

SOCIETE D'AGENTS, MODELES ETHOLOGQUES ET VISUALISATION DE DONNEES DYNAMIQUES LEA : APPLICATION A LA VISUALISATION DE COURRIERS ELECTRONIQUES

Valérie RENAULT, Alexis DROGOUL

*Laboratoire d'Informatique de Paris 6 - Pôle IA
U. Pierre et Marie Curie – case 169
4, place Jussieu, F-75252 Paris Cedex 05, France
Mél : {Valerie.Renault, Alexis.Drogoul}@lip6.fr*

Résumé

La diversité des données dynamiques auxquelles un utilisateur est confronté est en perpétuelle augmentation. Le développement d'outils de visualisation est donc de plus en plus nécessaire. Nous proposons d'utiliser des systèmes multi-agents pour concevoir de tels outils. La visualisation des données se traduit par la visualisation d'agents simples et de leurs interactions dans leur environnement. Chaque agent prend en charge une partie de l'information et est doté de capacités de communication rudimentaire. Nous montrons comment l'introduction de mécanismes d'attraction et de répulsion, basés sur des modèles éthologiques élémentaires, peut induire une organisation réactive de données complexes et comment la formation de groupes d'agents aboutit à une synthèse visuelle de l'information. Dans le cadre de l'application LEA, ces principes sont appliqués à la visualisation, en temps réel, du contenu d'une boîte aux lettres électroniques. Chaque agent prend en charge un message et se positionne en fonction des interactions qu'il a avec les autres agents. Ces interactions évoluent dynamiquement selon l'intérêt de l'utilisateur pour certains messages.

Abstract

SOCIETY OF AGENTS, ETHOLOGICAL MODELS AND VISUALIZATION OF DYNAMICAL DATA

As volume of data is growing every day, the development of data visualization tools becomes more and more useful, especially when data are dynamic. We propose to use multi-agent systems to develop such tools. Data visualization is performed as visualization of basic agents and of their interactions. Each agent takes charge of a part of information and has rudimentary means of communication via its environment. We show how by using agents with behaviors based on basic ethological models, information is dynamically organized on the screen. First, we show how the introduction of attractive and repulsive behaviors succeeds in data organizing and screening. Next, we present a new way of visual information synthesis thanks to agent groups. LEA visualizes an electronic mailbox, in real time. Each agent takes charge of an e-mail and it finds a place in its environment according to its interactions with the other agents. These interactions dynamically evolve with the user's interests.

Valérie Drogoul, Alexis Renault

Resumen

*SOCIEDAD DE AGENTES, MODELOS ETOLÓGICOS Y VISUALIZACIÓN DE DATOS DINÁMICOS
LEA : APLICACIÓN A LA VISUALIZACIÓN DEL CORREO ELECTRÓNICO*

La diversidad de los datos dinámicos a los cuales se enfrenta uno va creciendo, lo que hace cada vez más necesario el desarrollo de herramientas de visualización. Proponemos utilizar sistemas multi-agentes para concebir tales herramientas. La visualización de datos se reduce a visualizar agentes simples y sus interacciones dentro de su entorno. Cada agente se responsabiliza por una parte de la información y es dotado de capacidades de comunicación básicas. Enseñamos cómo la introducción de mecanismos de atracción y de repulsión basados en modelos etológicos simples, puede llevar a una organización reactiva de datos complejos y cómo la constitución de grupos de agentes lleva a una síntesis visual de la información. Dentro del marco de la aplicación LEA, estos principios se aplican a la visualización en tiempo real del contenido de un buzón electrónico. Cada agente se encarga de un mensaje y se posiciona según las interacciones que tiene con los demás. Estas interacciones evolucionan dinámicamente según el interés del usuario por ciertos mensajes.

Resumo

*SOCIEDADE DE AGENTES, MODELOS ETOLÓGICOS E VISUALIZAÇÃO DE DADOS DINÁMICOS. LEA
: APLICAÇÃO NA VISUALIZAÇÃO DE CORREIOS ELETRÔNICOS.*

A diversidade dos dados dinâmicos com os quais um usuário se confronta está em crescimento perpétuo. O desenvolvimento de ferramentas de visualização é cada vez mais necessário. Propomos utilizar sistemas multi-agentes para conceber tais ferramentas. A visualização dos dados se traduz pela visualização de agentes simples e de suas interações em seu ambiente. Cada agente está à carga de uma parte da informação e está dotado de capacidade de comunicação rudimentar. Mostramos como a introdução de mecanismos de atração e repulsão, baseados nos modelos etológicos elementares pode induzir uma organização reativa de dados complexos e como a formação de grupos de agentes chega a uma síntese visual da informação. No quadro da aplicação LEA, esses princípios são empregados na visualização, em tempo real, do conteúdo de um correio eletrônico. Cada agente está a cargo de uma mensagem e se posiciona em função das interações que tem com outros agentes. Essas interações evoluem dinamicamente de acordo com o interesse do usuário em algumas mensagens.

Riassunto

*SOCIETÀ D'AGENTI, MODELLI ETOLOGICI E VISUALIZZAZIONE DI DATI DINAMICI. LEA :
APPLICAZIONE ALLA VISUALIZZAZIONE DI MAILS.*

La diversità dei dati dinamici ai quali un utente è confrontato è in perpetuo aumento. Lo sviluppo di strumenti di visualizzazione è dunque sempre più necessario. Proponiamo l'uso dei sistemi multi-agenti per concepire tali strumenti di lavoro. La visualizzazione dei dati si traduce con la visualizzazione d'agenti semplici e delle loro interazioni nel loro ambiente. Ogni agente si assume una parte dell'informazione ed è dotato di capacità di comunicazione rudimentale. Mostriamo come l'introduzione di meccanismi d'attrazione e di repulsione, basati su modelli etologici elementari, può indurre un'organizzazione reattiva di dati complessi e come la formazione di gruppi d'agenti arriva ad una sintesi visiva dell'informazione. Nell'ambito dell'applicazione LEA, questi principi sono applicati alla

visualizzazione, in tempo reale, del contenuto di una mailbox. Ogni agente si assume un messaggio e si piazza in funzione delle interazioni che ha con gli altri agenti. Queste interazioni evolvono dinamicamente secondo gli interessi dell'utente.

1. Introduction

Avec l'essor des nouvelles technologies et l'arrivée d'informations de plus en plus nombreuses et diversifiées, de nouvelles interfaces de visualisation doivent être proposées à l'utilisateur pour lui permettre d'accéder rapidement à une information pertinente à un instant donné. De plus, les interfaces standardisées ont de plus en plus de difficultés à répondre à l'ouverture des technologies à un public de plus en plus large. Il est maintenant nécessaire que les interfaces puissent s'adapter aux besoins des différents utilisateurs. Les interfaces doivent intégrer des processus d'organisation de l'information et des mécanismes d'interactions avec l'utilisateur tout en conservant une certaine autonomie afin de s'adapter à un environnement changeant. L'approche agent est un paradigme peu exploité dans des domaines de recherche tels que les interfaces homme-machine et la visualisation d'informations. Pourtant de nombreux points communs peuvent être mis en évidence entre les caractéristiques des systèmes multi-agents et les besoins des interfaces homme-machine. Par exemple, les sociétés d'agents sont composées de nombreuses entités autonomes et hétérogènes. Les nombreuses interactions entre ces entités conduisent à des organisations capables de s'adapter aux modifications de leur environnement. Malgré ces similitudes, il existe peu d'interfaces multi-agents de visualisation de données (Proctor et Winter, 1998). Les interfaces à base d'agents qui existent actuellement sont généralement conçues à partir d'entités peu nombreuses et relativement complexes (Lieberman *et al.*, 1999). À partir des similitudes entre les objectifs des interfaces de visualisation et les possibilités des sociétés d'agents, nous avons défini une double problématique. D'une part, montrer comment, en s'inspirant de modèles éthologiques, la conception de systèmes multi-agents pour la visualisation d'informations peut conduire à la construction de nouveaux modèles d'organisation de sociétés d'agents. D'autre part, décrire comment une interface multi-agent de visualisation peut répondre à des contraintes propres à la visualisation telles que la présentation organisée d'informations ou l'adaptation dynamique aux actions de l'utilisateur. Nous proposons ainsi une alternative aux interfaces existantes en concevant une interface de visualisation multi-agent basée sur une société d'agents relativement peu évolués, mais dont les nombreuses interactions conduisent à une certaine "intelligence" du système dans sa globalité.

