouvons ainsi détecter des utilisations périodiques d'une interaction. L'échantillon est limité dans le temps : nous ne pouvons donc trouver que des usages périodiques d'une interaction qui se répète dans cet échantillon. Comme nous ne pouvons faire une simulation durant un temps infini, nous approximons l'usage répété d'une interaction à l'existance d'un cycle.

Nous pouvons dégager 2 niveaux d'analyse dans le suivi des interactions :
– Un suivi macroscopique, c'est-à-dire tenant compte de l'ensemble des interactions qui révèlent la dynamique globale du système. La découverte de fréquences remarquables peut mettre en évidence le couplage de certaines interactions.
– Un suivi microscopique, centré sur un agent, permet de suivre la résolution des interactions. Cette analyse permet de relever de la dépendance entre le comportement de l'agent et ses assignations[9]. Elle se heurte toutefois à l'espérance de vie des agents dans certains modèles (par exemple le modèle proie/prédation). De plus, certaines interactions peuvent désactiver l'agent (au choix du développeur) qui peut être amené à ne subir que des interactions.

L'analyse des fréquences reste à pondérer car certaines interactions peuvent suivre un cadençage, une périodicité qui leur est intrinsèque (et qui est donc indépendante du déroulement de la simulation elle-même).

Analyse de l'évolution de la population Certains modèles, comme le modèle proie/prédateur, vont conduire à la variation périodique des populations, révélateur de phénomènes de rétroaction. La détection de ces phénomènes périodiques est effectuée par analyse de Fourier, comme décrit pour l'analyse des interactions.

4 LEIA : le browser dans l'espace des simulations

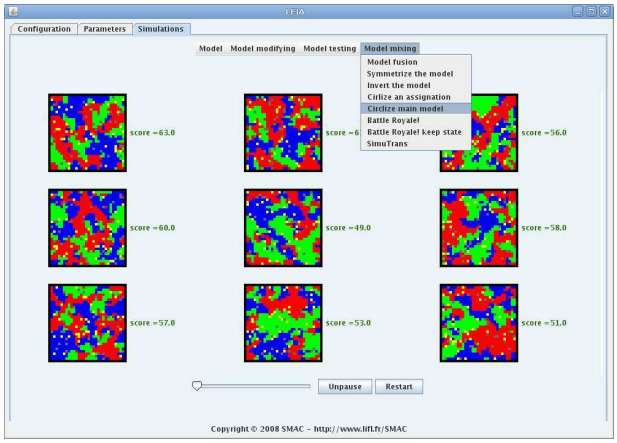

FIG. 6 – LEIA, le navigateur dans l'espace des simulations

Le browser LEIA[3] est une application embarquant n instances du moteur JEDI. Il permet à l'utilisateur de faire instantanément une comparaison visuelle de n simulations en les voyant toutes évoluer en parallèle (Fig.6). Ces n instances sont créées à partir d'un modèle de référence comme dans JEDI Builder. Ce modèle des références est basé sur une ontologie [5], constituée de familles d'agents et d'interactions pré-chargées. Nous pouvons de surcroît modifier le nombre initial d'agents et leur distribution initiale. LEIA est capable de construire automatiquement plusieurs simulations issues de l'espace des simulations associé à l'ontologie de domaine donnée en entrée.

Manipulation du modèle LEIA fournit à l'utilisateur un ensemble d'outils de transformations et de génération de modèle, ainsi qu'un ensemble de jeux de tests afin de parcourir l'espace des simulations. Ces outils permettent de faire varier les paramètres du modèle, que ce soit par l'ajout ou la suppression d'assignations, la modification des priorités

[3] pour *LEIA lets you Explore Interactions for your Agents* `www2.lifl.fr/SMAC/LEIA/applet.html`

ou des gardes de distance, la variation des populations initiales ou des opérations sur la matrice d'interaction toute entière telle que sa symétrisation, voire la fusion des matrices d'interaction de 2 modèles. L'utilisateur peut alors, par ces outils, modifier automatiquement le modèle de référence en générant n sous modèles. Ces modèles sont alors chargés dans les n instances du moteur JEDI.

LEIA peut être utilisé avec autant d'instances que l'on souhaite. Bien entendu, LEIA sera d'autant plus lent qu'un grand nombre d'instances seront chargées. Néanmoins, chaque instance possède sont propre thread afin de profiter des architectures multi-coeurs. Sur une architecture à quatre coeurs, nous faisons fonctionner 4 instances avec les mêmes performances qu'une seule.

