

Les Jardins de Données : des sociétés d'agents pour La visualisation de données complexes

Guillaume HUTZLER* - Valérie RENAULT*,**

* LIP6, case 169 (OASIS-MIRIAD)
Université Paris 6, CNRS
4, place Jussieu
75 252 Paris cedex 05
téléphone : (33) 01 44 27 87 44
télécopie : (33) 01 44 27 70 00

**Centre National d'Etudes des
Télécommunications
France Télécom
38-40, rue du Général-Leclerc
92 794 Issy Moulineaux Cedex 9

e-mail : {Guillaume.Hutzler, Valerie.Renault}@lip6.fr

RESUME

Les systèmes complexes, et plus généralement les données complexes, posent des problèmes de visualisation importants du fait de leur dynamique, de leur aspect souvent distribué, de la multiplicité des relations et interactions qui les caractérisent, autant d'éléments qui brouillent la compréhension et l'exploitation de ces données. Dans le cadre de la simulation des systèmes complexes, leur visualisation peut pourtant répondre à des besoins d'analyse et de conception, tandis que dans le cadre de systèmes industriels complexes, ce besoin s'exprime en termes de contrôle et de prise de décisions. La solution proposée fait appel aux techniques multi-agents, utilisées de manière à permettre une construction dynamique d'une interface de visualisation. Dans ce cadre, des agents d'interface prennent chacun en charge une petite partie de l'information à visualiser et s'organisent collectivement pour hiérarchiser cette information en fonction des priorités accordées par l'utilisateur aux différentes données.

MOTS CLES

Systèmes complexes, systèmes multi-agents, visualisation et représentation de l'information, hiérarchisation

NOM ET VERSION DU LOGICIEL UTILISE

Microsoft Word 97 pour Windows 95

FORME DE PARTICIPATION

Communication de recherche

THEMES DE LA SOUMISSION

Bases théoriques : modèles des organisations humaines, modèles sociétaux, modèles écologiques

Méthodes, analyse, conception et évaluation : architecture des systèmes interactifs

Dispositifs, techniques et formes d'interaction : interfaces intelligentes

Systèmes, outils : systèmes multi-agents

Types d'utilisation et d'utilisateurs : visualisation et représentation de l'information, interfaces adaptatives

Domaines d'application : systèmes industriels complexes, réseaux, télécommunications, simulation

Les Jardins de Données¹ : des sociétés d'agents pour La visualisation de données complexes

Guillaume Hutzler*, Valérie Renault*,**

* LIP6, Université de Paris 6, case 169
4, place Jussieu, 75 252 Paris cedex 05
{Guillaume.Hutzler, Valerie.Renault}@lip6.fr

**France Télécom R&D,
38-40, rue du Général-Leclerc,
92 794 Issy-les-Moulineaux Cedex 9

RESUME

Les systèmes complexes, et plus généralement les données complexes, posent des problèmes de visualisation importants du fait de leur dynamique, de leur aspect souvent distribué, de la multiplicité des relations et interactions qui les caractérisent, autant d'éléments qui brouillent la compréhension et l'exploitation de ces données. Dans le cadre de la simulation des systèmes complexes, leur visualisation peut pourtant répondre à des besoins d'analyse et de conception, tandis que dans le cadre de systèmes industriels complexes, ce besoin s'exprime en termes de contrôle et de prise de décisions. La solution proposée fait appel aux techniques multi-agents, utilisées de manière à permettre une construction dynamique d'une interface de visualisation. Dans ce cadre, des agents d'interface prennent chacun en charge une petite partie de l'information à visualiser et s'organisent collectivement pour hiérarchiser cette information en fonction des priorités accordées par l'utilisateur aux différentes données.

MOTS CLES : Systèmes complexes, systèmes multi-agents, visualisation et représentation de l'information, hiérarchisation

INTRODUCTION

Les systèmes que l'on qualifie de "complexes" ont ceci de particulier qu'ils ne sont pas observables de manière globale [19]. Cette difficulté provient du grand nombre d'entités qui peuvent interagir les unes avec les autres, selon un grand nombre de modalités différentes, et de manière dynamique. Un système dans son ensemble, la société humaine par exemple, est une abstraction impossible à percevoir ou à se représenter mentalement. Pour étudier ou gérer cette complexité, il pourra alors être utile, voire nécessaire, d'en construire des représentations sensibles qui, en rendant un système perceptible, facilitent son analyse ou son contrôle [8]. En retour, la connaissance acquise par l'analyse du fonctionnement d'un système pourra être mise à profit pour concevoir des systèmes artificiels ayant des propriétés analogues, tandis que la connaissance acquise

par le contrôle du système sera indispensable pour aider à la prise de décisions appropriées.

Dans ces différents contextes d'analyse et de conception d'une part, de contrôle et de décision d'autre part, la représentation visuelle et sonore d'un système constitue un moyen de compréhension essentiel de son fonctionnement. En particulier, elle peut rendre perceptibles les processus par lesquels les actions et interactions des entités composant le système résultent, au niveau global, en des comportements émergents, chaotiques ou évolutifs [3]. La construction de ces représentations pose cependant un grand nombre de problèmes, qui tiennent aux propriétés des systèmes considérés autant qu'aux processus perceptifs des utilisateurs à qui elles sont destinées. Les systèmes complexes sont notamment caractérisés par leur évolution parallèle, distribuée, ouverte, qui introduit une dynamique à tous les niveaux, depuis l'entité la plus élémentaire jusqu'à l'organisation globale, en passant par le système d'interactions qui assure le lien micro/macro [7]. Dans le même temps, la perception humaine est également un processus dynamique par lequel l'utilisateur se construit une représentation mentale du monde qui l'entoure sur la base des informations perçues. Ce processus est lui-même très variable en fonction de l'utilisateur, de ses connaissances, de sa culture, de ses attentes, de sa motivation, de son activité courante ou encore de son attention [18].

