HAL
open science

Une approche multi-agents pour la gestion de la communication dans les réseaux de capteurs sans fil

Jean-Paul Jamont, Michel Occello

To cite this version:

Jean-Paul Jamont, Michel Occello. Une approche multi-agents pour la gestion de la communication dans les réseaux de capteurs sans fil. Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques, 2006, 25 (5), pp.661-690. hal-00184978

HAL Id: hal-00184978

https://hal.science/hal-00184978v1

Submitted on 1 Feb 2008

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

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

Une approche multiagent pour la gestion de la communication dans les réseaux de capteurs sans fil

Jean-Paul Jamont — Michel Occello

Université Pierre-Mendès France
Laboratoire de Conception et d'Intégration des Systèmes (LCIS-INPG)
Institut National Polytechnique de Grenoble
50 rue Barthélémy de Laffemas
F-26902 Valence

{jean-paul.jamont,michel.occello}@iut-valence.fr

RÉSUMÉ. *Cet article traite de l'utilisation de l'auto-organisation multiagent comme technique de maintien de l'intégrité fonctionnelle d'un système distribué soumis aux contraintes d'un environnement physique agressif. Il présente une approche originale des systèmes multiagents ouverts dans le contexte des réseaux sans fil de capteurs intelligents. De tels réseaux sont constitués d'entités autonomes du point de vue de leur énergie. Ils nécessitent la mise en œuvre de techniques de routage très coûteuses en énergie et devant être fortement tolérantes aux pannes. L'article décrit enfin la plate-forme de simulation qui nous a permis de montrer la qualité de notre solution basée sur l'auto-organisation face à des techniques classiques fondées sur des protocoles de routage* ad hoc.

ABSTRACT. *This paper deals with multiagent self-organization giving adaptive features to a distributed system embodied in an agressive environment. It proposes an original approach of open multiagent systems in the context of wireless networks of intelligent sensors. Such networks are composed by sensors with independant energy sources. These networks need routing techniques which minimize communication energy cost with a strong fault tolerant feature. The paper presents finally the simulation platform, which we built, who allows us to compare our solution based on self-organization and common* ad hoc *routing protocols.*

MOTS-CLÉS : *systèmes multiagents, agents hybrides, auto-organisation, réseaux de capteurs sans fil, environnements physiques agressifs.*

KEYWORDS: *multiagent systems, hybrid agent, self-organization, wireless sensors network, agressive physical environment.*

1. Introduction

L'approche multiagent a montré son intérêt pour les problèmes distribués pour lesquels il est difficile de prévoir toutes les situations rencontrées. Le découpage d'une application en entités décentralisées coopérantes déplace les problèmes d'analyse d'un niveau global à un niveau local et réduit ainsi la complexité de la conception.

Les systèmes fortement distribués mettent en jeu de nombreuses entités, souvent autonomes d'un point de vue énergétique. Ces entités sont à la fois matérielles et logicielles. Elles sont chargées de tâches complexes et de natures très diverses : tâches d'acquisition de mesures, d'action, de comportement, de calcul, de communication. Ces entités sont généralement reliées par des réseaux de communication sans fil. Dans de tels réseaux les liens sont asymétriques, la topologie est dynamique, la bande passante limitée et aucun organe dédié au routage n'est présent. L'ensemble des éléments du réseau participe activement au routage de l'information. Il est donc rendu plus difficile que dans les réseaux filaires traditionnels (Milanosic *et al.*, 2004).

Nous voulons montrer dans cet article que l'approche multiagent permet de gérer efficacement le système complexe que constitue un réseau sans fil de capteurs. Les spécificités de ces systèmes réels immergés dans des environnements agressifs rendent les techniques multiagents encore plus attractives par l'utilisation de l'auto-organisation pour assurer la gestion des communications et un caractère hautement adaptatif. En effet, le maintien de l'intégrité fonctionnelle sera intelligent dans le sens où il prendra en considération les contraintes des constituants élémentaires du réseau qui peuvent évoluer en fonction du niveau d'énergie des capteurs.

Nous précisons tout d'abord quelques notions sur systèmes multiagents ouverts auto-organisés. Nous présentons dans une deuxième partie les réseaux sans fil de capteurs comme une famille particulière de réseaux *ad hoc* et les techniques de communication qui leurs sont associées. Nous analysons ensuite les principaux travaux qui montrent la pertinence de l'utilisation de l'approche multiagent dans le contexte des réseaux sans fil. L'accent sera mis plus particulièrement sur l'aspect émergent de la structure organisationnelle obtenue par des techniques d'auto-organisation. Nous présentons alors, dans une troisième section, notre approche pour la gestion adaptative des communications dans les réseaux sans fil de systèmes physiques autonomes par son application concrète au projet EnvSys. Ce projet a pour but l'instrumentation sans fil de réseaux hydrographiques souterrains afin de déterminer par exemple la présence de polluant et leur évolution. Avant de conclure, nous exposons, dans une dernière partie une analyse comparative des résultats quantitatifs obtenus sur notre plate-forme de simulation.

2. Les systèmes multiagents ouverts et auto-organisés

Nous introduisons dans cette partie le paradigme multiagent. Cette définition passe par l'introduction du concept de système multiagent ouvert. Nous présentons dans un second temps l'auto-organisation comme un outil pour l'émergence.

2.1. *Les systèmes multiagents ouverts*

2.1.1. *Définition*

Les systèmes multiagents. Un système multiagent (SMA) est un ensemble d'agents qui évoluent dans un environnement commun. Cet ensemble d'agents, pas nécessairement intelligents, constitue un système complexe d'où se dégage une intelligence collective. Cette intelligence provient de l'émergence d'un comportement global (Gasser, 1992). L'émergence s'illustre traditionnellement à l'aide, par exemple, de la métaphore éthologique des fourmis où la société de fourmis constituée d'entités ne possédant aucune capacité cognitive atteint pourtant un haut degré d'organisation et d'adaptation.

Un agent est une entité informatique réactive, pro-active et dotée de capacités sociales, capable d'agir de manière autonome dans un environnement (Wooldridge, 1999). La *réactivité* fait référence au maintien d'un lien constant avec son environnement afin de répondre aux changements qui y surviennent. La *pro-activité* signifie que le système génère et satisfait des buts : son comportement n'est donc pas uniquement dirigé par des événements. Les *capacités sociales* indiquent que le système est capable d'interagir ou de coopérer avec d'autres systèmes. Les agents peuvent être décrits à différents niveaux d'abstraction : le *modèle*, l'*architecture* et l'*implémentation* (Wooldridge *et al.*, 1995). Le modèle d'agent décrit l'agent, ses propriétés et comment on peut les représenter. L'architecture est un niveau intermédiaire entre le modèle et l'implémentation. Elle organise la construction de l'agent, de ses capacités et de son contrôle, conformément au modèle. L'implémentation assure la réalisation pratique de l'architecture des agents à l'aide de langages ou outils de programmation.