2. Interfaces homme-machine pour la visualisation de données

2.1. Systèmes de visualisation d'informations

Un document textuel nécessite un effort important de compréhension et de mémorisation de la part d'un lecteur qui souhaite en retirer un certain contenu informatif. Assimiler de nouvelles informations, les comprendre et en extraire de nouvelles connaissances nécessitent souvent de pouvoir se les représenter visuellement. Du dessin d'enfant représentant sa famille aux dessins issus de

Valérie Drogoul, Alexis Renault

logiciels d'infographie, de nombreux artefacts existent pour nous permettre de visualiser ou de construire nos propres représentations mentales à partir de stimulations externes. La visualisation d'informations (Jacquemin et Jardino, 2002) est l'un de ces artefacts. Dans Card *et al.* (1999), les auteurs définissent la visualisation comme l'utilisation de l'informatique pour représenter visuellement des données. Les représentations obtenues permettent ainsi à l'utilisateur d'interagir avec ces données et ainsi de mieux les comprendre.

Idéalement, un système de visualisation doit pouvoir posséder plusieurs propriétés. Les premières contraintes imposées à l'interface proviennent des données à visualiser. En effet, le système doit avoir la possibilité de prendre en charge et de représenter des données *nombreuses* ayant des caractéristiques parfois *hétérogènes*. Dans un second temps, il est nécessaire que le système puisse aider à la construction perceptive (Reuchlin, 1993), d'une part en diminuant la charge de mémorisation de l'utilisateur et d'autre part en utilisant des représentations en accord avec les mécanismes sélectifs et les stratégies d'exploration visuelle. C'est le cas des systèmes de cartographie d'informations dans lesquels la représentation spatiale des données induit une interprétation de proximité des caractéristiques de ces données (Honkela *et al.*, 1997). Cette interprétation correspond ainsi à la loi d'organisation perceptive de proximité de la Gestalt (Reuchlin, 1993). Des mécanismes de *structuration et d'organisation visuelle dynamique* de l'information peuvent donc permettre d'orienter la perception de l'utilisateur et de faciliter ainsi sa compréhension des informations visualisées. Cependant, l'utilisateur a une capacité perceptive limitée et il ne pourra pas appréhender un trop grand nombre d'informations présentées de façon simultanée, même si ces informations sont structurées. Il est donc nécessaire que l'interface de visualisation puisse aussi lui fournir une *synthèse visuelle dynamique* de l'information. Les techniques "focus + contexte", telles que les *Fisheye Views* graphiques (Furnas, 1986), permettent de fournir une synthèse visuelle de l'information en représentant la structure globale des données tout en privilégiant la visualisation d'une information particulière. L'organisation et la synthèse peuvent ainsi conduire à des mécanismes de création de *hiérarchies visuelles* de l'information, chaque niveau de cette hiérarchie représentant alors un degré de détail.

De plus, quelles que soient les données, leurs caractéristiques peuvent varier de façon plus ou moins importante dans le temps. Face à cette dynamique, l'interface doit être suffisamment *autonome* pour s'adapter d'elle-même sans que l'utilisateur ait l'obligation de la remettre constamment à jour. La dynamique peut aussi provenir de changements dans les centres d'intérêt de l'utilisateur, ce qui nécessite de la part de l'interface de *s'adapter* à l'utilisateur et d'*apprendre* à partir des actions de ce dernier (Lieberman, 2001).

Enfin, l'interactivité et la manipulation de l'interface éveillant l'attention et l'intérêt de l'utilisateur, différentes possibilités d'*interaction* doivent donc être mises en œuvre. L'utilisateur doit ainsi avoir la possibilité de parcourir l'information en changeant d'angle de vue ou en émettant des requêtes (Lieberman, 1997). Par exemple, lorsque l'interface lui présente une information synthétique, l'utilisateur doit avoir la possibilité d'obtenir les informations ayant permis d'aboutir à cette synthèse, tout en conservant le contexte global de ces données. De plus, la personnalisation de l'interface est un élément important pour l'utilisateur, comme le montre l'organisation des "bureaux virtuels" de chaque personne. Cependant, actuellement, peu de systèmes de visualisation peuvent *s'adapter à une réorganisation dynamique* de l'information par l'utilisateur, tel qu'il peut le faire avec son bureau. Les systèmes de

cartographie d'informations, bien que très proches de la notion de bureau, ne possèdent généralement pas cette caractéristique.

2.2. Interfaces adaptatives

De nombreux travaux sont menés autour des interfaces adaptatives (Ruvini, 2000). Malinowski *et al.* (1992) ont regroupé les interfaces adaptatives en sept types : détection et signal des erreurs, passage d'un type d'interaction à un autre (souris, synthèse vocale, expertise de l'utilisateur, etc.), adaptation de l'information présentée à l'utilisateur (Vernier, 2001), suggestion de commandes adaptées à l'expertise de l'utilisateur, suggestion de paramétrage de fonction, attribution dynamique d'une tâche soit à l'utilisateur, soit à la machine et automatisation des tâches répétitives. Ces interfaces sont définies comme adaptatives par rapport aux actions et à l'expertise de l'utilisateur et ne considèrent pas le cas d'une adaptation aux flux de données.

3. Introduction d'agents dans les interfaces

Les outils de visualisation présentés jusqu'à présent reposent essentiellement sur une approche centralisée du traitement de l'information, i.e. un algorithme centralisé traite et visualise lui-même toutes les données à l'écran. Ces systèmes ne permettent cependant pas toujours de prendre en charge des données hétérogènes, distribuées et dynamiques.

Afin de pallier ces limites, nous proposons une démarche plus structurelle mais aussi plus intuitive dans laquelle ce sont les données elles-mêmes qui s'organisent. Cette approche a été initiée avec la programmation objet, dans laquelle un système est décomposé en éléments plus simples. Un outil de visualisation peut aussi être confronté à deux types de dynamiques : d'une part à la dynamique introduite par des modifications des données (valeur, nombre, type), d'autre part à la dynamique issue de la nécessité que l'interface s'adapte aux actions de l'utilisateur (ce dernier cas recouvrant la notion d'interfaces adaptatives de Malinowski). Il est donc nécessaire que le système soit lui-même dynamique c'est-à-dire qu'il puisse être capable de s'adapter aux modifications et donc de s'auto-organiser. Cette notion d'auto-organisation d'un système à base d'objets nous a ainsi conduit à la notion d'agent et de système multi-agent. Nous proposons ainsi de recourir à un système multi-agent afin de concevoir une interface de visualisation de données capable de s'adapter à la fois aux actions de l'utilisateur et aux flux de données.

3.1. Agents d'interface

Les agents *autonomes* d'interface sont définis comme des agents capables de faire des actions sans intervention de l'utilisateur ou en parallèle avec lui, que l'utilisateur soit oisif ou en train de faire d'autres actions (Lieberman, 1997). Les agents n'attendent donc plus de sollicitation de l'utilisateur pour agir. Ainsi l'utilisateur et les agents interagissent ensemble dans l'interface. C'est le cas, par exemple des collaborateurs ou des secrétaires virtuels capables de réserver un billet d'avion (Morris et Maes, 2000) ou de faire visiter un appartement (Cassell *et al.*, 2000). La plupart de ces interfaces ne se composent que de quelques agents et parfois même d'un seul, le côté "multi" des systèmes multi-agents est souvent négligé et les systèmes agents remplacent les systèmes multi-agents.