Suite à ces différentes constatations, nous aborderons le problème de la visualisation des systèmes complexes comme celui de la construction dynamique d'un médium d'interface plutôt que comme la conception *a priori* d'affichages plus ou moins statiques. Autrement dit, il s'agit de doter un système d'interface des moyens de se réorganiser, en fonction aussi bien des évolutions apparues au sein du système complexe représenté, qu'en fonction de l'utilisateur et de ses préoccupations potentiellement changeantes. Pour ce faire, la solution proposée repose sur l'utilisation de systèmes multi-

¹ Ce travail est soutenu par France Télécom Recherche et Développement, au sein de l'Atelier de Créativité, sous la forme d'un contrat de recherche CTI n° 981B165.

agents [7], eux-mêmes systèmes complexes, composés d'ensembles d'agents logiciels. En dotant ces agents des connaissances nécessaires pour décrire la complexité du système visualisé puis en leur donnant les moyens de récupérer dynamiquement l'information décrivant son évolution, nous montrerons comment une construction dynamique de l'interface est possible. Symétriquement, il sera possible de donner aux agents des connaissances concernant l'utilisateur et de les faire réagir à son action.

Nous commencerons par aborder cette question de manière théorique, ce qui constituera l'objet du deuxième paragraphe. Nous présenterons, au troisième paragraphe, l'architecture logicielle correspondante. Nous montrerons enfin, au quatrième paragraphe, comment des connaissances simples et des stratégies de coopération peuvent être données aux agents de manière à adapter en temps réel l'interface de visualisation, problématique au cœur du projet industriel des *Jardins de Données* [11].

LA VISUALISATION DES SYSTEMES COMPLEXES

Avant d'aller plus loin dans notre réflexion, il est nécessaire de s'arrêter un instant pour préciser ce que nous entendons par « systèmes complexes », analyser les besoins liés à leur représentation visuelle, et positionner notre démarche par rapport aux principales approches de visualisation.

Complexité

Différents systèmes peuvent être qualifiés de complexes [14]. On peut mentionner, pêle-mêle, des systèmes biologiques tels qu'une fourmilière ou l'ensemble d'un écosystème, des systèmes économiques tels que la bourse, des systèmes informatiques tels que le réseau Internet, ou encore des systèmes industriels, un atelier flexible par exemple. Au vu de ces exemples, même s'il n'est pas possible de donner de définition générale et définitive de ce qu'est un système complexe, il est au moins utile de préciser celle que nous utiliserons par la suite. Si l'on peut considérer qu'un système est un ensemble d'entités, en relation les unes avec les autres et organisées au sein d'une structure, nous ajoutons la dynamique comme critère de complexité. En effet, cette composante dynamique se manifeste au niveau des entités, de leurs interactions mutuelles, et donc de l'organisation qui en résulte.

La dynamique intervient au niveau des entités qui pourront être non seulement actives mais également évolutives, aussi bien dans leur comportement que dans leur forme ; c'est le cas par exemple d'une ouvrière au sein d'une fourmilière qui passe par les stades d'œuf, de larve et de cocon avant sa transformation en ouvrière pleinement fonctionnelle, et qui pourra assumer successivement après sa naissance, des tâches de soin aux œufs, de recherche de nourriture et finalement de soin aux cocons [2]. La dynamique intervient ensuite au niveau des interactions qui pourront s'établir entre deux

entités, avant éventuellement de disparaître, et qui pourront elles aussi changer de forme. Dans le cas des fourmis, ces interactions se font et se défont dynamiquement en fonction des tâches, des besoins, des positions des différents individus, et également en fonction de changements globaux intervenant dans leur environnement ou dans la structure des relations sociales qui les lient. [2] La structure sociale de la fourmilière en est elle-même à son tour modifiée, introduisant ainsi la notion de dynamique jusqu'au niveau organisationnel.

Problèmes de visualisation

Outre l'aspect dynamique et la dualité entités/processus déjà évoqués, la construction de moyens de visualisation appropriés pour les systèmes complexes pose encore bien d'autres problèmes. Parmi eux, les plus importants nous semblent liés d'une part à la nature potentiellement distribuée et ouverte de ces systèmes, qui impose une souplesse et une modularité importante, d'autre part à la multiplicité des niveaux d'analyse possibles pour aborder leur étude, qui impose la versatilité des moyens de représentation correspondants.