Analyse du modèle Le navigateur est secondé par un moteur de statistiques, basé sur les outils de mesures présentés dans la section précédente, afin d'aider l'utilisateur à apprécier les qualités et différences des modèles visualisés. Nous obtenons un retour quantifié de chaque simulation dans lequel nous considérons :

- l'ensemble des interactions résolues par pas de temps par rapport aux interactions envisagées ;
- le nombre de modifications de l'environnement ;
- la répartition des familles d'agents : cohésion et mélange des familles d'agents ;
- l'évolution de l'occupation de l'environnement ;
- évolution remarquable des individus ;
- évolution remarquable des interactions.

Ces mesures heuristiques permettent à l'utilisateur d'accéder à des informations sur chaque modèle grâce par un détail des scores et l'affichage de graphiques mis à jour au fur et à mesure de la simulation.

En plus des outils de mesure déjà présentés, nous pouvons étudier dans le cadre précis de LEIA la construction de la matrice d'interaction du modèle lui-même. Sont remarquables les assignations dites circulaires : c'est-à-dire que pour la même interaction, priorité et garde de distance, les sources et cibles varient de manière cyclique.

Définition 4.1 ***Assignations Cycliques***
Soit $e = (I, p, d)$ *un élément d'assignation.*
On définit : $Assi(e) = \{(\mathcal{S}, \mathcal{T}) \in \mathbb{F}^2 | e \in \mathrm{assi}_{\mathcal{S}/\mathcal{T}}\}$ *l'ensemble des assignations issues d'une matrice d'assignation ayant la même interaction* I*, priorité* p *et garde de distance* d*.*
Si $\mathrm{assi} \in \mathrm{assi}_{\mathcal{S}/\mathcal{T}}$*, alors :*

- $\mathrm{Src}(a) = \mathcal{S}$ *est la famille d'agent étant source de* assi *;*
- $\mathrm{Tgt}(a) = \mathcal{T}$ *est la famille d'agent étant cible de* assi*.*

Les ***assignations cycliques*** *est l'ensemble des couples* $(\mathcal{S}, \mathcal{T})$ *participant à un cycle* e *:*
$\mathrm{Assicyliques}(e) = \{(\mathcal{S}, \mathcal{T}) \in Assi(e) / \exists (\mathrm{assi}_i)_{i \in [1,n]} \subseteq Assi(e) / \mathrm{Src}(\mathrm{assi}_1) = \mathcal{S} \wedge \mathrm{Tgt}(\mathrm{assi}_1) = \mathcal{T} \wedge (\forall i \in [1, n-1], \mathrm{Src}(\mathrm{assi}_{i+1}) = \mathrm{Tgt}(\mathrm{assi}_i)) \wedge \mathrm{Src}(\mathrm{assi}_1) = \mathrm{Tgt}(\mathrm{assi}_n)\}$.
*L'****aspect Cyclique*** *est défini comme :*
$\mathrm{Cyclique} = \mathrm{card}(\mathrm{Assicyliques}) / \mathrm{card}(\mathrm{Assi})$

L'aspect cyclique d'un modèle est donc la proportion des assignations cycliques parmi l'ensemble des assignations. Son étude permet de mettre en valeur des cycles dans la construction du modèle qui peuvent éventuellement se traduire par des boucles de rétroaction. Plus généralement, l'étude du modèle ouvre aussi les perspectives de la simplification automatisée du modèles, dont nous posons les bases dans LEIA en éliminant les interactions inatteignables de par leurs priorités et gardes de distance.

Score d'une simulation Nous fournissons à l'utilisateur un score global afin de réunir les résultats de tous les outils de mesure.

Définition 4.2 ***Score global***
Soit $\mathbb{S} = \{S_1, ..., S_N\}$ *l'ensemble des scores fournis par les outils de mesure (scores compris entre* 0 *et* 100*), le* ***score global*** *est défini comme :*
$ScoreGlobal = 1/N \times \sum_{i=1}^{N} (S_i - 50)$

Nous avons fait le choix de ramener le score global à une note typiquement entre -50 et 50 afin de ne priviligier aucun score par rapport à un autre. La note est centrée sur 0 afin de donner un repère simple à l'utilisateur. Nous ne faisons pas le produit au cas où un score particulier qui serait nul. Le modèle d'une simulation jugée interessante parmi les simulations en parallèle peut être défini comme nouveau modèle de référence par l'utilisateur. Il peut alors réitérer le processus de modification du modèle de référence, que ce soit manuellement ou en utilisant nos outils de transformation.