Le caractère ouvert. Les systèmes multiagents ouverts (Vercouter, 2000) partagent les caractéristiques des systèmes ouverts. Les entités élémentaires du système, que sont les agents, n'ont pas la possibilité d'avoir une représentation complète de l'environnement. Le système dans sa globalité, doit être modulaire et extensible. La modularité concerne le fait que le système multiagent est composé de plusieurs sous-systèmes mis en relation. Ces sous-systèmes ont chacun leur propre mode de fonctionnement. L'extensibilité se traduit par le fait que le système multiagent supporte l'ajout et le retrait dynamique d'éléments. Ces éléments peuvent bien sûr être des agents, mais aussi des rôles, des connaissances ou des liens.

2.1.2. *L'émergence*

Comme nous l'avons constaté précédemment, de la mise en relation des agents situés dans un même environnement et qui interagissent, peuvent émerger différents phénomènes. L'émergence traite de l'apparition soudaine non programmée et irréversible de phénomènes dans un système (éventuellement multiagent) (Jean, 1997). On peut classer en trois catégories les phénomènes émergents (Marcenac, 1996) : l'émergence structurelle, l'émergence de comportements et l'émergence de propriétés. De ces différents types d'émergence nous nous intéresserons par la suite à l'émergence de structures qui sera opérationnalisée par l'auto-organisation.

Il est difficile de qualifier la caractéristique émergente d'un phénomène. Selon les travaux relatés dans (Forrest, 1991; Jean, 1997; Müller *et al.*, 1998) on peut affirmer qu'un phénomène est émergent si :

– on a un ensemble d'agents interagissant *via* un environnement dont l'état et la dynamique ne peuvent pas être exprimés dans les termes du phénomène émergeant à produire mais dans un vocabulaire ou une théorie D ;

– la dynamique des agents en interaction produit un phénomène global comme par exemple une trace d'exécution ou un invariant ;

– le phénomène global est observable soit par les agents (sens fort) soit par un observateur extérieur (sens faible) en des termes distincts de la dynamique sous-jacente D c'est-à-dire un autre vocabulaire ou une théorie D'.

Pour fournir à un système multiagent une fonctionnalité particulière, on peut donc avoir recours à une autre méthode que celle utilisée traditionnellement et qui consiste à procéder à une décomposition fonctionnelle du problème en un ensemble de primitives qui seront implantées dans les agents. L'alternative proposée par l'approche «fonctionnalité émergente» (Steels, 1990; Picard *et al.*, 2004) a pour objectif de faire émerger les fonctionnalités à partir des interactions entre les agents (ou sous-systèmes multiagents dans le cadre de systèmes multiagents imbriqués). Les avantages sont un renforcement de la robustesse du système - il est très adaptatif - et une simplification de la conception - il n'est pas nécessaire que le concepteur envisage toutes les possibilités pour chaque situation.

2.2. *Les systèmes multiagents auto-organisés*

Pour concevoir un système multiagent et ainsi résoudre un problème de manière distribuée il est nécessaire de se poser des questions quand à l'organisation. Dans le cas, par exemple, d'un système multiagents ouvert, il va être nécessaire de lui donner les moyens d'*adapter* son organisation afin de satisfaire au mieux ses objectifs. En effet, si un agent qui avait une tâche bien précise vient à quitter l'organisation, il faut trouver quelqu'un pour le remplacer ou assumer, en plus de ses propres tâches, celles qu'exerçait l'agent. Par « adapter » on entend agir sur les agents ou la structure organisationnelle afin de réorganiser le travail. Il est maintenant acquis que pour les systèmes multiagents utilisant des techniques d'auto-organisation, l'adaptation est le but premier (Ishida *et al.*, 1992). Cette réorganisation sera appelée auto-organisation si elle est prise en charge par l'ensemble des membres de l'organisation. La fonctionalité qui doit émerger est l'organisation comme nous allons le voir au paragraphe suivant avant d'examiner quelques techniques d'auto-organisation et de se pencher sur l'évaluation du processus d'auto-organisation.

2.2.1. *L'émergence de structure au centre de l'auto-organisation*

Comme nous l'avons vu précédemment, les systèmes auto-organisés font émerger des phénomènes issus des interactions entre les divers composants du système (éven-

tuellement un autre système multiagent auto-organisé). Certains (Marcenac, 1997) interprètent cette émergence en distinguant deux de ses aspects : sa forme structurelle et ses caractéristiques.

La forme structurelle concerne l'émergence de structures. Cette émergence est ce qui fait de la construction d'une organisation une auto-organisation. Elle est produite à partir des interactions entre les composants du système. En effet, comme l'illustrent de nombreux travaux (Drogoul, 1993), des phénomènes non programmés et non présents à un instant $t = 0$ naissent de la confrontation d'un environnement actif et d'une activité structurée entre les différents éléments qui composent un système. Le degré de complexité de cette structure émergente va bien au-delà de celle de ses composants et augmente au fur et à mesure que le système s'auto-organise.

Cette émergence repose sur des mécanismes de régulation de l'auto-organisation qui veillent à la cohérence du système et assurent la stabilité du système. Nous les examinons ci-après.

2.2.2. *Mécanismes d'auto-organisation*

Les mécanismes d'auto-organisation permettent aux agents de transformer l'organisation. En un sens ils décrivent les différentes étapes de l'auto-organisation. L'auto-organisation est articulée autour de trois points (Calderoni *et al.*, 1997) qui sont :

– un mécanisme qui permet de détecter les meilleures conditions d'apparition du phénomène d'auto-organisation, de décrire la sensibilité des différents paramètres qui l'influencent. Cette tâche est distribuée sur les différents composants des organisations. Cette opération consiste, au niveau de l'organisation, à trouver les agents qui sont dans un état dit instable (l'agent ne sera stable que si ses aspirations, fixées par le rôle qu'il tient dans l'organisation, sont satisfaites). Les agents ayant en commun certaines caractéristiques sont regroupés et signalent, aux autres agents de la société, par un message (dit de *recomposition*) qu'ils peuvent participer au phénomène émergent ;

– l'apparition du phénomène : ce mécanisme est chargé d'agréger les agents et d'assurer la persistance de la structure. L'interprétation des messages de recomposition conduira à la constitution de la société, l'établissement des relations entre les agents et la création des groupes ;

– l'arrêt du développement du phénomène nécessite de repérer les agents qui sont stables ; il est donc nécessaire de procéder à des observations.

Nous avons identifié onze techniques différentes d'auto-organisation : auto-organisation avec des *compétences réflexes* (Drogoul, 1993), par *contrôle des interactions* (Demazeau, 1993), avec le *W-learning* et *le Q-learning* (Humphrys, 1995), avec *liens préférentiels* (Lee *et al.*, 1995), par la *relaxation* (Camps, 1995), avec *réflexivité* (MARCIA, 1996), par *modification de l'attribution de rôles* (Smith, 1980), par *instanciation de structures organisationnelles* (So *et al.*, 1996), avec *introspection* (Kornman, 1993), par *partage des connaissances* (Ishida *et al.*, 1992). Il est à noter que certains mécanismes hybrides peuvent être utilisés (Guichard, 1996).