Le réseau Internet, composé d'un grand nombre d'ordinateurs reliés les uns aux autres, grâce auquel un grand nombre d'utilisateurs peuvent échanger et accéder à un grand nombre d'informations, en constitue un bon exemple. Tout s'effectue de manière totalement distribuée, ce qui nécessite de pouvoir récupérer les données décrivant le fonctionnement du réseau depuis différentes sources et de les intégrer au sein d'un système de représentation unique. Le réseau est par ailleurs ouvert, des ordinateurs, des utilisateurs ou de l'information pouvant y entrer ou en ressortir dynamiquement, ce qui impose de pouvoir également, au niveau du système de représentation, ajouter ou supprimer dynamiquement des sources de données. Les systèmes complexes sont par ailleurs caractérisés par une hiérarchisation en niveaux de complexité [14]. Un écosystème biologique par exemple est constitué d'un ensemble d'organismes animaux et végétaux, eux-mêmes composés d'organes puis de cellules, de molécules, d'atomes, etc. De même une entreprise est elle composée d'usines, d'ateliers, de machines, de pièces, etc. En fonction du niveau d'analyse souhaité dans le cas du système biologique, ou du niveau de contrôle dans le cas du système industriel, l'interface devra elle aussi proposer différents niveaux d'abstraction dans la présentation des informations [13]. Cela devra se traduire par un filtrage et une hiérarchisation permanents de l'information présentée d'une part, de manière à ne pas surcharger l'utilisateur d'informations inutiles, par l'adaptation du mode de représentation suivant la tâche à effectuer d'autre part.

Solutions existantes

Ces différentes contraintes de dynamique, de modularité, de hiérarchisation ne sont pas nouvelles, mais elles ont eu jusqu'à présent un impact limité sur les

interfaces de visualisation proposées. Les solutions existantes relèvent de différents domaines, depuis les interfaces pour les systèmes industriels complexes industriels [13], jusqu'à la simulation des systèmes complexes [17] en passant par la visualisation et la fouille de données [9].

Dans le cadre des systèmes industriels complexes, l'outil de base est le synoptique [13], fournissant une vue synthétique d'un processus, selon un agencement proche de celui de l'agencement physique du processus. Les différents indicateurs associés au fonctionnement du système sont quant à eux repris sous la forme de chiffres, de cadrans, de barre-graphes, etc. Cela ne convient donc que pour des systèmes dont la structure est fixe, et cela ne permet pas en outre, de hiérarchiser les indicateurs en fonction de l'importance qu'ils présentent du point de vue par exemple de la sécurité du fonctionnement du système.

Les domaines de la visualisation et de la fouille de données ont développé un grand nombre de techniques permettant de représenter sous différentes formes, généralement synthétiques, de grands ensembles de données [9]. Par l'utilisation de cartes, d'arbres, de représentations « détail et contexte » que l'utilisateur peut manipuler, de paysages 3D dans lesquels il peut se déplacer, par différentes métaphores permettant à l'utilisateur d'être tout de suite familier avec la structure des informations visualisées, ces techniques offrent un choix très large pour la représentation de données complexes. Elles se limitent malheureusement le plus souvent à la visualisation, à un instant donné, d'un ensemble fixe de données qui résume l'évolution passée d'un système sur une période plus ou moins longue, ce qui ne permet pas leur utilisation dans un contexte dynamique. Le domaine de la simulation des systèmes complexes, et des systèmes multi-agents [7] en particulier, fournit quant à lui toutes les techniques nécessaires à la prise en compte de ces aspects dynamiques mais laisse dans le même temps de côté, à quelques exceptions près, les problématiques de visualisation des systèmes modélisés.

LES JARDINS DE DONNEES : VERS DES INTERFACES DE VISUALISATION MULTI-AGENTS

Nous proposons donc à présent d'étudier dans quelle mesure la synthèse des techniques de simulation multi-agent et de visualisation peut permettre d'apporter des solutions nouvelles au problème de la représentation en temps réel des systèmes complexes. Le projet des *Jardins de Données* est le fruit de cette synthèse, utilisant des écosystèmes virtuels pour filtrer et hiérarchiser des flux de données complexes [11].

Principes généraux

En premier lieu, il est nécessaire de préciser ce que sont les systèmes multi-agents avant d'examiner en quoi ils peuvent constituer une réponse adaptée aux problèmes

abordés. On pourra se contenter de dire qu'un système multi-agent (SMA) est « un ensemble d'agents qui interagissent dans un environnement commun » [6]. L'agent est quant à lui considéré comme « une entité réelle ou virtuelle autonome qui évolue dans un environnement, peut percevoir cet environnement, peut agir dans cet environnement, peut communiquer avec les autres agents, et exhibe un comportement autonome qui est la conséquence de sa connaissance, de ses interactions avec les autres agents et du but qu'elle poursuit » [6]. L'étude des SMAs repose sur un double objectif : d'une part « la réalisation d'artefacts distribués capables d'accomplir des tâches complexes par coopération et interaction », d'autre part « l'analyse théorique et expérimentale des mécanismes d'auto-organisation qui ont lieu lorsque plusieurs entités autonomes interagissent » [7].

Les systèmes multi-agents peuvent ainsi être eux-mêmes considérés comme des systèmes complexes, ce qui explique qu'ils possèdent les bonnes propriétés recherchées pour leur représentation. La dynamique est rendue possible du fait de l'autonomie des agents vis à vis de leur environnement, environnement entendu au sens large, c'est-à-dire incluant un éventuel environnement physique ou simulé, les autres agents et enfin les utilisateurs du système. Cette autonomie permet d'introduire un « jeu », donc une dynamique, dans les relations des agents les uns avec les autres, et également dans l'organisation du système. La modularité est quant à elle une propriété inhérente à ces systèmes, liée à leur décomposition en agents. Enfin la notion d'agent est suffisamment générale pour autoriser différents types de décomposition, permettant de représenter au choix les entités du système ou leurs interactions, et permettant d'adapter le niveau de décomposition au niveau d'abstraction souhaité pour l'interface.