3.2. Des agents-fourmis organisant des données

A l'opposé des agents autonomes d'interface, d'autres modèles agents ont été développés. Ils diffèrent notamment par le nombre et la simplicité des agents qu'ils comportent. Ainsi, Resnick (Resnick, 1994) a modélisé la formation de tas par les termites à partir d'agents réactifs ayant des comportements très simples. Les agents sont de deux types : soit des termites, soit des copeaux de bois. Lorsqu'un termite trouve un copeau de bois, il le prend s'il ne porte rien, sinon il pose à côté le copeau qu'il portait. Chaque termite déplace ainsi aléatoirement des copeaux de bois les uns à côté des autres. Il peut cependant aussi lui arriver de défaire un tas commencé par d'autres congénères. Au bout d'un certain nombre d'itérations, le nombre de copeaux isolés diminue et la probabilité de placer un copeau près d'un tas déjà formé augmente. Ainsi, peu à peu, le nombre de tas diminue et la taille de chaque tas augmente. Les termites ne possèdent ni de comportements explicites de construction de tas, ni de plan de l'environnement indiquant où se trouvent les groupes, ni enfin de contrôle centralisé sur le comportement collectif mais ils aboutissent malgré tout au regroupement des copeaux de bois (voir Figure1).

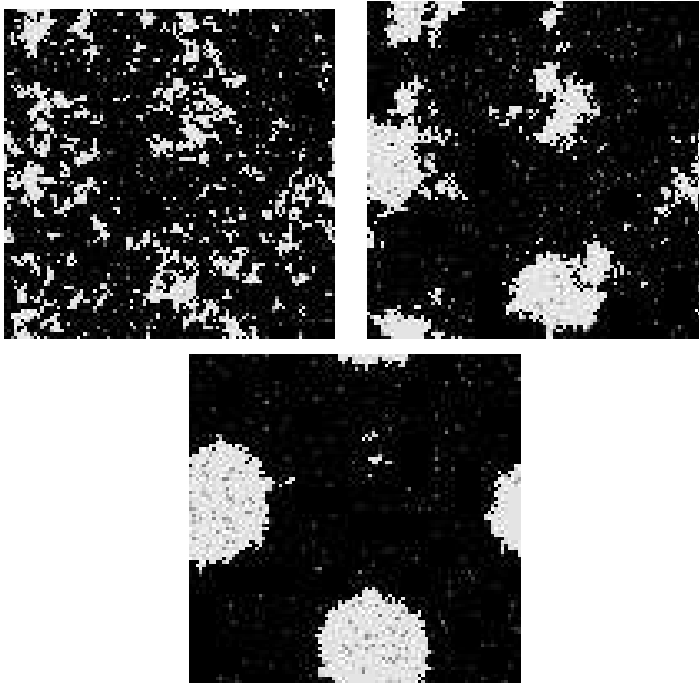


Figure 1 — *Modèle des Termites regroupant des copeaux de bois photos d'écran de StarLogo de Resnick*

Dans la suite des travaux de Resnick, Lumer et Fata (Lumer et Fata, 1994) ont proposé un modèle de regroupement de données à partir d'agents-fourmis. Ce modèle repose sur le tri collectif du couvain par les fourmis. L'environnement est constitué d'une grille à deux dimensions. Comme dans le modèle de Resnick, les agents-fourmis sont distincts des données et agissent sur elles pour les regrouper.

En plus d'être l'une des premières applications de tri collectif de données, ce travail a l'originalité d'avoir testé de nouvelles stratégies afin d'améliorer les performances du système : diversifier la population de fourmis, rajouter des capacités temporaires de mémorisation dans les individus, etc.

Enfin, dans Bonabeau *et al.* (1999), les auteurs généralisent ces mécanismes. Le principe est alors relativement simple : les fourmis qui portent un œuf ou une larve le déposent avec une probabilité d'autant plus grande qu'elles perçoivent d'autres objets du même type à proximité et, inversement, elles s'emparent d'un objet avec une probabilité d'autant plus grande que cet objet est isolé ou différent des objets environnants. Les auteurs transposent ainsi ce modèle à l'organisation de données par des agents-fourmis, chaque agent déplaçant une donnée vers des données similaires. Cependant, ces mécanismes ne permettent pas de positionner *a priori* les données. L'introduction de phéromones localisées permet de résoudre ce problème. En effet, l'organisation spatio-temporelle du couvain et la localisation de phéromones permettent d'influencer le déclenchement d'un comportement d'une fourmi à un lieu donné (à un endroit précis du nid, une fourmi a une probabilité plus grande de déclencher une action particulière) (Bonabeau et Theraulaz, 1994).

Ces modèles montrent les potentialités des systèmes multi-agents basés sur des agents réactifs à traiter des informations plus ou moins complexes. Cependant, ils ne fournissent pas d'interface de visualisation à proprement parler, l'utilisateur a ainsi très peu de moyen d'interagir avec les données. De plus, ces modèles reposent sur une analyse de données statiques et non sur la visualisation de flux de données. Enfin, ces modèles n'utilisent pas toutes les potentialités des agents réactifs et ils n'approfondissent pas l'apport que peuvent avoir des comportements simples d'interaction entre les agents dans l'organisation et la structuration des informations.

4. LEA : Learning E-mail Agents

4.1. Problématique

Afin de pouvoir expérimenter comment les caractéristiques des systèmes multi-agents, et notamment les propriétés d'auto-organisation et d'adaptation, peuvent permettre de structurer, relativement simplement, un ensemble dynamique d'informations à l'écran, nous avons développé l'application LEA (Learning E-mail Agents). LEA est dédiée à la visualisation, en temps réel, d'une boîte aux lettres électroniques. Cette application se base sur un double postulat. D'une part, l'organisation globale de l'interface repose sur les mécanismes d'interactions entre les *agents-données*. D'autre part, l'interface, résultant de la visualisation de l'environnement des agents, permet de structurer et de synthétiser visuellement l'information tout en donnant la possibilité à l'utilisateur d'interagir directement avec les agents.

Le choix de cette application repose sur plusieurs contraintes pour nous permettre de confronter nos hypothèses à des exemples concrets. Tout d'abord, avec l'arrivée de nouveaux messages, le système doit traiter, en temps réel, un flux de données plus ou moins important, tout en s'adaptant aux nouvelles relations qui peuvent exister entre ces données et aux nouveaux centres d'intérêt de l'utilisateur. C'est par exemple le cas lorsque, pendant une certaine période, l'utilisateur reçoit plusieurs messages consacrés à l'organisation d'une session d'examen, alors qu'après la date de l'examen, il sera plutôt intéressé par l'échange de messages qu'il a avec Paul et

Valérie Drogoul, Alexis Renault

aux messages consacrés à son prochain séminaire. Dans un second temps, afin de pouvoir expérimenter les interactions entre l'utilisateur et les agents, il est nécessaire que l'application choisie repose sur des données facilement manipulables et intelligibles auprès d'utilisateurs mais aussi que ces données soient facilement disponibles.

De nombreuses interfaces de messagerie existent dans le commerce. Nous ne prétendons pas ici reproduire l'ensemble des fonctionnalités de ces interfaces et nous nous focaliserons dans la suite de cet article sur la partie *interface de visualisation* des courriers reçus. Notre but n'est pas, non plus, de proposer une alternative aux méthodes classiques de fouilles de données ou de classifications, mais bien de nous situer en aval de ces méthodes en proposant une approche réactive et interactive de visualisation d'informations. De plus, en nous rapprochant de la notion de bureau virtuel ou de cartographie de données, nous proposons une alternative aux interfaces classiques en visualisant les messages sous une forme iconographique et non textuelle comme c'est généralement le cas. Ce dernier choix repose sur l'intention de ne pas présenter à l'utilisateur une liste ordonnée de messages, comme c'est le cas dans les interfaces classiques de messageries ou de moteurs de recherche. En effet, avec les listes l'utilisateur a tendance à considérer les informations dans l'ordre où elles lui sont données, ce qui ne correspond pas toujours à ses priorités.