2.2.3. *L'évaluation de l'auto-organisation*

Il est important lorsque l'on s'intéresse aux systèmes auto-organisés de penser à la problématique de l'évaluation des résultats. En effet, comme nous l'avons énoncé précédemment, les systèmes auto-organisés sont par essence même tolérants aux fautes et donc particulièrement indiqués lorsque l'on ne peut pas prévoir toutes les situations qu'ils vont être appelés à rencontrer. Il est donc difficile d'évaluer une méthode d'auto-organisation dont la force est justement de s'adapter à des situations imprévues. Nous allons donc, dans cette partie, donner des éléments qui serviront de base aux évaluations des systèmes auto-organisés. Pour cela, nous allons nous intéresser dans un premier temps à l'observabilité de ces systèmes puis, dans un second temps, à l'analyse de ces observations.

2.2.3.1. L'observation de l'auto-organisation

On peut distinguer trois niveaux d'observation d'un système multiagent auto-organisé (PLEIAD, 1992) :

– un niveau externe au système multiagent, si l'on considère le système comme une boite noire dont on ne peut observer que les entrées et les sorties ;

– un niveau interne au système multiagent si l'on prend comme objet d'étude les interactions entre les éléments de la société d'agents. L'observateur est ainsi plongé dans le système et regarde ce qui se passe entre ses différents membres ;

– un niveau interne à l'agent si l'on prend comme objet d'étude les membres de la société et plus précisément leurs architectures.

L'auto-organisation émerge des interactions entre les composants du système. Dans les systèmes multiagents auto-organisés les états stables se traduisent par l'émergence de structures. Si ces structures ne peuvent être identifiées que par un observateur exterieur aux agents mais *interne au système multiagent* il s'agit alors d'une émergence au sens faible (cf. 2.1.2).

2.2.3.2. L'analyse des résultats observés

Dans un premier temps, l'auto-organisation peut être évaluée sans prendre en compte l'application (analyse en termes de processus). Dans un second temps, on peut évaluer la cohérence entre le produit du processus d'auto-organisation et les attentes pour l'application (analyse en termes de résultats).

Analyse en termes de processus. D'une manière générale, pour qu'un processus d'auto-organisation soit intéressant et exploitable, il doit être à la fois *stable*, *sensible* et *convergent* (MARCIA, 1996). La stabilité d'un système fait référence à une constance de celle-ci dans le temps. Le système doit donc être stable afin de mettre en évidence des structures persistantes qui ne se modifieraient que sous l'influence de perturbations (internes ou externes). La sensibilité du processus d'auto-organisation fait référence à sa capacité à faire évoluer une structure vers un état de reconnaissance (partiel ou total) selon que tous les critères retenus aient été satisfaits ou non. Il faut

donc que le processus d'auto-organisation soit sensible afin qu'une remise en cause des structures qui émergent soit possible. La convergence d'un système reflète sa capacité à évoluer vers des structures connues. Le système doit donc être convergent afin que l'on aboutisse à de nouvelles structures.

Analyse en termes de résultats. Comme nous l'avons vu précédemment, cette partie de l'analyse consiste à quantifier l'adéquation entre le processus d'auto-organisation conçu et les résultats espérés. Il nous faudra donc déterminer si le système est *valide* et/ou *pertinent*. Ces deux propriétés seront corrélées avec *l'intérêt* et la *simplicité*. Le système est dit *valide* si le résultat produit par le processus auto-organisé est conforme avec ce que l'on attendait. L'*intérêt* est une grandeur qui estime la variation de cette *validité* en fonction de la complexité d'un raisonnement. Le système est dit *pertinent* si la structure peut être considérée comme simple pour ce qui est de la mise en œuvre et de la compréhension qu'en aurait un observateur. La *simplicité*, à l'image de l'*intérêt*, est une grandeur qui estime la variation de la *pertinence* en fonction de la complexité du raisonnement.

D'une manière générale il faut s'intéresser, lorsque l'on désire établir une bonne analyse, à différentes classes de perturbations. Injecter dans le système des événements perturbateurs crée des dysfonctionnements permettant de mesurer son degré d'auto-adaptation et d'auto-organisation. Cela permet de déterminer ses facultés à atteindre des états stables et ses performances.

3. Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont une famille particulière des réseaux *ad hoc*. Nous commençons par décrire quelques caractéristiques communes à toutes les classes de réseaux *ad hoc* puis nous mettons l'accent sur la problématique propre des réseaux de capteurs sans fil (Akyildiz *et al.*, 2004). Nous détaillons ensuite les techniques de routage utilisées puisque c'est au niveau de la gestion des communications que nous voulons situer notre contribution. Nous consacrons enfin la fin de cette section à une discussion des travaux multiagents dans ce domaine.

3.1. *Caractéristiques des réseaux* ad hoc

Ces réseaux sont composés uniquement de stations, sans contrôle global, qui doivent donc assumer elle-même les fonctions de routage. Ces réseaux doivent s'organiser automatiquement afin d'être opérationnels dès qu'ils sont mis en place (Agha *et al.*, 2002). Lorsqu'ils possèdent des nœuds mobiles on parle alors de réseaux MANET[1] (Mobile *Ad hoc* NETwork). La tendance des recherches actuelles sur ces réseaux tend à intégrer naturellement la notion de mobilité. Dans ces réseaux, contrai-

1. MANET est un groupe de travail issu de l'IETF (Internet Engineering Task Force) qui s'intéresse, depuis peu, à la normalisation des protocoles *ad hoc* fonctionnant sous IP.

rement aux réseaux cellulaires, il n'existe aucune administration centralisée. De plus, aucune contrainte ne limite leur taille en termes d'étendue ou de nombre d'unités. Les applications des réseaux *ad hoc* étaient à l'origine essentiellement militaires mais on les retrouve de plus en plus dans des applications telles que les bases de données parallèles, le télé-enseignement, la simulation distribuée, etc.

On peut dégager six grandes caractéristiques (Lemlouma, 2000; Agha *et al.*, 2002) :

– la topologie est dynamique. C'est une conséquence directe de la mobilité des nœuds. Le tracé des routes peut donc changer pendant l'émission d'un paquet ;

– les liens sont asymétriques. En effet, la liaison entre deux unités n'est pas forcément bidirectionnelle ;

– la bande passante est limitée car les communications par voie hertzienne imposent un partage du médium de communication entre les différents hôtes ;

– les contraintes d'énergie sont fortes. Cela est dû au fait que chacune des unités doit embarquer une alimentation autonome ;

– il n'y a pas d'infrastructure pré-existante et persistante. En effet, les hôtes sont mobiles et c'est à eux qu'il incombe de maintenir les connexions ;

– il y a de nombreuses interférences entre les hôtes du réseau ou encore d'une onde avec elle même (cas d'une réflexion). Elles accroissent le nombre d'erreurs sur la transmission et entraînent un amoindrissement des performances.

3.2. *Problématiques particulières aux réseaux de capteurs sans fil*

De nombreux problèmes sont inhérents à la gestion des réseaux sans fil. Un état de l'art de ces difficultés, aujourd'hui collectivement acceptées, est effectué dans (Zhang *et al.*, 2002). Les réseaux de capteurs sans fil présentent cependant des aspects différents des autres sortes de réseaux ad hoc.