Information – Filtrage – Visualisation

Plus concrètement, l'architecture proposée pour la construction d'une interface de représentation multi-agent (Figure 1) repose sur l'utilisation de deux types d'agents différents : des agents d'information chargés de récupérer les données caractéristiques du système complexe à représenter ; des agents de représentation chargés de filtrer et de hiérarchiser ces données avant de les représenter graphiquement.

Les agents d'informations, associés chacun à une source de données, fonctionnent ensemble comme un système d'information dont le rôle est de récupérer en temps réel les données nécessaires à la description du fonctionnement du système représenté. Ils transmettent ces données au système d'interface proprement dit par l'intermédiaire d'une structure de données partagée, dans laquelle ils peuvent placer des informations à la disposition des agents de représentation. Il faut remarquer au passage que dans ce schéma de

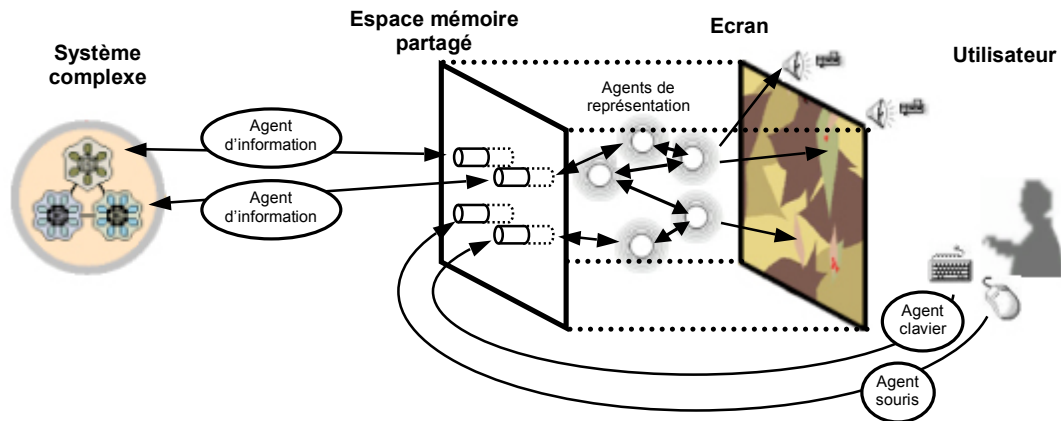


Figure 1 : Architecture générale du système de représentation. Les agents d'informations sont chargés de récupérer les données caractéristiques du système complexe. Ces données sont transmises aux agents de représentation chargés de les filtrer et de les hiérarchiser avant de les représenter graphiquement.

fonctionnement, l'utilisateur du système d'interface est considéré comme étant lui-même un système dont on observe le fonctionnement, notamment par ses actions avec la souris et le clavier. Deux agents d'information sont donc spécifiquement chargés de récupérer les données correspondantes, données auxquelles les agents de représentation pourront réagir, au même titre qu'ils réagiront aux données du système complexe.

Par l'intermédiaire de la zone de mémoire partagée, les agents de représentation récupèrent finalement un flux très important de données, données qu'ils ont à charge de représenter. Dans le cas le plus simple, on pourra associer un agent de représentation à une entité du système complexe représenté, et faire correspondre l'aspect de l'agent au type d'entité. Mais tout l'intérêt de l'approche est de pouvoir doter les agents de connaissances leur permettant d'adapter leur représentation visuelle en fonction de l'état courant des entités, du type de fonctionnement qui les caractérise, des interactions que les entités établissent entre elles, des groupes qu'elles constituent et défont dynamiquement. Symétriquement, l'intérêt de l'approche est de pouvoir doter ces mêmes agents des connaissances leur permettant d'adapter la représentation en fonction des actions de l'utilisateur. En adoptant le point de vue de l'utilisateur, cela revient à dire qu'on lui donne les moyens d'agir directement sur l'interface de visualisation en cours de construction, de manière à orienter la forme finale de la représentation.

La conception des agents d'information est fortement dépendante des sources de données utilisées, et nous ne la détaillerons donc pas. En revanche, le système de représentation proprement dit, depuis la lecture des données dans la zone de mémoire partagée jusqu'à leur affichage à l'écran, est réutilisable quel que soit le type de données visualisé, ce qui permet d'en donner une description plus précise.

Un espace de mémoire partagé

On supposera donc, à cette étape, que des agents d'information alimentent le système de représentation en données, de manière continue. L'espace de mémoire partagée prend la forme d'un ensemble de canaux de communication fonctionnant chacun à la manière de tubes Unix (voir Figure 2). Dans ce contexte, les agents d'information sont producteurs, plaçant les données qu'ils récupèrent dans le canal de communication qui leur est alloué, tandis que les agents de représentation sont consommateurs, s'emparant des données dès qu'elles sont disponibles dans l'un ou l'autre des canaux.

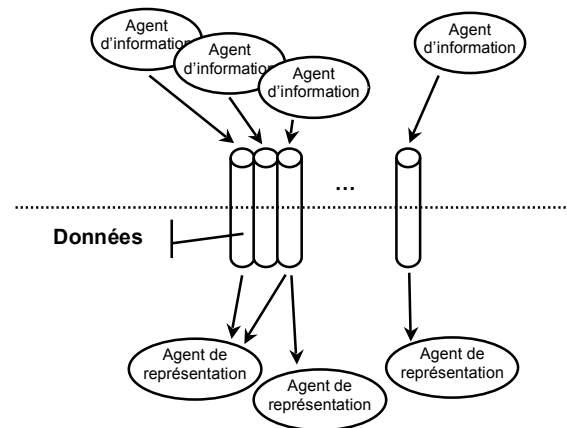


Figure 2 : agents d'information et de représentation échangent les données à afficher par l'intermédiaire d'une structure de données partagée.