4.2. Définition des agents

L'application LEA a été développée à partir du noyau multi-agent OSCAR (Renault, 2001) dont l'originalité est de permettre le développement de simulations multi-agents mais aussi la prise en charge et la visualisation de données dynamiques. Afin de répondre aux spécificités liées au domaine de la messagerie, quatre classes correspondant à quatre types d'agents particuliers ont été implémentées. L'*agent-boîte-aux-lettres* gère la connexion au serveur de messagerie et surveille l'arrivée de nouveaux courriers. Lorsqu'un nouveau message arrive sur le serveur, cet agent crée un nouvel *agent-message* auquel il confie le courrier, puis il le rajoute à la session. Le nouvel *agent-message* peut alors interagir avec les autres agents présents dans l'interface : les *agents-messages*, les *agents-dossiers* et l'*agent-poubelle*. Les *agents-dossiers* permettent de regrouper et de conserver certains messages. L'*agent-poubelle* est un dossier unique particulier qui stocke les messages que l'utilisateur souhaite supprimer ultérieurement.

La définition de ces agents permet donc de considérer LEA comme n'importe quel bureau virtuel dans lequel les mails sont assimilés à des fichiers (représentés graphiquement par des enveloppes ou par des enveloppes à bord rouge pour les nouveaux messages) et les groupes de messages à des dossiers (représentés graphiquement par des dossiers). C'est la possibilité d'interaction entre les agents qui va permettre de différencier notre approche, illustrée par LEA, d'un simple bureau de visualisation de courriers.

4.3. Tri collectif auto-organisé des agents-messages

4.3.1. Tri collectif

Comme nous l'avons décrit dans le paragraphe 3.2., des algorithmes de tri collectif, basés sur le modèle de Resnick, permettent de regrouper des informations à l'écran à partir d'agents ayant des comportements relativement simples. La vitesse à

laquelle se forment les groupes d'informations dépend du nombre d'agents " trieurs ". Le tri des morceaux de bois peut alors être fait par un seul agent termite, sans que cela influence le résultat final et la disposition des copeaux de bois. Le qualificatif de tri collectif n'est donc plus forcément le mieux choisi. La différence entre une entité " à trier " et une entité " trieuse " repose uniquement sur leur capacité à se déplacer.

D'un autre côté, dans les modèles de Bonabeau et Dorigo, c'est grâce à l'accumulation des stimuli et donc des données à un endroit précis que va émerger l'organisation des données. Cependant, les stimuli sont émis par les données elles-mêmes, et comme dans le modèle précédent, les fourmis ou les termites n'ont qu'un rôle de " porteurs ".

Nous proposons ici de supprimer les agents " trieurs " ou " porteurs " et de fournir des mécanismes de déplacement aux agents-données eux-mêmes. De même que dans le modèle de Bonabeau et Dorigo, les œufs ou les larves diffusaient des stimuli qui leur permettaient d'être déplacés par les fourmis, nous proposons que ces stimuli guident directement le déplacement de nos agents-données. Ainsi, ces stimuli vont conduire à doter nos agents de mécanismes d'interactions leur permettant de traiter et de comparer simplement les éléments d'informations qu'ils portent.

4.3.2. Interaction entre agents par attraction-répulsion

Nos agents ont été dotés de mécanismes simples d'interactions par attraction-répulsion, ces mécanismes étant basés sur la métaphore des phéromones, stimuli chimiques émis par certains insectes. Un agent réactif peut communiquer avec les autres en diffusant un signal dans son environnement, i.e. l'environnement sert alors de support de communication (Drogoul, 1993). Généralement, l'intensité de la diffusion décroît lorsque la distance à l'agent augmente. Le signal peut soit être rattaché à l'agent, soit être laissé dans l'environnement. Dans un premier cas, cela correspond, par exemple, à un signal de faim qui se déplace avec l'agent. Dans le deuxième cas, le signal peut être associé à une trace laissée dans l'environnement pour permettre à un autre individu de suivre le même parcours. Le comportement de diffusion peut se complexifier avec la présence d'obstacles. Les agents réactifs n'ont, le plus souvent, qu'une représentation partielle de leur environnement. Ils réagissent à leur environnement selon un principe de stimulus/réponse en fonction des stimulations qu'ils reçoivent de leurs congénères. Certaines stimulations peuvent ainsi correspondre à un message répulsif (obstacle) et d'autres à un message attractif (but). L'environnement est ainsi structuré par des zones favorables ou défavorables à la présence de certains agents.

Dans le cas des *agents-messages* de LEA, chaque agent est en charge d'un message et va interagir avec les autres selon les similitudes qu'ils comportent. Ces similitudes reposent sur des mots-clés communs et sont traduites en terme de diffusion de stimuli : deux *agents-messages* ayant des mots-clés communs vont envoyer des stimuli identiques et ainsi s'attirer, alors que les agents ayant des mots-clés distincts vont se repousser. Par le biais de mécanismes d'attraction et de répulsion, les *agents-messages* ont alors la possibilité de se regrouper dans leur environnement et donc visuellement à l'écran.

Plus précisément, l'utilisateur peut définir une liste de mots-clés pour signifier ses centres d'intérêt, ainsi qu'un poids pour chaque mot-clé servant à moduler son importance par rapport aux autres. Il peut aussi choisir de laisser le système détecter automatiquement des mots-clés. Cette détection automatique se fait par la méthode TF-IDF (" Term Frequency / Inverse Document Frequency ") développée par Salton

(Salton et Buckley, 1988). Le poids du mot-clé est alors calculé par rapport à sa fréquence d'apparition. La valeur du poids a un double rôle. Tout d'abord, c'est à partir de cette valeur qu'est calculée l'intensité et la portée du stimulus émis dans l'environnement. Ensuite, c'est en modulant cette valeur que des mécanismes d'apprentissage sont mis en place pour permettre au système de s'adapter aux centres d'intérêt de l'utilisateur. A tout moment, l'utilisateur a la possibilité de modifier, de supprimer ou de rajouter un mot-clé mais aussi d'accepter ou de refuser l'apparition automatique d'un nouveau mot-clé.

Ainsi, lorsqu'un nouveau message arrive sur le serveur, l'*agent-boîte-aux-lettres* parcourt le message et vérifie quels sont les mots-clés présents dans ce message. La détection de mots-clés peut se faire aussi bien dans l'expéditeur, dans le sujet, dans la date ou dans le corps du message. Dans la version actuelle du système, les fichiers attachés ne sont pas pris en compte. À chaque mot-clé trouvé dans le message, un nouveau stimulus est rajouté à l'*agent-message*. Lorsque l'agent est visualisé, il est représenté par un icône ayant une forme d'enveloppe indicé par le mot-clé le plus important contenu dans le message.

Pour chaque mot-clé qu'il possède, l'agent émet donc un stimulus local dont l'intensité correspond à l'intérêt que porte l'utilisateur à ce mot. Plus l'intérêt de l'utilisateur est important (i.e. le poids du mot-clé est important), plus le stimulus va être propagé dans l'environnement. Chaque agent peut donc détecter, dans son voisinage, les agents porteurs de messages ayant des mots-clés identiques aux siens (voir Figure 2). L'agent calcule ainsi son déplacement en fonction des stimuli qu'il reçoit : il est attiré par les agents qui propagent les mêmes stimuli et repoussé par les stimuli différents. Ainsi, les messages ayant des points communs s'attirent mutuellement à l'écran. Inversement, les messages émettant des stimuli différents se repoussent et s'éloignent donc visuellement à l'écran.