Généralement les éléments du réseau sont indépendants du point de vue de leur énergie et ne peuvent être rechargés de façon simple comme dans les réseaux cellulaires. Cette contrainte forte doit être prise en compte à tous les niveaux de la conception et de la réalisation. Une attention toute particulière est nécessaire lorsque l'on s'intéresse au problème du routage de l'information (l'énergie dépensée lors des communications est une part importante des dépenses énergétiques globales). Le fait d'avoir peu d'énergie peut se traduire au niveau du capteur par l'apparition de fautes internes et peut influer sur divers paramètres tels que la portée des messages que l'on envoie, la qualité et la précision des mesures effectuées etc. Par contre, les nœuds sont généralement stationnaires après une période de déploiement exceptés pour quelques uns d'entre eux. L'environnement hostile (en termes de température, pression, humidité...) dans lequel sont généralement plongés les dispositifs peut aussi contribuer à créer des fautes inattendues et à requérir des re-configurations automatiques et dyna-

miques. L'ensemble peut aussi être soumis à des contraintes temporelles. Dans le cas des réseaux d'entités mobiles la dynamique du système est encore plus accentuée.

3.3. *Les protocoles de routage* ad hoc

Les réseaux sans fil d'entités autonomes font l'objet de recherches récentes et très actives. Un des problèmes les plus importants dans cette classe de réseaux est le routage de l'information. En effet, comme nous l'avons vu précédemment, les fonctionnalités de routage sont réparties sur l'ensemble des entités contrairement aux réseaux traditionnels où cette tâche est dédiée à certains systèmes. Les protocoles de routage associés, que l'on appelle protocoles de routage *ad hoc*, reposent sur les techniques d'inondation. Ces techniques consistent à envoyer un message à tous les membres du réseaux afin d'être sûr que le destinataire a bien reçu le message : un nœud est à la fois récepteur et émetteur du message diffusé. L'inconvénient majeur de l'inondation est d'être très coûteuse en énergie. De plus, elle entraîne souvent un faible rendement c'est-à-dire que le rapport entre le volume des messages utiles sur le volume total transmis est faible. Il existe trois familles de protocoles de routage *ad hoc* :

– les protocoles *ad hoc réactifs* qui ne cherchent des routes que lorsque cela est nécessaire (on-demand protocol) c'est-à-dire lorsqu'un message doit être transmis. Ces protocoles utilisent beaucoup les techniques d'inondation car lorsqu'une station souhaite transmettre des données, elle ne sait pas encore le chemin qu'il faut prendre pour joindre le destinataire. Un des avantages majeurs de cette famille de techniques de routage est le fait que la bande passante n'est essentiellement utilisée que pour la transmission de données. Le protocole Ad hoc On Demand Distance Vector (AODV (Royer *et al.*, 1999)) et le protocole Dynamic Source Routing (DSR (Johnson *et al.*, 1996)), que nous verrons par la suite, sont des membres des plus représentatifs de cette famille ;

– les protocoles *ad hoc proactifs* qui nécessitent l'utilisation d'une table de routage mise à jour continuellement *via* l'échange de paquets de contrôle. En quelque sorte, ils créent un modèle du réseau (généralement partiel). Le protocole Destination Sequence Distance Vector (DSDV (Perkins *et al.*, 1994)) est l'un des premiers protocoles à avoir été spécifié par le groupe de travail MANET et il utilise le même principe de fonctionnement que RIP. Le protocole Optimized Link State Routing protocol (OLSR (Clausen *et al.*, 2001)), qui utilise des règles de routage centrées lien, et le protocole Clusterhead Gateway Switch Routing protocol (CGSR (Chiang *et al.*, 1997)), qui utilise les clusters pour diminuer le volume des inondations, sont des plus représentatifs de cette famille ;

– les protocoles *ad hoc hybrides* qui se comportent comme des protocoles réactifs utilisant, à la demande, des tables de routage pour accroître leur rendement.

3.4. *Les systèmes multiagents appliqués aux réseaux sans fil*

On trouve des composants intelligents dans plusieurs domaines de l'administration des réseaux (Pujolle, 2000) : la configuration, la sécurité, les pannes et la comptabilité. Cette intelligence provient généralement de l'intelligence artificielle distribuée.

Depuis peu, il existe de plus en plus de tentatives de confrontation des deux domaines de recherche que sont les systèmes multiagents et les réseaux de capteurs sans fil. La nature distribuée et ouverte des réseaux de capteurs rend l'approche multiagent particulièrement adaptée. Le fait que cette approche fournisse une représentation externe des interactions et de l'organisation offre plusieurs fonctionnalités comme le contrôle par un observateur externe. Nous avons identifiés plusieurs domaines d'applications des systèmes multiagents dans le contexte des réseaux sans fil : le routage de l'information, la fusion de données, la recherche et la description de services.

La fusion de données. L'article (Petriu *et al.*, 2002) présente un projet dans lequel les agents sont utilisés pour réaliser une structure ouverte et flexible d'assemblage de cellules dans laquelle les données collectées par les agents sont fusionnées pour créer un environnement virtuel. Le projet Unmanned Ground Vehicle Program ARPA (Cook *et al.*, 1996) utilise l'approche multiagent pour fusionner des données provenant de plusieurs véhicules militaires afin d'avoir une vue globale du champ d'opération à partir des différentes vues locales.

La recherche et la description de services. La recherche et la description de services est un domaine de recherche très actif où les systèmes multiagents sont utilisés dans le contexte des réseaux *ad hoc* (Chen *et al.*, 2000; Langley, 2003; Ratsimor *et al.*, 2002; Campo *et al.*, 2002). Dans ce cadre, FIPA, qui travaille sur la normalisation des interactions dans les systèmes multiagents, s'est d'ailleurs intéressé au cas de cette famille de réseaux de communication (Berger *et al.*, 2003). Certains travaux sur ce standard ont été réalisés (FIPA, 2002) et des résultats intéressants ont été mesurés pour certains types de réseaux *ad hoc* (Helin *et al.*, 2002). Un outil pour aider à la conception et au management de réseaux sans fil est présenté dans (Minar *et al.*, 1999). Ce travail concerne la gestion de services en utilisant des agents mobiles selon l'idée d'un écosystème où les agents évoluent dans l'environnement que représente le réseau. Le problème décrit dans (Zhang *et al.*, 2002) est similaire au notre mais l'approche utilise des algorithmes stochastiques distribués. Une infrastructure multiagent réalisant un service de type pages jaunes est proposé dans (Campo *et al.*, 2002).

Le routage de l'information. Peu de travaux, traitant de la problématique du routage de l'information dans les réseaux *ad hoc* d'entités physiques autonomes, utilisent les systèmes multiagents. On peut cependant citer les travaux relatés dans (Choudhury *et al.*, 2000) utilisant des agents mobiles et centrés sur l'optimisation des temps d'acheminement de messages. Leur technique repose sur la gestion des affinités entre agents (similaire aux techniques de routage dites par état des liens). Le projet militaire ActComm (Gray, 2000) pour lequel le routage de l'information est essentiel : l'objectif est de gérer les communications qu'échangent les soldats entre eux et le quartier général *via* un satellite. D'autres approches de l'intelligence artificielle dis-

tribuée connexe aux systèmes multiagents sont utilisées. Dans (Arabshahi *et al.*, 2001) des algorithmes sont décrits pour le routage d'information reposant sur le modèle de coopération de Swarm.