Une société d'agents de représentation

Les agents de représentation fonctionnent ensemble comme une société dans laquelle les agents ont une place d'autant plus grande que les informations qu'ils ont à transmettre à l'utilisateur sont importantes ou urgentes. Nous avons choisi d'implanter ce principe sous la forme d'un système de simulation multi-agent

(voir Figure 3). Dans ce cadre, les différents agents de simulation sont caractérisés par leurs ressources propres, ainsi que par des signaux qu'ils sont susceptibles de propager autour d'eux dans leur environnement. L'environnement est lui-même constitué d'un ensemble d'agents organisés selon une grille régulière. On peut penser l'ensemble comme la représentation d'un écosystème, les agents de simulation correspondant à des organismes, végétaux ou animaux, évoluant dans un espace physique représenté ici par le quadrillage de l'environnement.

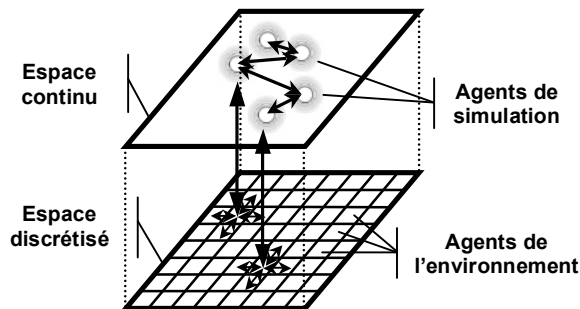


Figure 3 : la partie du système chargée de filtrer et d'afficher les données est conçue comme un système de simulation multi-agent : des agents de simulation sont placés dans un environnement dans lequel ils peuvent interagir, soit par envoi de messages, soit en propageant des signaux autour d'eux.

Les ressources attribuées aux différents agents de simulation correspondent par exemple à un numéro d'identification, la valeur d'une donnée à représenter, le rang hiérarchique de l'agent au sein de la société, une valeur d'énergie, etc. Les signaux qu'ils propagent sont quant à eux le support d'interactions entre ces agents, ce qui leur permet de se « mesurer » entre eux, de manière à déterminer ceux qui doivent avoir un rôle prépondérant dans la construction de la représentation. En fonction des données à représenter par un agent à un instant donné, et également en fonction de ses interactions précédentes avec les autres agents, chacun est capable de déterminer sa place au sein de la société d'agents.

La construction d'une représentation

En fonction de son rang dans la société globale, un agent sera représenté de manière plus ou moins saillante. Un agent dont le rang est élevé, donc a priori important pour l'utilisateur devra en effet être perçu plus facilement que ceux dont le rang est faible. Il s'agit donc de transcrire une hiérarchie d'agents en une hiérarchie picturale qui exprime de manière globale l'importance plus ou moins grande des différentes données.

Pour ce faire, deux stratégies principales sont mises en œuvre. La première consiste à adapter les caractéristiques graphiques d'un agent (sa forme, sa couleur, sa taille, etc.) en fonction de la donnée à

représenter et de son importance relative par rapport aux autres données. La seconde consiste à considérer l'espace graphique comme une ressource pour laquelle les agents se battent, les agents les plus importants gagnant progressivement le centre de l'image et repoussant les autres de manière à ce que la perception de l'information qu'ils transmettent ne soit pas brouillée par la présence à proximité d'autres informations de moindre importance.

La structuration de l'image intervient alors de manière continue par l'arrivée ininterrompue de nouvelles données et leur prise en charge par de nouveaux agents qui devront se battre pour faire reconnaître la pertinence ou l'importance des données qu'ils gèrent.

APPLICATION A LA CONCEPTION D'UNE INTERFACE DE SUIVI DU TRAFIC ROUTIER

L'architecture décrite n'est qu'un cadre permettant de concevoir différents systèmes d'interface multi-agents. Implantée sous la forme d'un noyau de conception de systèmes multi-agents, elle ne prescrit aucune connaissance ni aucun modèle de comportement prédéfini pour les agents de représentation. Ces connaissances et ces comportements sont spécifiés par l'intermédiaire d'un langage de script par lequel le concepteur de l'interface décrit les stratégies de représentation qu'il souhaite voir adopter par les agents. Dans le cas le plus simple, il peut s'agir d'associer un agent au suivi d'une variable, et de lui spécifier d'ajuster sa taille ou sa couleur en fonction de la valeur de la variable. L'inconvénient de cette première approche est que l'on risque alors de se retrouver face à une multitude d'indicateurs graphiques, sans aucuns liens les uns avec les autres, rendant difficile l'obtention d'une vue globale et synthétique du système. Il est donc nécessaire que les différents agents tiennent compte non seulement de l'information qu'ils doivent visualiser, mais également du contexte dans lequel ils se trouvent, c'est-à-dire de l'information que les autres agents doivent eux-mêmes représenter.