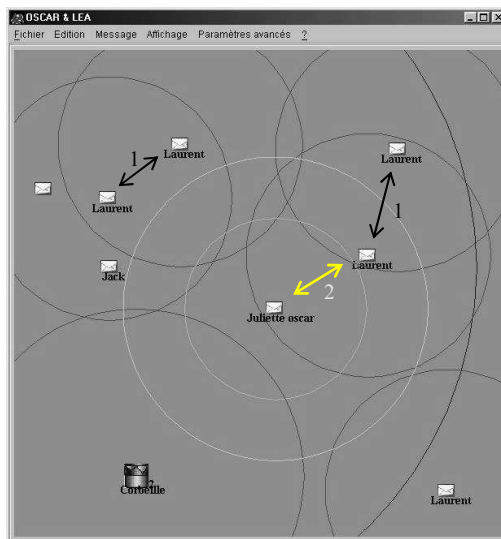


Figure 2 — Visualisation de la diffusion et de la portée des stimuli.
1) indique l'attraction entre des agents-messagers
2) indique la répulsion entre des agents-messagers

4.4. Agents-dossiers : synthèse visuelle dynamique de l'information

Lorsque le nombre de messages devient trop important, le nombre d'agents à visualiser à l'écran augmente, il n'est alors plus possible, pour des raisons d'ergonomie et de lisibilité, de les représenter tous sur le même écran. Il est donc nécessaire de doter le système de mécanismes de synthèse d'informations. Notre proposition consiste à créer des groupes d'agents autour de stimuli communs. Lorsque deux agents sont suffisamment proches et qu'ils possèdent des mots-clés communs, ils construisent un groupe dans lequel ils vont se déplacer et évoluer. L'architecture d'OSCAR (Renault, 2001) permettant de considérer un groupe comme n'importe quel agent, un groupe diffuse dans son environnement le stimulus commun aux agents ayant conduit à sa création. L'intensité de ce stimulus augmente alors proportionnellement avec le nombre d'agents contenus dans le dossier. Lorsque de nouveaux agents, ayant le même mot-clé, arrivent suffisamment près du groupe, ils y entrent à leur tour.

Au niveau de l'application à la messagerie, un groupe correspond à la création d'un dossier permettant de stocker les messages possédant un point commun. Graphiquement, ces agents sont représentés par un dossier. Cette représentation graphique peut contenir d'autres informations telles que le nombre de messages présents dans le groupe ainsi que le nombre de messages non lus. L'augmentation du nombre de messages dans le groupe provoque une augmentation relative de sa taille graphique à l'écran jusqu'à atteindre une limite maximale.

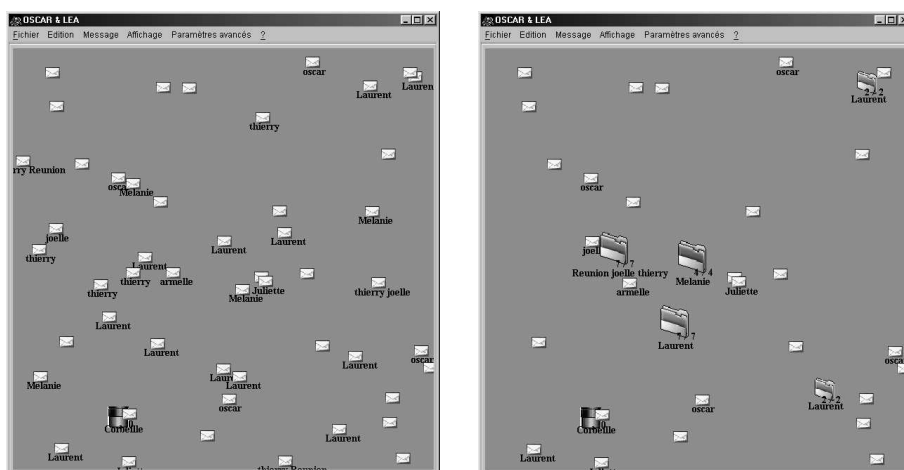


Figure 3 — Synthèse de l'information par regroupement de messages et formation de dossiers : la 1^{ère} photo présente 50 messages disposés aléatoirement au début d'une session ; la 2^{ème} photo montre comment ces messages s'organisent progressivement dans des dossiers.

Grâce à ce mécanisme, les messages sont organisés dans une structure hiérarchique visuelle, de la même façon que les fichiers des ordinateurs sont organisés dans une arborescence. Le dossier constitue alors une représentation synthétique de l'ensemble des éléments qu'il contient. Visualisé par une seule entité graphique, il permet ainsi de réduire et de structurer l'information à l'écran (voir Figure 3).

Valérie Drogoul, Alexis Renault

La diffusion de stimuli locaux peut conduire à la formation de deux groupes suivant le même critère à deux endroits différents de l'environnement et donc de l'écran. Lorsqu'il y a peu d'agents, l'association des agents entre eux se fait alors visuellement par l'utilisateur qui associe les messages ayant le même graphisme (texte, couleur, forme, etc.). Comme dans Proctor et Winter (1998), le système profite donc ici des capacités humaines d'organisation perceptive (Reuchlin, 1993) à associer des informations de même forme ou de même couleur. Lorsque le nombre d'*agents-messages* augmente, les *agents-messages* ont plus de "chance" d'interagir et donc de former des groupes. L'intensité et la portée des stimuli d'un groupe étant proportionnelles au nombre d'agents contenus dans ce groupe, les groupes dont la taille augmente auront à leur tour plus de "chance" d'interagir et de fusionner.

Nous avons jusqu'à présent décrit les comportements des *agents-messages* lorsqu'ils ne comportent qu'un seul mot-clé. Lorsqu'un agent possède plusieurs mots-clés, il peut être amené à se déplacer vers plusieurs groupes. Le choix du groupe pose le problème de la signification de l'appartenance d'un agent à plusieurs groupes distincts en même temps. En faisant abstraction d'une application particulière, un agent, capable d'appartenir à plusieurs groupes en même temps, peut :

- Soit choisir un groupe aléatoirement ou en fonction d'un critère de priorité;
- Soit appartenir à l'un des groupes en ayant un clone de lui-même dans l'autre groupe, la suppression de l'un entraînant alors la suppression de l'autre;
- Soit se dupliquer, chaque copie allant dans un groupe, la suppression de l'un n'entraîne alors pas la suppression de l'autre;
- Soit enfin, créer un nouveau groupe ayant un critère correspondant à l'union des deux critères.

Il n'y a pas de solution meilleure, *a priori*, dans toutes les circonstances. Même dans le cadre d'une application particulière, les utilisateurs peuvent vouloir regrouper leurs informations différemment, ceci nous ayant été confirmé par les tests utilisateurs que nous décrivons dans la prochaine section. Dans le cadre d'une messagerie, certains vont préférer dupliquer les messages, d'autres vont créer un nouveau dossier. Dans le cas de LEA, nous avons choisi de ne pas dupliquer un agent : lorsqu'un agent rentre dans un groupe, ce dernier effectue une synthèse des stimuli contenus dans son environnement interne et propage les stimuli issus de cette synthèse dans son environnement externe. Ainsi, l'information que l'*agent-dossier* contient aussi d'autres stimuli que celui ayant conduit à sa création, est conservée. Les *agents-dossiers* ayant ces stimuli communs peuvent alors se rapprocher, formant ainsi des zones thématiques à l'écran.

4.5. Adaptation à l'utilisateur

Les sections précédentes nous ont permis de montrer comment organiser nos agents sur la base de comportements d'attraction et de répulsion pour qu'émergent à l'écran une structuration et une organisation de l'information. Bien que ne prétendant pas répondre à toutes les demandes d'un utilisateur de messagerie, les fonctionnalités que nous avons choisi de développer offrent cependant un éventail assez large des actions possibles que peuvent effectuer des agents réactifs dans une interface de visualisation. Il est maintenant nécessaire de doter nos agents de

capacités supplémentaires d'interaction pour qu'ils puissent aussi interagir avec l'utilisateur et qu'ils s'organisent ainsi en fonction de ses centres d'intérêt.