LEIA permet donc d'explorer l'espace des simulations générées à partir de l'ontologie de domaine embarquée. Il faut bien noter que le navigateur LEIA est ouvert à n'importe quelle ontologie de domaine, tant que les interactions et agents sont conçus selon IODA.

Résultats Les outils que nous présentons permettent à l'utilisateur d'effectuer un travail de rétro-ingénierie sur les simulations en explorant l'espace des simulations. A l'instar du « Continuator » [16] qui exacerbe la créativité musicale, LEIA se révèle être un « chatouilleur d'idées » pour la découverte de nouveaux modèles. Même avec un ontologie simple, avec quelques interactions, les bénéfices tirés du navigateur LEIA sont étonnants, comme nous pouvons le constater avec le « modèle d'infection » présenté ensuite.

L'observation d'une expérience montrant un phénomène de synchronisation entre plusieurs familles d'agents a révélé une organisation intéressante des interactions à l'origine de ce phénomène : la première clone l'agent source dans une case voisine libre ; l'autre tue un agent ciblé. Cet ensemble a été simplifié par une seule interaction : Brièvement, le but de cette nouvelle interaction est de détruire l'agent ciblé trouvé dans le voisinage de la source puis de le remplacer par une copie de la source. Ce modèle, obtenu avec les LEIA, affecte de manière cyclique plusieurs familles d'agent avec cette interaction appelée « Infecter » (Fig.7).

Au moins 3 familles d'agents sont nécessaires pour éviter le blocage de cette simulation. À partir d'une distribution initiale aléatoire des agents, ce modèle conduit les agents à former des spirales d'infection (voir Fig.8 gauche). LEIA révèle ainsi que le « modèle d'infection » ne peut fonctionner sans remplir l'environnement avec un nombre important et égal d'agents de chaque famille. Plus le nombre d'agents est grand, plus grande est la probabilité qu'un agent source trouve un agent cible. De plus, la distance de garde a un réel impact : une plus grande distance pour une interaction implique pour un agent source d'avoir une plus grande probabilité de trouver une cible pour cette interaction. Donc, plus cette limite est grande, meilleures sont les chances de réaliser « Infecter ». Bien entendu, augmenter cette distance permet d'augmenter le nombre de familles d'agents dans ce modèle.

Nous avons également noté la robustesse de

Source \ Cible	Envir.	Rouge	Bleu	Vert
Rouge	-	-	Infecter 0 (1.0)	-
Bleu	-	-	-	Infecter 0 (1.0)
Vert	-	Infecter 0 (1.0)	-	-

FIG. 7 – La matrice d'interactions du modèle d'infection, extensible à n'importe quel nombre de familles d'agents supérieur à 2. « Infecter 0 (1.0) » signifie que l'interaction « Infecter » ne peut se produire qu'à une distance maximale de 1.0.

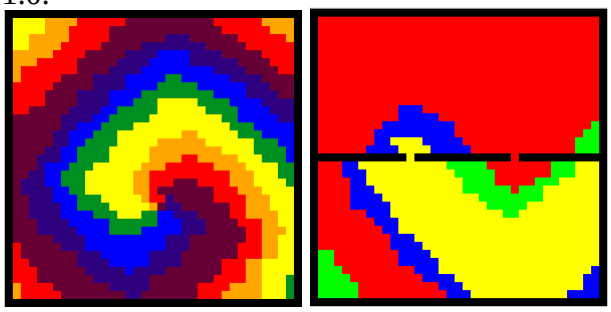

FIG. 8 – Capture d'écran de l'environnement du moteur JEDI utilisant le « modèle d'infection » : à gauche, la formation de spirale entre 7 couleurs ; à droite, ajout d'un obstacle coupant l'environnement en 2 parties sauf en 2 points.

ce modèle à l'ajout d'obstacles dans notre expérience. Ces obstacles sont des agents vides qui ne peuvent intéragir avec d'autres agents de la simulation : ils occupent juste une cellule de l'environnement. Des spirales apparraîssent malgré la présence de ces obstacles et peuvent même faciliter l'apparition de spirale suivant leurs positions, telle que la Fig.8 droite. Cette dynamique[4] est bien connue en chimie dans les réactions cycliques telle que la réaction de Belousov-Zhabotinsky [19, 1]. De tels modèles sont également étudiés dans le domaine des automates cellulaires via le modèle de Greenberg-Hastings [4].