L'intérêt de l'utilisation des systèmes multi-agents apparaît dans ce cadre, en donnant la possibilité aux agents d'interagir, de coopérer ou de négocier afin de déterminer dynamiquement les informations les plus importantes à un instant donné. En s'inspirant des processus biologiques de hiérarchisation sociale, présents dans différentes sociétés animales [4], nous allons examiner comment des processus similaires peuvent être implantés au niveau des agents de manière à hiérarchiser le flux de données en fonction de différents critères de priorité. En associant un rang hiérarchique aux agents, traduisant la priorité associée aux données qu'ils représentent, il devient simple d'intégrer l'utilisateur dans le mécanisme de construction de l'interface, en lui offrant la possibilité de définir ses propres priorités, et en laissant le système se

réorganiser dynamiquement à chaque fois que ces priorités changent.

Le rang hiérarchique d'un agent sera finalement fonction de trois types d'informations :

1. les valeurs des données décrivant le système à représenter ;
2. les interactions entre les agents et l'histoire de ces interactions ;
3. les modifications provoquées par l'utilisateur ;

Des données aux agents

L'exemple présenté concerne la visualisation de véhicules en circulation sur une autoroute ; les données issues de ces véhicules sont leur vitesse et leur position. Notre objectif est d'aboutir à une représentation mettant en relief des situations dites « anormales », et donc dangereuses dans ce contexte, tels que des véhicules roulant trop vite ou trop lentement par rapport aux autres.

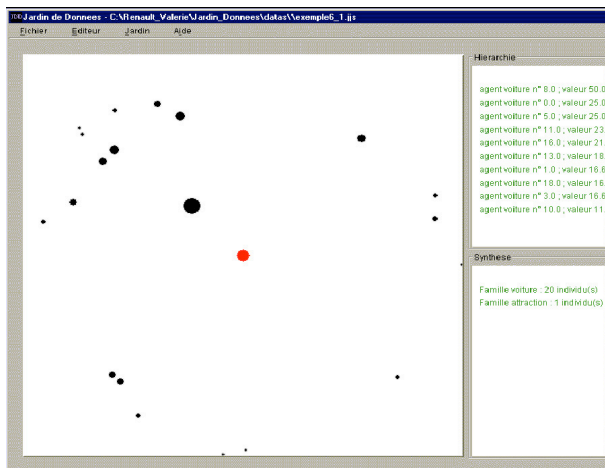


Figure 4 : les agents dont la dangerosité est la plus importante gagnent le centre de l'écran, repoussant les autres vers la périphérie.

Pour conserver une correspondance intuitive entre le système routier et sa représentation, nous avons conservé une correspondance entre agents de représentation et véhicules. En revanche, la position des agents à l'écran ne reflète pas leur position sur le tronçon autoroutier mais correspond à leur degré d'anormalité, lié en l'occurrence à l'écart entre la vitesse

du véhicule et la vitesse moyenne des véhicules qui l'entourent : plus un agent est anormal, plus il sera attiré vers le centre de l'écran, repoussant les agents normaux vers la périphérie (voir Figure 4). De ce fait, l'environnement dans lequel les agents évoluent ne correspond plus à l'environnement physique des véhicules, mais constitue un environnement spatialement structuré dans lequel les agents vont pouvoir interagir et communiquer. En communiquant localement via cet environnement par des stimuli, les agents se hiérarchisent de manière totalement distribuée, en fonction de leur degré d'anormalité.

Hiérarchisation par interaction locale des agents

Pour mettre en œuvre cette hiérarchisation dynamique de la société d'agents de représentation, la démarche, comme souvent dans le domaine du multi-agent, consiste à s'inspirer de modèles éthologiques [3]. Ces modèles décrivent les processus d'organisation à certains moments particuliers de la vie d'un groupe : au moment de la sociogenèse, lors de la mort du leader, ou lors de l'introduction d'un nouveau membre dans le groupe. Le modèle d'organisation développé par P. Hogeweg et B. Hesper [10] nous a servi de point de départ afin de définir les interactions entre nos agents. Ce modèle décrit l'organisation d'une société d'abeilles et la répartition des individus dans la ruche en fonction de leur rang hiérarchique, déterminé au cours de combats. Le script présenté à la Figure 5 montre la traduction de ce modèle éthologique en modèle de comportement pour les agents de représentation. Le calcul du rang d'un agent, en plus de son degré d'anormalité, y est effectué en fonction de l'issue des combats auxquels il participe et de l'histoire des combats passés.

Ces interactions peuvent se traduire ainsi : un agent diffuse un stimulus de dominance dans son environnement (*voit_dominance*), et est capable de détecter les stimuli émis par les autres agents dans son environnement immédiat (*voit_dominance_s*). Les stimuli de dominance sont initialisés de manière à ce que les véhicules ayant le plus haut degré d'anormalité soient les plus dominants et aient le plus de chance de se retrouver au centre de l'écran.

Si les stimuli qu'un agent détecte dépassent son seuil de tolérance (*seuil_tolerance*), un 'combat' a lieu entre

```
voiture.law("calcul_index", 1, 0,
"if ( norme($voit_dominance_s) > $seuil_tolerance) {
  $rand_combat = random();
  if ($rand_combat <= $index_dominance) {$issue_K = 1;};
  if ($rand_combat > $index_dominance) {$issue_K = 0;};
  $voit_dominance = $voit_dominance + ($issue_K - $rand_combat)*0.01;
  $seuil_tolerance = $seuil_tolerance - ($issue_K - 1 + $rand_combat)*0.01;
  $index_dominance = $voit_dominance / (norme($voit_dominance_s) + $voit_dominance);
};");
```

Figure 5 : Extrait du fichier script traduisant l'interaction locale des agents

les agents. L'issue du combat, bien que faisant intervenir un facteur aléatoire, est influencée par l'index de dominance. Ainsi, lors d'un combat entre deux agents de 'force' différente, l'agent initialement le plus 'fort' a plus de chance de gagner que l'agent initialement 'faible', même si l'issue du combat reste incertaine. En fonction de cette issue, les seuils de tolérance et index de dominance sont réajustés pour les deux agents ayant participé au combat.