4.5.1. Actions élémentaires

Comme sur un bureau virtuel, l'utilisateur a plusieurs possibilités d'action sur les agents. Il peut ainsi :

- Déplacer des agents, les stimuli diffusés par ces agents sont alors émis à partir de leur nouvelle position ;
- Figurer des messages ou des dossiers à une position donnée pour qu'ils ne soient pas attirés par un autre agent ; ce mécanisme permet d'éviter que des agents se déplacent lorsque l'utilisateur a décidé de les positionner à un endroit fixe de l'écran ;
- Lire et supprimer des messages ;
- Rajouter et supprimer des mots-clés, cette action peut alors entraîner une certaine réorganisation dynamique de l'interface.

L'utilisateur peut aussi cliquer sur un dossier afin de l'ouvrir. L'*agent-dossier* possédant son propre environnement interne, l'ouvrir revient donc à faire apparaître les agents qu'il contient, ces agents pouvant à leur tour être organisés en sous-groupes. Visuellement, l'ouverture d'un groupe est associée à un mécanisme de *Fisheye Views* (Furnas, 1986). Ce mécanisme permet à l'utilisateur de garder le contexte présent autour du dossier ouvert. Les messages et les dossiers contenus dans le dossier sont visibles et l'utilisateur a la possibilité d'interagir avec eux (lecture, suppression). Le principe de distorsion par *Fisheye Views* est particulièrement bien adapté à une vision individu centré : il suffit de considérer le point de focus comme l'*agent-dossier* à ouvrir. L'avantage de l'approche agent du *Fisheye Views* est que chaque groupe peut moduler sa distorsion et l'affichage de son environnement en fonction du nombre d'agents qu'il contient. Ce mécanisme permet de se déplacer graphiquement dans l'arborescence visuelle formée par les dossiers.

4.5.2. Apprentissage distribué d'un profil utilisateur

Les actions que peut faire un utilisateur traduisent, implicitement ou explicitement, ses centres d'intérêt à un moment donné. Par exemple, les changements dans les poids des mots-clés indiquent des changements d'intérêt de l'utilisateur. Ces changements peuvent être opérés soit directement par l'utilisateur lorsqu'il modifie les valeurs des mots-clés, soit par le système lui-même en introduisant des mécanismes de variation de poids à partir de certaines actions de l'utilisateur. Ainsi, lorsque l'utilisateur ouvre la fenêtre de lecture d'un message, cela signifie que le message semble important pour lui. Le poids des mots-clés qu'il contient est augmenté de façon inversement proportionnelle à son importance (un mot-clé faible augmente de façon plus importante son poids qu'un mot-clé important). Au contraire, lorsqu'un *agent-message* ou un *agent-dossier* est mis à la poubelle (dans l'*agent-poubelle*), le poids des mots-clés qu'il contient est diminué de façon proportionnelle à son importance (un poids faible est diminué faiblement, alors qu'un poids important est largement diminué). Ainsi, les poids de mots-clés associés à un message lu puis mis à la poubelle diminuent très faiblement. À l'inverse, ceux d'un message lu et gardé augmentent et ceux d'un message directement mis à la poubelle sont plus fortement diminués. Cette diminution du poids lorsqu'un mot-clé n'est plus utilisé ou

Valérie Drogoul, Alexis Renault

qu'il reçoit des renforcements négatifs est proche du mécanisme d'effacement graphique progressif d'un icône à l'écran (Viegas et Donath, 1999) lorsque l'agent n'est plus sollicité.

Si un mot-clé est fortement sollicité à un moment donné, par exemple si lorsque plusieurs messages proviennent d'un même utilisateur (lors d'un échange de mails par exemple), le poids des mots-clés associés (par exemple le nom de cet utilisateur) va alors augmenter. L'interface va ainsi regrouper les nouveaux messages émis lors de l'échange et modifier la configuration existante de l'interface pour les dossiers ou les messages non figés par l'utilisateur : ainsi les messages non figés et ayant des mots-clés communs avec les messages figurant dans cet échange vont se rapprocher et créer ainsi une zone thématique autour de cet échange.

Ces modifications permettent ainsi de suivre l'évolution des centres d'intérêt de l'utilisateur. Les mots-clés et leurs poids constituent ainsi le profil de l'utilisateur. À partir de l'évolution de ce profil, les agents mettent donc plus ou moins en évidence certaines informations. En effet, même avec une session initialisée par des mots-clés aléatoires, le système peut apprendre des actions de l'utilisateur, améliorer son organisation et personnaliser ainsi la représentation qu'il propose. L'utilisateur peut, par exemple, positionner les agents à certains endroits de l'écran selon des repères qui lui sont propres et fixer ces agents à cette position. Lorsque l'utilisateur déplace un dossier, les dossiers ayant des mots-clés communs peuvent aussi se déplacer vers la nouvelle zone choisie par l'utilisateur, sauf s'ils ont été figés. L'utilisateur construit ainsi des zones thématiques sans que ces zones aient nécessairement un sens pour un autre utilisateur. Ensuite, les nouveaux messages ayant des mots-clés communs, viendront se positionner dans ces zones thématiques.

Ce mécanisme est très proche des travaux développés autour de la cartographie d'informations, avec des systèmes tels que *Landscapes* (Wise *et al.*, 1995) et du système *Websom* (Honkela *et al.*, 1997) issu des travaux sur les cartes de Kohonen. Les particularités communes à ces systèmes résident essentiellement dans le fait que ces approches utilisent des mécanismes d'auto-organisation. D'une part, la proximité des données dans l'environnement virtuel est facilement interprétée comme la proximité des caractéristiques des données réelles. D'autre part, ces systèmes donnent à l'utilisateur la possibilité de se "promener" dans le paysage pour explorer l'ensemble des données. Les différences entre LEA et ces systèmes de cartographie résident essentiellement dans la possibilité qu'a l'utilisateur de LEA de modifier dynamiquement les positions des zones thématiques mais aussi dans l'arrivée dynamique de nouvelles informations. En effet, des cartes obtenues avec *Landscapes* et *Websom* sont générées à partir d'une base de données fournie à l'initialisation et elles ne peuvent pas être modifiées par la suite par l'utilisateur. LEA, au contraire, permet à l'utilisateur de construire une cartographie dynamique et personnalisée de sa messagerie, cartographie qu'il peut lui-même modifier à tout instant.

4.6. Tests de concept et premiers résultats

Des tests de concept¹ ont été réalisés afin d'analyser ce que les utilisateurs comprennent intuitivement de l'interface et d'examiner s'ils perçoivent les

¹Ces tests de concept ont eu lieu grâce à la collaboration de Denis Chene et de Naima Lankri, ergonomes à France Telecom R&D. Ils servent à présenter de nouveaux concepts, de

comportements des *agents-messages*. Nous avons ainsi essayé de savoir si les utilisateurs sont conscients de la relative autonomie des éléments d'interface et si cela pouvait les gêner dans la confiance qu'ils accordent au système. Enfin ces tests permettent aussi de faire un premier point sur l'ergonomie de l'interface et ils nous aideront ainsi à en améliorer la lisibilité.

4.6.1. Echantillon d'utilisateurs

Ces tests ont eu lieu sur un échantillon de 10 personnes recrutées par une société d'étude pour France Telecom R&D. Suite à ces tests, LEA a aussi été présentée à une vingtaine de personnes issues du milieu universitaire.

4.6.2. Description du scénario

Après avoir recueilli les caractéristiques des individus (âge, sexe, catégorie socioprofessionnelle, degré de familiarité avec les outils informatiques et les outils de messagerie), nous avons pu distinguer deux groupes d'utilisateurs : les " novices " et les " experts ". Les novices correspondent à des personnes utilisant très peu Internet ou depuis très peu de temps. Ces utilisateurs sont peu familiarisés avec les notions de courriers électroniques. Les experts utilisent couramment Internet et les outils de messagerie dans leur travail et/ou pour leurs loisirs. Ces séances ont été filmées et enregistrées durant le déroulement des tests et des questionnaires.