C'est dans ce dernier domaine que se situe notre contribution que nous décrivons dans la prochaine section.

4. Une contribution SMA dans le contexte des réseaux sans fil de capteurs

Nous commençons, dans cette section, par introduire le projet EnvSys qui nous a permis de nous intéresser à la problématique des réseaux de capteurs sans fil. Dans un second temps nous présentons notre approche multiagent appliquée aux réseaux de capteurs sans fil. Cette présentation se fera selon les quatre axes de décomposition AEIO (Demazeau, 1995) que nous exposerons.

4.1. *Le projet ENVSYS*

Le projet ENVironment SYStem [2] (Jamont *et al.*, 2002; Jamont *et al.*, 2004) a pour objectif l'instrumentation de cavités souterraines. Ce système permettra de collecter des données pour étudier des écoulements souterrains, surveiller des captages profonds, anticiper des risques de crues, détecter en amont des réseaux hydrographiques des risques de pollution etc.

Les paramètres intéressants à mesurer dans un réseau hydrographique souterrain sont nombreux : température de l'air et de l'eau, pression de l'air et éventuellement de l'eau pour les réseaux noyés, degré de pollution par certains polluants classiques, débit d'eau, vitesse de courant d'air, etc. Leur mesure est une chose compliquée car l'accès aux réseaux souterrains est difficile (leur structure étant souvent fort chaotique). Il est nécessaire de faire appel à des spéléologues pour la pose des réseaux de communications filaires. Dans le cas de réseaux de communication hertziens, le fait d'être sous terre rend la propagation des ondes difficile et les techniques associées ne sont pas encore totalement maîtrisées.

Depuis quelques années, des systèmes de communication radio ont vu le jour pour les secours spéléologiques. L'idée générale du projet est d'étudier la faisabilité d'un réseau maillé de capteurs à partir de cette couche physique existante. Les informations mesurées seront collectées à la sortie immédiate du réseau *via* une station de travail type PC qui en effectuera le traitement.

Les capteurs devront donc assumer plusieurs fonctions :

– une fonction mesure : c'est le rôle premier d'un capteur. Il consiste à interagir avec l'environnement pour saisir l'information ;

2. Projet soutenu par le Fonds d'Incitation au Transfert de Technologie (FITT) de la région Rhône-Alpes

– une fonction relais : chaque capteur a une portée de transmission limitée. Cette limitation provient à la fois des solutions technologiques qui ont été utilisées pour réaliser le module d'émission du capteur (fréquence, puissance, antenne), de la manière dont ces solutions ont été implantées et de l'environnement du capteur. En effet, l'onde électromagnétique, selon les obstacles qu'elle aura à traverser, ne sera pas exploitable à une distance unique selon chacune de ses directions.

Le rôle des capteurs défini, nous pouvons donc représenter la structure de notre réseau de communication. Il consiste en un ensemble de capteurs et d'une station d'écoute comme l'illustre le graphique 1.

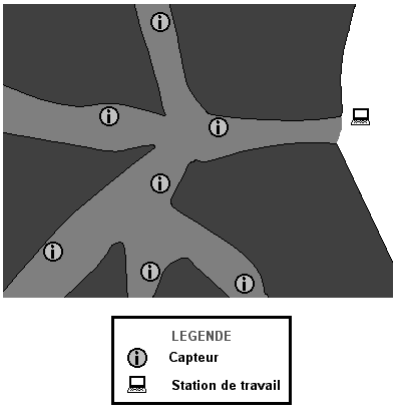


Figure 1. *Le sous-terrain instrumenté*

Voici une liste non exhaustive des problèmes qui nous sont posés :

– comment réaliser la couche physique (modèle OSI) ? Des travaux sont en cours et concernent la caractérisation du médium de communication (Schira, 2003) ;

– quel protocole de niveau liaison choisir au dessus de cette couche physique ? Le choix du protocole de réseau industriel CAN (Controler Area Network) est fortement envisagé mais il nous faudra en montrer la pertinence et sa validité dans le cas de transmission non filaire par des expériences en grandeur réelle ;

– comment router au mieux les messages ? Chacun des capteurs ne peut physiquement pas communiquer avec la station de collecte des informations. Afin de maîtriser la dépense énergétique, lesquels devront prendre la décision de relayer l'information ?

– quelle intelligence donner au réseau pour gérer au mieux cet environnement complexe ?

Les deux premières questions sont relatives à un niveau physique. Par l'application multiagent, nous repondons plus précisément aux deux dernières questions.

4.2. *Notre approche multiagent appliquée aux réseaux de capteurs sans fil*

Il existe plusieurs méthodes de développement de systèmes multiagents (Iglesias *et al.*, 1998) dont la plupart sont centrées sur l'analyse des tâches des agents comme les méthodes Gaia (Wooldridge *et al.*, 2000), MaSE (Multiagent Systems Engineering methodology) (Wood *et al.*, 2000), et ADELFE (Picard *et al.*, 2004), ou sur les rôles ou l'organisation comme la méthode AALADIN (Ferber *et al.*, 1998).
Nous allons nous intéresser dans la suite à la décomposition AEIO (Demazeau, 1995) qui privilégie une description explicite des interactions et de l'environnement, ce qui est plus adapté pour un problème tel que le notre. Il est en effet important de disposer d'une description externe, indépendante des agents mais partagée par ceux-ci, afin de garantir une certaine généricité et une intégration plus aisée. Nous utiliserons la méthode complète d'analyse et de conception multiagent AEIO abordée dans (Occello *et al.*, 2001) et qui est associée à cette décomposition. Cette approche est articulée autour de quatre axes communément admis aujourd'hui et qui sont développés ci-dessous.

4.2.1. *L'axe environnement*

L'axe environnement regroupe les informations permettant de qualifier l'espace commun aux agents du système. On peut identifier certaines de ses caractéristiques (Russel *et al.*, 1995; Wooldridge *et al.*, 2000). Cet environnement peut être *accessible* si un agent peut, à l'aide des primitives de perception, déterminer l'état de l'environnement et ainsi procéder, par exemple, à une action. Si l'environnement est *inaccessible* alors il faut que l'agent soit doté de moyens de mémorisation afin d'enregistrer les modifications qui sont intervenues. L'environnement est déterministe si son état futur n'est fixé que par son état courant et les actions de l'agent. Il est *épisodique* si le prochain état de l'environnement ne dépend pas des actions réalisées par les agents. Un environnement statique suppose que son état est stable (ne change pas) pendant que l'agent réfléchit. Dans le cas contraire, il sera qualifié de *dynamique*. Si le nombre des actions faisables et des états de l'environnement est fini l'environnement sera dit stable. Enfin l'environnement sera *avec ou sans adversaire relationnel* selon que les agents plongés dans cet environnement sont ou non en compétition.