Hierarchisation par regroupement des données

Une deuxième stratégie visant à hiérarchiser les données consiste à regrouper un ensemble de données qui présentent les mêmes caractéristiques. S'il est important de pouvoir détecter qu'un véhicule a un comportement anormal, inversement il n'est pas utile de distinguer des véhicules différents mais qui ont un comportement similaire. Ainsi, lorsque se forme un train de véhicules, c'est-à-dire un ensemble de véhicules qui se suivent avec une vitesse similaire, il n'est pas nécessaire de représenter tous les véhicules comme autant de points distincts. Il est suffisant de représenter le train comme une seule entité, dont le comportement reproduit celui des véhicules qu'il remplace.

Pour évaluer son degré d'anormalité, un véhicule compare son allure à ceux qui l'entourent. Si au contraire deux véhicules proches ont des allures semblables, les agents qui représentent chacun en créent un nouveau pour représenter l'association des deux véhicules. Cet agent-groupe adopte comme caractéristiques la moyenne des caractéristiques des véhicules composant le groupe, et se comporte dès cet instant comme n'importe quel autre agent-véhicule. A tout moment, de nouveaux agents-véhicules peuvent à leur tour intégrer le groupe s'ils ont un comportement similaire à celui du groupe.

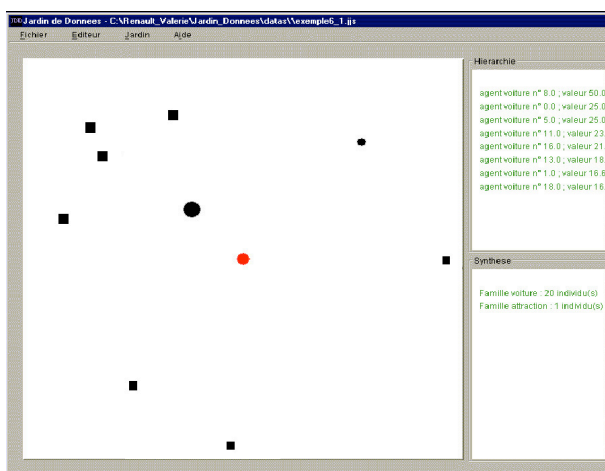


Figure 6 : les véhicules proches et ayant des vitesses comparables sont fusionnés et représentés à l'écran sous la forme d'une seule entité (à comparer à la Figure 4).

Lors de la création d'un agent-groupe, la représentation à l'écran de ce dernier se substitue à celle des agents individuels qui n'apparaissent plus à l'écran (voir Figure 6). Ces derniers ne sont cependant pas supprimés puisque les groupes doivent pouvoir se former et se défaire de manière très dynamique. Les agents-individuels restent donc actifs et vérifient continuellement que leurs caractéristiques demeurent proches de celles de l'agent-groupe auquel ils appartiennent. Lorsque ce n'est plus le cas, ils reprennent leur indépendance et les caractéristiques du groupe sont mises à jour. S'il ne reste plus qu'un seul agent-véhicule dans le groupe, celui-ci est naturellement dissout.

Interaction avec l'utilisateur

Puisqu'il s'agit de visualiser une hiérarchie d'informations plus ou moins prioritaires, il est nécessaire d'offrir à l'utilisateur la possibilité d'informer le système de ce qu'il considère comme prioritaire à un moment précis. L'architecture de la plate-forme permettant de prendre en compte l'utilisateur comme une source de données particulières, les agents peuvent être rendus sensibles aux actions de l'utilisateur avec la souris.

Le principe retenu consiste à doter chaque agent de représentation d'un paramètre interne (`action_utilisateur`) qui va moduler l'interaction ou la répulsion par rapport au centre de l'écran : si l'utilisateur éloigne un agent du centre de l'écran, le degré d'attraction de l'agent vers le centre va être diminué proportionnellement à l'action de l'utilisateur.

Si les fonctions définies dans le script permettent de garder une cohérence interne au système face aux perturbations induites par les données, l'introduction de l'utilisateur comme un agent pouvant interagir avec les autres, permet aussi d'introduire dans le système une cohérence externe : en interagissant et en modifiant le poids de certaines données, l'utilisateur peut modifier la représentation qu'il a du système, et donc maintenir une représentation qui lui est cohérente.

On peut ainsi imaginer que l'utilisateur s'intéresse momentanément à la formation de bouchons et non plus à la détection de véhicules dangereux. Il donnera alors une priorité plus importante aux groupes contenant un grand nombre de véhicules, et dont la vitesse est peu élevée. Par le jeu des interactions entre agents, ceux dont la priorité est ainsi relevée voient du même coup leur position sociale améliorée, ce qui leur permet de gagner le centre de l'écran en repoussant les véhicules jusqu'alors prioritaires.

Autres applications

L'exemple qui vient d'être détaillé ne présente naturellement pas une très grande complexité, mais il permet d'illustrer simplement quelques uns des

problèmes posés par la représentation dynamique de systèmes complexes et de présenter les solutions proposées sur la base de l'utilisation de systèmes multi-agents. Le modèle exposé ne faisant aucune hypothèse quant à la nature des informations représentées, cela permet d'envisager des applications dans des domaines différents.