La première étape consiste à présenter très rapidement l'interface. Il s'agit d'introduire l'équivalence entre un icône "enveloppe" et un message, un icône "dossier" et un ensemble de messages et d'expliquer ce que nous entendons dans la notion de "mot-clé". Cette introduction est présentée le plus simplement et le plus rapidement possible, sans introduire les notions d'agents ni de métaphore éthologique.

Après cette rapide introduction, l'utilisateur est confronté à un nouvel ensemble de messages et de dossiers qu'il doit manipuler lui-même selon un scénario que nous lui donnons au fur et à mesure. L'utilisateur doit, par exemple, trouver un message particulier, soit directement à l'écran, soit en le cherchant dans un dossier. Il doit aussi supprimer ou déplacer des messages et faire face à l'arrivée, en temps réel, de nouveaux messages.

Pour chaque tâche, nous avons collecté l'ensemble des hésitations et des problèmes signalés ou non oralement par les sujets. Les sujets ont aussi pour consigne de dire tout ce qu'ils comprennent ou tout ce qui les gêne, ainsi que ce qu'ils souhaitent voir se réaliser lorsqu'ils font une action particulière. Dans ce dernier cas, ils doivent, par exemple, dire si lorsqu'un message peut être mis dans plusieurs dossiers le message doit se déplacer automatiquement dans le dossier le plus important, se recopier dans chacun des dossiers ou demander l'intervention de l'utilisateur.

Un post-questionnaire est ensuite posé aux sujets afin qu'ils puissent s'exprimer sur l'utilisation de l'interface proposée et sur la compréhension intuitive qu'ils en ont. Ce questionnaire permet aussi de mesurer jusqu'à quel point les utilisateurs peuvent accepter que l'interface trie leurs messages elle-même et donc de connaître le degré d'autonomie qu'ils sont prêts à accepter. Enfin, la notion de regroupement

automatique est comparée avec la notion de filtres telle qu'elle existe dans des outils de messagerie plus classiques.

4.6.3. Résultats

Nous devons rappeler ici que notre but n'était pas de fournir une interface de messagerie finalisée mais d'analyser l'apport des systèmes multi-agents dans le domaine de la visualisation de données. Ainsi, en dehors de quelques difficultés dues au fait que l'interface n'est pas encore finalisée (impossibilité de répondre réellement aux messages reçus, limitation dans le choix des icônes, etc.), les résultats issus des tests de concepts sont très encourageants.

Qu'ils soient novices ou experts, les utilisateurs nous ont tous fait part de leur compréhension rapide et intuitive de l'interface, en particulier au travers de termes proches du domaine des jeux ou de la métaphore du bureau. De même, les déplacements et les regroupements des messages, dans certaines zones de l'écran, ont été rapidement interprétés comme des signaux indiquant des relations entre les messages et les dossiers. Intuitivement, les utilisateurs prêtent des intentions dans le déplacement des *agents-messages* : " ils se sont rapprochés parce qu'ils avaient des points communs ".

Peu de nos utilisateurs, même " experts ", connaissaient et utilisaient les filtres dans leur messagerie quotidienne. Ils ont donc été très intéressés par la formation automatique de dossiers et par le regroupement visuel d'informations. Certains souhaiteraient cependant une séparation plus nette entre les nouveaux messages et ceux déjà lus. L'idée est alors de découper l'interface en deux zones : une pour les nouveaux messages et une pour les autres. En terme d'agents, cela revient à créer deux *agents-dossiers* particuliers toujours ouverts et visibles, un pour les nouveaux messages et un pour les autres. Certains utilisateurs souhaiteraient aussi que les nouveaux messages ne rentrent pas d'eux-mêmes dans un dossier, mais qu'ils restent à proximité des dossiers jusqu'à ce que l'utilisateur les lise.

Des différences existent cependant entre les deux groupes d'utilisateurs. En particulier en ce qui concerne l'autonomie de l'interface, les novices préfèrent déléguer le plus d'actions possibles à l'interface et ne sont donc pas gênés lorsque les *agents-messages* se déplacent d'eux même dans un dossier. Les experts, quant à eux, sont plus partagés : certains font " confiance " au système, d'autres préféreraient que les messages les consultent lorsqu'ils font une action particulière et, dans ce cas là, ils souhaiteraient donc une meilleure lisibilité des actions des agents. Enfin, certains experts, cherchant des références par rapport à l'outil de messagerie qu'ils ont l'habitude d'utiliser, ont exprimé leur souhait de voir afficher certaines informations supplémentaires (liste des messages dans un dossier, date, etc.). Plusieurs experts envisagent la possibilité d'utiliser cette interface comme page d'accueil d'un outil de messagerie qui permettrait de faire le lien vers une interface plus précise ou plus classique.

L'interface n'étant actuellement pas finalisée, ces tests ont été réalisés afin d'analyser les réactions des utilisateurs placés devant une interface autonome basée sur un système multi-agent et non afin de valider l'interface en tant que telle. Il s'agissait aussi d'étudier si les comportements purement réactifs et élémentaires des agents pouvaient être compris de façon intuitive par les utilisateurs. Les résultats montrent que ce but est largement satisfait.

4.6.4. Perspectives

Plusieurs mécanismes sont envisagés afin d'améliorer cette interface. Les études éthologiques montrent que lors de son déplacement, une fourmi laisse une trace (sous forme de phéromone) dans son environnement, cette trace sert alors à guider ses congénères et à lui permettre de retrouver la fourmilière. Cette phéromone se dissipe dans le temps et la trace disparaît progressivement, sauf si elle est renforcée par le passage d'une autre (ou de la même) fourmi. Ce mécanisme, appliqué à LEA, devrait permettre, lorsqu'il y a plusieurs mots-clés dans le même message, de privilégier le déplacement d'un message vers le dossier comportant le mot-clé qui intéresse le plus l'utilisateur à un moment donné. En effet, les traces des derniers messages reçus récemment comportant ce mot-clé devraient favoriser le déplacement du nouveau message vers une zone thématique qui intéresse l'utilisateur à un moment donné.

D'autres modifications de l'interface sont envisagées telles que la disparition progressive des anciens messages, tel que cela se fait dans (Viegas et Donath, 1999) ou le passage à une troisième dimension permettant d'indiquer depuis combien de temps un message est présent dans la boîte aux lettres.

Les tests utilisateurs ont montré les besoins de ce type de système en ce qui concerne l'apprentissage par rapport aux actions de l'utilisateur. Par exemple, certains utilisateurs souhaitent déléguer de nombreuses actions au système et d'autres souhaitent pouvoir contrôler un maximum de chose. Par exemple, les novices préfèrent que le système range les messages directement dans un dossier alors que les experts préfèrent généralement que les messages se groupent autour du dossier jusqu'à leur lecture. Une façon d'aborder ce problème serait de rajouter une notion de seuil d'entrée dans un dossier. En deçà d'un certain seuil, les messages restent à proximité du dossier et au-delà d'une certaine valeur les messages sont autorisés à rentrer. Ces seuils pourraient aussi évoluer en fonction des habitudes des utilisateurs ou du nombre de messages. Afin d'approfondir cette notion d'adaptabilité du système à l'utilisateur, nous orientons aussi nos recherches vers les travaux autour de l'apprentissage par l'exemple développés dans (Leiberman, 2001). La détection des séquences d'action effectuées par l'utilisateur permettrait alors au système d'apprendre s'il faut ou non qu'il duplique ses messages lorsqu'ils interviennent dans plusieurs dossiers ou si les messages doivent ou non rentrer automatiquement dans les dossiers.