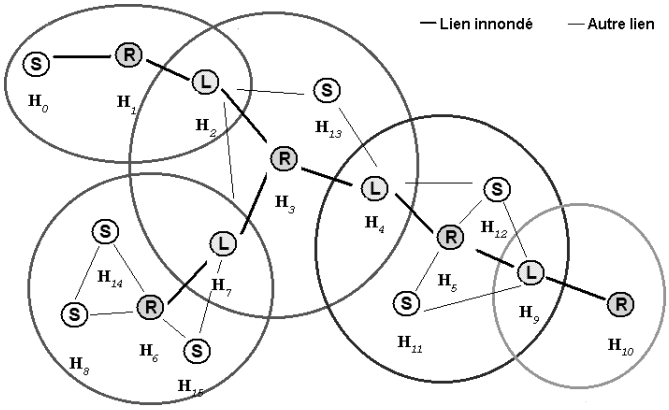
Cette partie de l'analyse consiste donc à trouver les éléments nécessaires à la réalisation des interactions extérieures au système comme par exemple la perception de cet environnement, les actions que l'on peut y faire. Dans notre cas, l'environnement est fait de toutes les informations mesurables par les agents. Nos agents sont situés mais ne peuvent percevoir leur position géographique. Notre environnement, comme la majorité des environnements physiques, est non déterministe, non épisodique, dynamique et continu.

4.2.2. *L'axe organisation*

L'organisation peut être définie comme étant une structure décrivant les relations qui existent entre les membres d'une société (Fox, 1981).

Nous avons décidé d'organiser les agents en groupes ayant une structure hiérarchique semblable à celle que l'on trouve dans les clusters de CGSR (cf. 3.3) afin de localiser au mieux l'inondation et ainsi obtenir un gain conséquent en énergie. La localisation d'un agent consiste donc à repérer le groupe auquel appartient l'agent. La propriété principale de l'organisation est qu'elle est dynamique. Dans ce type d'application nous ne pouvons pas contrôler l'organisation *a priori* : la relation entre les agents émerge des interactions qu'ils ont entre eux et de l'évolution de leurs états.

Nous définissons trois rôles d'agent différents. L'*agent représentant* administre les communications au sein de son groupe. L'*agent de liaison* permet aux différents représentants de communiquer (ils sont aux frontières de plusieurs groupes). L'*agent simple membre* n'a aucun rôle actif dans la gestion des communications



(R) Agent Représentant , (L) Agent de liaison , (S) Agent Simple Membre

Figure 2. *Organisation en groupes des agents capteurs*

La structure de base de notre organisation (voir figure 2) est composée d'un agent représentant, d'un ou plusieurs agents de liaison et d'éventuellement des *agents simple membres*. Les agents fonctionnent tous sur le même schéma mais leur rôle conditionnera des comportements spécifiques face aux messages qui proviennent des autres agents. En effet, tous les agents ont un même protocole d'introduction pour leur entrée dans le système. Ils participent tous à la recherche d'incohérence lors de l'envoi de message et désirent faire parvenir leurs mesures à une station de collecte des informations (nous reviendrons sur ces deux notions dans la suite de l'exposé). Cependant, un agent représentant devra en plus assumer toutes les requêtes d'envoi de message des membres de son groupe : quand le destinataire d'un message n'est pas voisin de l'émetteur, c'est à lui que revient le rôle de trouver une route, de vérifier le bon acheminement du message. C'est aussi lui qui est inondé par les requêtes de recherche de routes des autres représentants. Il est donc plus sollicité que les autres agents de son groupe (hormis éventuellement les agents de liaison). Un agent de liaison est chargé d'assurer la communication entre les groupes c'est-à-dire ses voisins qui sont représentants. Ce sont donc eux qui distribuent les requêtes de recherche de routes aux

représentants. Les agents simple membres n'ont par contre aucun rôle particulier dans la gestion des messages.

Le fonctionnement de notre solution sera cependant très différente de CGSR. En effet, nous intégrons une réelle *politique de gestion de l'énergie* et nous procédons à la *suppression des envois périodiques* de messages qui lui permet de maintenir la cohérence de l'organisation. Le degré d'adaptation de ce protocole est donc fonction de cette fréquence d'envois de messages.

Une politique de gestion d'énergie. Un agent représentant étant très sollicité, d'un point de vue communication, il dépensera en moyenne plus d'énergie que les autres : il doit donc disposer d'une importante réserve énergétique. Plus un groupe est important, plus il présente un intérêt certain dans la réduction du volume transmis induit par l'inondation : le système préserve donc d'autant plus d'énergie. L'attribution de rôle doit intégrer ces informations. Elle se fait d'une manière très simple et avec la seule connaissance de ces voisins. La technique d'auto-organisation que nous utilisons est basée sur le mécanisme d'auto-organisation par attribution de rôle. Nous avons privilégié ce mécanisme (cf. 2.2.2) car il est à la fois simple et particulièrement adapté à la structure de groupe que nous souhaitons mettre en place. L'adéquation entre le désir qu'a un agent de devenir représentant et sa capacité à l'être est estimée grâce à une fonction $score$. Cette fonction doit mesurer le compromis entre les deux critères précédents et départager les agents lors de l'élection du représentant. La fonction $score$ que nous avons implémentée est très simple : elle multiplie le nombre de voisins par le niveau d'énergie de l'agent.

```
SI monNombreDeVoisin>0 ALORS
    // On a des voisins
    SI nbRepresentantsVoisin=0 ALORS
        // Aucun de nos voisins n'est représentant : on décide de le devenir. Ce cas intervient quand on vient
        // juste de créer l'agent ou quand il vient de se déplacer. On ne procède pas à un vote car on pourrait
        // rendre le système instable (le capteur va sûrement continuer sa route).
        monRole = REPRESENTANT ;
    SINON SI nbRepresentantsVoisin==1 ET monRole !=REPRESENTANT ALORS
        // Un de nos voisin est représentant : on se soumet à son autorité et ce même si il est moins efficace
        // que nous : on privilégie (pour le moment) la stabilité dans l'organisation à sa performance. On
        // attendra une défaillance ou son souhait de quitter son mandat pour prendre sa place
        monRole = SIMPLEMEMBRE ;
    SINON
        //Il existe, dans notre voisinage, plusieurs représentants.
        SI monRole = REPRESENTANT ALORS
            // Je suis moi même représentant : j'entre en conflit avec les autres prétendants à ce rôle. Une
            // élection va avoir lieu et l'agent au meilleur score restera en place
            ProcedureElectionRepresentant()
        SINON
            // On n'est pas représentant : on devient agent de liaison pour ses représentants
            monRole = LIAISON
        FINSI
    FINSI
SINON
    // on n'a pas de voisin : on n'a plus aucun rôle
    monRole = AUCUN
FINSI
```

Tableau 1. *Algorithme de sélection du rôle*

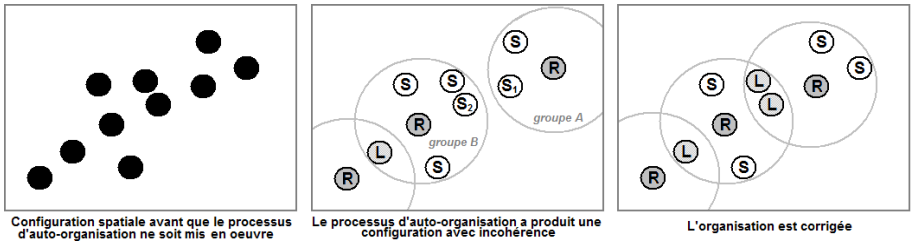
Comme le montre notre algorithme (voir tableau 1), lorsqu'un conflit intervient, une procédure d'élection du meilleur représentant est engagée. L'agent qui a détecté le conflit va calculer son $score$ et le transmettre au représentant avec lequel il est en conflit. Ce dernier, quand il recevra ce message, prendra conscience (si ce n'était pas déjà le cas) du problème et le comparera à son propre score. Si il est moins élevé, il démissionnera de son mandat, changera de rôle et le signalera à ses voisins. Cet algorithme est en réalité modulé par la correction d'incohérences (voir figure 3) que nous abordons par la suite.