Nous nous intéressons en particulier au filtrage du courrier électronique. Dans une perspective de classement, les courriers seront rapprochés suivant différents critères d'origine ou de destination, de date, ou encore de sujet. Dans une perspective de traitement, le critère retenu sera associé à la priorité avec laquelle chaque courrier doit être traité. Dans ce cadre, le modèle présenté est développé de manière à faciliter la gestion du courrier par l'utilisateur, à la fois en adaptant l'aspect de l'interface en fonction des priorités attachés par ce dernier aux différents courriers reçus, en les regroupant suivant différents critères, ou encore en autorisant le changement dynamique de niveau de représentation.

CONCLUSION

Nous avons présenté dans cet article un modèle nouveau de construction d'interfaces de visualisation pour des flux de données complexes, modèle qui est implanté dans une plate-forme de conception multi-agent. Ce modèle permet de prendre en compte l'aspect dynamique de ces données, aspect très important bien que souvent négligé. Dans le même temps, les principes exposés autorisent une conception modulaire, adaptative et ouverte du système d'interface. L'exemple proposé, bien que relativement simple, permet cependant de montrer la validité de l'approche consistant à assimiler la construction d'une interface à l'organisation dynamique d'une société d'agents de représentation.

Cette approche doit encore être développée par l'application des solutions à un ensemble de problèmes différents, tels que les systèmes électroniques de courrier ou de dépêches, ou encore les systèmes de bourse en ligne, pour lesquels une vision globale est nécessaire. Le rôle de l'utilisateur devra par ailleurs être renforcé, en donnant au système d'interface les moyens d'apprendre les préférences et les habitudes de l'utilisateur, de manière à adapter le processus de filtrage et de regroupement. A terme, il s'agit de mettre en place une véritable co-évolution entre l'utilisateur et le système d'interface : en adaptant l'interface à l'utilisateur, il s'agit de réduire le besoin de l'utilisateur à s'adapter à l'interface.

Bibliographie

- [1] Arnheim R., *La pensée visuelle*, Flammarion, Paris, 1976.
- [2] Avenier M.-J., « Apports d'un système interactif de représentations graphiques hiérarchisées, pour l'intelligence de phénomènes complexes », in *115^{ème}*

Congrès National des Sociétés Savantes, L'image et la Science (Avignon), pp. 285-302, 1990.

- [3] Bonabeau G. et Theraulaz E. édts., *L'intelligence Collective*, Hermès, Paris, 1996.
- [4] Chauvin R., *Les sociétés animales*, P.U.F., Paris, 1982.
- [5] Chi E. H., Pitkow J., Mackinlay J., Pirolli P., Gossweiler R. et Card S. K., "Visualizing the Evolution of Web Ecologies", in *Proceedings of Computer Human Interaction'98* (Los Angeles), pp. 400-407, ACM Press, 1998.
- [6] Demazeau Y. et Ferber J., "Introduction to Multiagent Systems", ICMAS'98 Tutorial, 1998.
- [7] Ferber J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, Paris, 1995.
- [8] Gelernter D., *Mirror Worlds – The Day Software Puts the Universe in a Shoebox... How it will Happen and What it Will Mean*, Oxford University Press, Oxford, 1992.
- [9] Gershon N., Card S. et Eick S. G., *Information Visualization*, CHI'98 Tutorial Notes n° 3, ACM Press, New York, 1998.
- [10] Hogeweg P. & Hesper B., "The Ontogeny of the Interaction Structure in Bumble Bee Colonies : a MIRROR Model", *Behav Ecol Sociobio*, 12 : 271-283 (1983).
- [11] Hutzler G., « Du Jardin des Hasards aux Jardins de Données : une approche artistique et multi-agent des interfaces homme / systèmes complexes », thèse de doctorat, Université Paris 6, 2000.
- [12] Ishizaki S., "Multiagent Model of Dynamic Design - Visualization as an Emergent Behavior of Active Design Agents", in *Proceedings of CHI '96* (Vancouver), ACM Press, 1996.
- [13] Kolski Christophe, *Interfaces homme-machine – application aux systèmes industriels complexes*, Hermès, Paris, 1997.
- [14] Le Moigne J.-L., *La modélisation des systèmes complexes*, Dunod, Paris, 1990.
- [15] Moukas A., "Amalthaea : Information Discovery and Filtering Using a Multiagent Evolving Ecosystem", in *Proceedings of PAAM'96*, London, 1996.
- [16] Mitchell C. M. et Sundström G. A., "Human Interaction with Complex Systems: Design Issues and Research Principle", in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 27, n°3, pp. 265-273, May 1997.
- [17] Proton H., Bousquet F. et Reitz P., "Un outil pour observer l'organisation d'une société d'agents; le cas d'une société d'agents chasseurs agriculteurs", in *JFIADMSA'97*, J. Quinqueton, M.-C. Thomas et B. Trousse édts, Hermès, 1997.
- [18] Reuchlin M., *Psychologie*, Presses Universitaires de France, 10^{ème} édition, 1993
- [19] de Rosnay J., *Le microscope – Vers une vision globale*, Editions du Seuil, Paris, 1975.
- [20] Tufte E. R., *The Visual Display of Quantitative Information*, Graphics Press, Cheshire CN, 1983.