Comme nous l'avons déjà indiqué LEA n'a pas la prétention d'être une interface finalisée pour gérer une messagerie. Elle constitue une première plate-forme nous permettant de tester différentes approches et différents mécanismes réactifs pour expérimenter les capacités organisationnelles des agents afin de rendre les interfaces plus interactives et plus dynamiques face aux actions de l'utilisateur. Le fait de choisir une application sur la visualisation de courriers électroniques nous a permis de concevoir ces premières expérimentations en faisant abstraction de la complexité sémantique des données elles-mêmes tout en conservant une certaine complexité due aux nombres de données et de relations pouvant exister entre ces données. Même si l'application semble assez rudimentaire, elle nous a permis de poser les bases de nos premières expérimentations sur l'utilisation d'agents réactifs pour des interfaces de visualisation, et de tester les mécanismes de tri collectif et distribué sur des données réelles.

5. Conclusion

Notre objectif principal était de construire une architecture multi-agent afin d'obtenir un certain nombre de propriétés aptes à constituer le noyau d'une interface de visualisation. LEA fournit un exemple concret permettant de montrer pourquoi les systèmes multi-agents sont des candidats adaptés pour fournir ces propriétés, en particulier grâce aux propriétés d'auto-organisation qui émergent des interactions des agents. Les résultats indiquent que les systèmes multi-agents réactifs se révèlent être des outils particulièrement pertinents pour résoudre des problèmes aussi complexes que la prise en compte en temps réel de données hétérogènes et distribuées, la mise en place d'une représentation visuelle structurée et synthétique de l'information, l'adaptation aux modifications de l'environnement ou la mise en place de mécanismes d'interactions avec l'utilisateur.

Jusqu'à présent, peu d'applications ont été développées pour coupler les caractéristiques des systèmes multi-agents aux besoins des systèmes de visualisation. Il était donc nécessaire de commencer sur un domaine applicatif simple dans lequel nous pouvions contrôler l'ensemble des paramètres. La simplicité de l'application de LEA et la concurrence des outils existants dans le domaine de la messagerie posent problème lorsqu'il s'agit d'effectuer des tests sur le système. En effet, d'une part les utilisateurs ont déjà intégré les mécanismes de fonctionnement des outils de messagerie plus classiques, mais surtout ces outils sont généralement suffisants pour traiter des informations peu complexes telles qu'une liste restreinte de messages. LEA a donc constitué un premier pas vers des applications plus complexes telles que celles qui peuvent se développer autour du Web (visualisation de sites internet, visualisation d'informations circulant sur un réseau, etc.) ou des données issues de la génétique.

Remerciements

Ces recherches ont été menées dans le cadre d'un partenariat entre le Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie et le Studio créatif de France Telecom R&D, Issy les Moulineaux.

Références bibliographiques

- Bonabeau E., Dorigo M., Theraulaz G. (1999). *Swarm Intelligence : From Natural to Artificial Systems*, New York : Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity.
- Bonabeau E., Theraulaz G. (1994). *Intelligence Collective*, Eds Hermès.
- Card S.K., Mackinlay J.D., Shneiderman B. (1999). *Readings in Information Visualization, using Vision to Think*. San Francisco, USA : Morgan Kaufmann Publishers.
- Cassell J., Bickmore T., Campbell L., Vilhjálmsson H., Yan H. (2000). Human Conversation as a System Framework: Designing Embodied Conversational Agents. *In Cassell, J. et al. (éd.) Embodied Conversational Agents*. Cambridge, MA: MIT Press. 29-63.
- Drogoul A. (1993). *De la simulation multi-agent à la résolution collective de problèmes, une étude de l'émergence de structures d'organisation dans les systèmes multi-agents*. Thèse de Doctorat, Université de Paris 6.

Société d'agents, modèles éthologiques et visualisation de données dynamiques

- Furnas G.W. (1986). Generalized fisheye views. *Conference on Human Factors in Computing Systems, CHI'86*. Boston, MA : ACM Press. 16-23.
- Honkela T., Kaski S., Lagus K., Kohonen T. (1997). WEBSOM, self-organizing maps of document collections. *Workshop on Self-Organizing Maps, WSOM'97*. Helsinki, Finland. 310-315.
- Jacquemin C., Jardino M. (2002). Multi-dimensional and Multi-scale Visualizer of Large XML Documents. Eurographics.
- Lieberman H. (2001). Your wish is my command. Programming by example. The Morgan Kaufmann Series in Interactive Technologies.
- Lieberman H., Van Dyke N., Vivacqua A. (1999). Let's browse: a collaborative web browsing agent. *International Conference on Intelligent User Interfaces*.
- Lieberman H. (1997). Autonomous interface agents. *Conference on Human Factors in Computing Systems, CHI'97*. Atlanta : ACM Press.
- Lumer E.D., Faieta B. (1994). Diversity and Adaptation in Populations of Clustering Ants. In Cliff D., Husbands P., Meyer J.-A., Wilson S.W. (éd.) *From animals to animats 3, Third International Conference on Simulation of Adaptive Behavior*. 501-508.
- Malinowski U., Khume T., Dieterich H., Schneider-Hufschmidt M. (1992). A taxonomy of Adaptive User Interfaces. *Proceedings of the HCI'92 Conference on People and Computers VII, Building Adaptive Systems*. 391-414.
- Morris J., Maes P. (2000). Negotiating beyond the Bid price. *Proceedings of the ACM CHI'00 Conference on Human Factors in Computing Systems*. La Hague, Pays-Bas.
- Proctor G., Winter C. (1998). Information Flocking : Data Visualisation in Virtual Worlds Using Emergent Behaviours. In Heudin J.-C. (éd) *First International Conference on Virtual Worlds*. LNAI 1434. Springer-Verlag. 168-176.
- Renault V. (2001). Organisation de Sociétés d'Agents pour la Visualisation d'Informations Dynamiques. Thèse de Doctorat, Université de Paris 6.
- Resnick M. (1994). Learning About Life. *Artificial Life*. volume 1. numéro 1-2.
- Reuchlin M. (1993). Psychologie. Paris, France : Presses Universitaires de France.
- Ruvini J.D. (2000). Assistance à l'utilisation d'un environnement interactif : apprentissage des habitudes de l'utilisateur. Thèse de Doctorat, UFR Sciences de Montpellier.
- Salton G., Buckley C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24. 513-523.
- Vernier F. (2001). La multimodalité en sortie et son application à la visualisation de grandes quantités d'information. Thèse de Doctorat, Université Grenoble 1 Joseph Fourier.
- Viegas F. B., Donath J. (1999). Chat circles. *Proceedings of the ACM CHI'99 Conference on Human Factors in Computing Systems*.
- Wise J.A., Thomas J.J., Pennock K., Lantrip D., Pottier M., Schur A., Crow V. (1995). Visualizing the non-visual : spatial analysis and interaction with information from text documents. *Proceedings Symp. On Information Visualization*, Atlanta, GA. 51-58.

Les auteurs

Valérie Renault est ATER à l'Université de Bordeaux 2. Elle a effectué une thèse sous la direction d'Alexis Drogoul, dans l'équipe MIRIAD du Laboratoire d'Informatique de Paris 6 (LIP6) et un post-doc sous la direction de Philippe

Valérie Drogoul, Alexis Renault

Codognet. Ses travaux portent sur l'introduction d'agents autonomes dans les interfaces homme-machines.

Alexis Drogoul est Professeur en Informatique à l'Université Pierre et Marie Curie, responsable de l'équipe MIRIAD du LIP6, au sein de laquelle il conduit des recherches sur la conception et l'utilisation de systèmes multi-agents dans les domaines de la simulation des systèmes complexes, de la résolution de problèmes et de la robotique collective. Il est membre des comités de programme des conférences majeures dans ces domaines (IJCAI, AAMAS, SAB, DARS, etc.) et expert auprès de la NSF et de la CEE pour les questions relevant des technologies orientées agents.