5 Conclusions et Perspectives

Le navigateur dans l'espace des simulations permet la construction itérative de modèles grâce à la méthodologie IODA, qui offre une séparation nette entre agents et interactions permettant leur composition sans génération de code. Nous proposons un ensemble d'outils de transformation et de simplification de modèles qui peuvent être visualisés en parallèle. Nous proposons alors des outils de mesure conçus dans la perspective de IODA. Ces mesures heuristiques mettent en relief certaines

[4] www2.lifl.fr/SMAC/LEIA/spirale.html

caractéristiques remarquables de ces modèles : activité du système, répartition spatiale, stabilité au cours du temps, bouclages, etc. Elles facilitent donc la compréhension des modèles ainsi construits. Nous pouvons donc renverser la vue habituelle de la conception de modèles, en voyant d'abord le résultat puis en étudiant leur comportement.

Par ses outils et la facilité de conception de modèles, le browser LEIA agit de ce fait comme un « chatouilleur d'idées » dont le premier résultat a été de retrouver un modèle à la dynamique similaire au modèle de Greenberg-Hastings. De plus, le browser est ouvert à n'importe quelle ontologie tant que les agents et les interactions sont envisagées suivant la méthodologie IODA : LEIA permet d'explorer l'espace des simulations de domaines aussi variés que la physique ou la biologie. A terme, nous envisageons la construction de modèles par des algorithmes génétiques, modèles dans lesquels la matrice d'interaction peut être vue comme un ensemble de gènes, les outils de mesure permettant l'évaluation de ces modèles et la recherche de caractéristiques remarquables.

Références

[1] B. P. Belousov. A periodic reaction and its mechanism. In *Compilation of Abstracts on Radiation Medicine*, 1959.

[2] Elliot J. Chikofsky and James H. Cross II. Reverse engineering and design recovery : A taxonomy. *IEEE Software*, 07(1) :13–17, 1990.

[3] R Dawkins. *The Blind Watchmaker*. W.W. Norton & Company, Inc., New York, 1986.

[4] R. Fisch, J. Gravner, and D. Griffeath. Metastability in the Greenberg-Hastings Model. In *eprint arXiv :patt-sol/9303005*, pages 3005–+, March 1993.

[5] Thomas R. Gruber. *Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, London, 1993.

[6] Olivier Gutknecht and Jacques Ferber. The MADKIT agent platform architecture. In *Agents Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55, 2000.

[7] D. Hofstadter. *Ma thémagie : En quête de l'essence de l'esprit et du sens*. Intereditions, London, 1988.

[8] J. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.

[9] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. La complexité dans les simulations multi-agents. In *Actes des 15e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2007)*, pages 139–148. Cépaduès, 2007.

[10] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. Formalisation et implémentation des interactions pour la simulation centrée individu. *L'objet*, 14(1-2) :9–33, Janvier-Juin 2008. Numéro spécial Architectures Logicielles.

[11] Yoann Kubera, Philippe Mathieu, and Sébastien Picault. Interaction-oriented agent simulations : From theory to implementation, ECAI 08 July 21-25 2008.

[12] Philippe Mathieu, Sébastien Picault, and Jean-Christophe Routier. Donner corps aux interactions. In *Actes des 4e Journées Francophones sur les Modèles Formels de l'Interaction (MFI'07)*, pages 333–340, 2007.

[13] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system, a toolkit for building multi-agent simulations, 1996.

[14] N. Monmarché, G. Nocent, M. Slimane, and G. Venturini. Imagine : a tool for generating HTML style sheets with an interactive genetic algorithm based on genes frequencies. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, volume 3, pages 640–645.

[15] Philippe Mathieu Nouredine Bensaid. A framework for cooperation in hierarchical multi-agent systems. *Journal of Mathematical Modelling and Scientific Computing*, 8, September 1998.

[16] F. Pachet. De la co-construction d'un langage homme-machine : quelques expériences en musique. In *Actes des 15e Journées Francophones sur les Systèmes Multi-Agents (JFSMA'2007)*, page 9, 2007.

[17] Yves Demazeau Pierre-Michel Ricordel. La plate-forme volcano - modularité et réutilisation pour les systèmes multi-agents. *Technique et Science Informatiques*, 21(4) :447–471, 2002.

[18] Uri Wilensky. Netlogo (and netlogo user manual), 1999.

[19] A. M. Zhabotinsky. Periodic processes of malonic acid oxidation in a liquid phase. In *Biofizika*, 1964.