Suppression des envois périodiques de messages. Nous adoptons le principe selon lequel l'objectif n'est pas d'avoir une organisation cohérente à tout moment : la perte énergétique associée ne serait pas forcément rentable. Cette organisation, si elle n'est pas cohérente, ne sera remise en question que lorsqu'elle posera problème dans le cadre d'un échange de communications. Pour cela, les agents font de l'*écoute indiscrète*. Ce mécanisme, que d'aucun appelle écoute flottante (Legras, 2002) consiste à espionner les messages qui passent dans son aire d'écoute afin d'en tirer des informations. Un capteur peut ainsi vérifier :

– la cohérence de sa table des voisins ou de ses éventuelles tables de routage (ex : quand un voisin émet un message, l'agent vérifie que le triplet *<id,role,groupe>* de ce voisin émetteur de la trame qu'il entend soit bien dans la table de voisins) ;

– la cohérence structurelle dans le voisinage comme l'illustre la figure 3. Dans ce cas l'agent *<S1,SIMPLE MEMBRE,A>* qui reçoit le triplet *<S2,SIMPLE MEMBRE,B>* le met dans sa table de voisinage. Il signale ensuite à son représentant qu'il peut éventuellement y avoir une incohérence (message *VerificationCoherence(<B>)*). Son représentant regarde dans sa table des groupes voisins si ce groupe B est présent. Si il n'est pas joignable, selon qu'une route à faible coût existe ou non, soit le représentant avec le plus mauvais $score$ démissionne, soit les représentants décident de l'utiliser (afin d'éviter le coût d'une réorganisation). Si cette route est utilisée fréquemment, cette décision peut être remise en question par un des représentants.

L'acquittement des messages intervient aussi dans ce processus de maintien de cohérence. Quand un message n'est pas acquitté, les représentants concernés suppriment la route de leurs tables et une nouvelle recherche de route est amorcée.



(R) Agent Représentant , (L) Agent de liaison , (S) Agent Simple Membre

Figure 3. *Correction d'incohérence résultant du processus d'auto-organisation*

4.2.3. *L'axe interaction*

Les interactions proviennent de la mise en relation dynamique de plusieurs agents par le biais d'un ensemble d'actions réciproques. Il existe plusieurs types d'interactions, qui dépendent de trois paramètres que sont les buts, les ressources et les compétences. Cet axe de l'analyse AEIO comprend donc les éléments qui servent à la structuration des interactions externes entre les agents (langage d'interaction, protocoles de communication...).

Dans notre système, les agents interagissent seulement avec les agents en accointance avec eux. Un agent est en accointance avec un autre si et seulement si il est conscient de son existence c'est-à-dire présent dans son aire d'émission/réception. Ils interagissent *via* l'envoi de message. Nous pouvons distinguer deux méthodes pour réaliser ce type d'interaction : une méthode synchrone, où les agents qui communiquent doivent être en rendez-vous, et une méthode asynchrone où les messages sont mémorisés. Il n'est pas nécessaire que le récepteur et l'émetteur soient donc synchronisés. Du point de vue de la communication, la solution asynchrone nous semble être la meilleure car la plus flexible.

Nom du message	Description
QuiSontMesVoisins	Message qui permet à un agent de se faire connaître et de demander à ses voisins de faire de même. Ce message est émis quand un agent est créé (le premier but d'un nouvel agent est de connaître ses voisins). Ensuite, ce message ne sera utilisé que lorsque l'agent aura l'impression (justifiée ou non) que sa table des voisins ne coïncide plus avec la réalité.
JeSuisUnDeTesVoisins	Message qui permet à un agent de répondre à la précédente requête. Avec ce message, il fournit donc son identifiant, son rôle et son groupe d'appartenance.
JeChangeDeRole	Message utilisé par tout agent changeant de rôle afin d'en avertir ses voisins. Il contient l'identifiant de l'agent, son rôle et son groupe d'appartenance.
DemandeGrpAgentLiaison	Message utilisé par les représentants qui veulent remettre à jour leur connaissance sur les groupes voisins. Elle oblige les agents de liaison à répondre.
ReponseGrpAgentLiaison	Message utilisé par un agent de liaison pour signaler à un représentant la liste des groupes que cet agent peut contacter.
VerifieCoherence	Message envoyé par un agent qui croit avoir détecté une incohérence avec un groupe voisin. Il y a incohérence entre deux groupes quand deux simple membres de groupes différents se voient et que leurs représentants ne peuvent pas communiquer à faible coût.
ResolutionConflit	Message utilisé par un représentant en conflit pour communiquer son score.
PasseRepresentant	Message qui est un ordre donné par un représentant à un des agents de son groupe. Il peut donner cet ordre s'il s'aperçoit par exemple qu'un groupe voisin est isolé et que la seule manière de briser cet isolement est qu'un agent particulier devienne représentant (les tables de voisins de l'agent et des heuristiques permettent à un agent de faire cette supposition).
ChercheRoute	Message utilisé par un représentant qui veut localiser un autre capteur.
ResultatChercheRoute	Message qui est la réponse du représentant qui à le destinataire à sa charge.
ACKMessage	Message optionnel qui confirme à l'émetteur que son message est parvenu à destination. Il est à noter que cet ACK est logiciel. C'est le protocole de la couche liaison du modèle OSI qui assure l'ACK des trames en point à point.
DonneesEncapsulees	Message qui encapsule des données comme une mesure.
RouteErronee	Message envoyé par un représentant qui s'est aperçu qu'une route est erronée.

Tableau 2. *Types des messages de communication*

Dans notre application nous utilisons, entre autres, un protocole d'introduction des agents dans l'organisation. Il permet à un agent de se faire connaître par ses voisins et d'entrer dans l'organisation. Il s'appuie sur certains des types de messages que nous avons définis (voir tableau 2), tous d'une petite taille (la contrainte sur l'énergie étant à prendre en considération). Ainsi, s'introduire correspond à envoyer le message *QuiSontMesVoisins* qui nous fait connaître de nos voisins et leur demande de se présenter. Ces voisins répondent avec le message *JeSuisUnDeTesVoisins* précisant ainsi le triplet <*id,role,groupe*> qui leur est associé. Le capteur détermine, en fonction de ce voisinage, le rôle qu'il doit prendre. Il le signalera à ses voisins *via* le message *JeChangeDeRole*.

4.2.4. *L'axe agent*

L'axe agent regroupe les éléments nécessaires à la construction des agents c'est-à-dire leur modèle, leur architecture et leur implémentation. Dans notre système, les capteurs sont représentés par des agents. Nous avons adapté l'architecture hybride ASTRO (Occello *et al.*, 1998) que nous baptisons e-ASTRO (embedded ASTRO) pour les réaliser (voir figure 4). Ces agents adoptent un comportement cognitif ou/et réactif. Il est de type cognitif quand une altération de la configuration est survenue. Il est en effet nécessaire d'envoyer des messages pour partager ou mettre à jour ses connaissances afin de collaborer efficacement avec les autres agents. Le comportement est réactif, par exemple, lorsque les agents relaient des messages.

L'agent ASTRO est basé sur le paradigme perception/évaluation/raisonnement/action. Il contient des connaissances (sur l'environnement, sur les autres agents et sur son propre état) qui définissent son modèle du monde. Une supervision continue permet de s'adapter aux changements de l'environnement, un module communication permet de prendre en compte les interactions avec les autres agents. Ces interactions sont traitées par le module de contrôle qui prévient l'agent s'il doit agir ou non. Ces mécanismes entraînent une forte adaptation de l'agent sur les changements de l'environnement.

Nos agents sont munis de deux fonctions : la gestion de la communication et la réalisation des tâches applicatives. La tâche de communication est celle de transmettre, si nécessaire, l'information à d'autres agents ou tout simplement de relayer des messages pour le compte d'autres agents. Tous les agents ont des capacités de communication mais leur comportement face à un message dépend de leur rôle dans l'organisation (cf. 4.2.2). Les tâches spécifiques consistent en la mesure de différents paramètres de l'environnement, enregistrement de données etc. Le rôle de l'agent influe sur ses données : un agent représentant possède des connaissances sur tous les groupes qui sont adjacents (mémorisés dans une table) contrairement aux agents de liaison qui eux n'ont conscience que des groupes auxquels ils appartiennent et éventuellement, comme les agents simple membres, à une partie des groupes adjacents (les groupes directement adjacents à l'agent et « non » au groupe). L'arbre de la figure 4 récapitule l'ensemble des données et des tâches de l'agent.

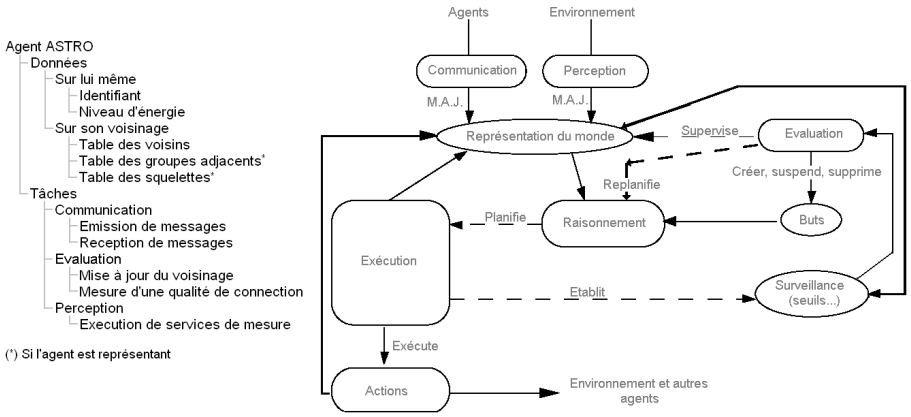


Figure 4. *Arbre des données/tâches d'un agent et description de son architecture*

4.2.5. *Implémentation des agents*

Nous avons implémenté un système minimal où une station maître collecte les informations émises par les capteurs afin de démontrer la faisabilité de notre approche. Pour les capteurs nous avons choisi une architecture classique en trois couches :

1) la *couche physique* utilisée est celle du système NICOLA (Graham, 1999). Cette couche est implémentée grâce à un Digital Signal Processor (DSP) ;

2) la *couche liaison* utilise le protocole CAN (Controller Area Network) qui a été éprouvé depuis longtemps dans l'industrie automobile ;

3) la *couche applicative* est constituée par le système multiagent. Les agents sont embarqués sur des cartes à processeurs. Elles sont équipées d'un module de communication et d'un module de mesure qui permet à l'agent d'instrumenter son environnement (prise de température uniquement pour le moment). Ces cartes fournissent un noyau temps réel. Le noyau temps réel KR-51 permet de créer une application multitâches utilisant le micro-contrôleur C515C. Nous pouvons ainsi associer à une tâche une compétence et ainsi implémenter facilement le parallélisme inhérent aux agents pour satisfaire les contraintes temps réel.

5. Evaluation de notre approche

Nous allons nous intéresser dans cette partie à l'évaluation de notre approche. Nous commençons par décrire notre plate-forme de simulation. Nous exposons et discutons ensuite les résultats obtenus avec notre processus d'auto-organisation.

5.1. *La phase de simulation*

5.1.1. *Raisons d'être de cette plate-forme*

La plate-forme de simulation va nous permettre d'expérimenter notre approche et les solutions logicielles que nous avons choisies pour implémenter les agents. Le passage par la simulation est indispensable car elle permet de *mettre à jour d'éventuels problèmes* qui seront plus faciles (en termes de programmation) à traiter sur le simulateur que sur les cartes à micro-contrôleur. Il permet aussi de *mesurer les performances* (rendement) de notre système et ainsi de les comparer *via* les mesures de rendement à d'autre solutions qui ont déjà été implémentées. Il sera aussi possible de *quantifier l'émergence* inférée par l' approche multiagent par l'intermédiaire d'autres graphiques mis à disposition par la plate-forme mettant en évidence l'évolution en fonction du temps du volume utile et de configuration, la variation de volume transmis entre deux mesures ou le nombre de message émis.

5.1.2. *Structure logicielle et fonctionnalités*

La structure logicielle est très simple : nous avons deux types de composants ; le simulateur de réseau *SimNetwork* et le simulateur de capteur *SimSensor*. Un composant SimNetwork simule le comportement d'un capteur et possède donc son propre modèle et sa propre architecture. Tous les simulateurs de capteurs sont dotés des mêmes fonctionalités de communication. Ils envoient leurs requêtes au composant SimNetwork qui y répondra suivant l'environnement simulé.

Le simulateur a été réalisé en Java et conçu de manière à ce qu'une personne qui désire ajouter une nouvelle architecture d'agent n'ait pas à connaître le fonctionnement du simulateur de réseau. Il suffit d'étendre une classe appelée Capteur, qui propose les primitives minimales d'envoi de trames et d'affichage d'informations, et de comprendre le mécanisme de notification d'événements. Ces événements permettent le rafraîchissement des informations affichées sur le capteur et de mesurer les performances de l'implantation de l'architecture.

La plate-forme de simulation fournit en premier lieu les fonctionnalités concernant la création de l'environnement dans lequel évoluent les capteurs. Ils s'agit de choisir une carte et de pouvoir fixer la position géographique initiale des capteurs. Les capteurs doivent pouvoir être espionnés c'est-à-dire qu'il faut avoir accès à leurs attributs (niveau d'énergie, rôle, portée), à leurs contenus mémoire (tables de routage, tables de voisins etc.) et obtenir un historique des messages émis et reçus ainsi que des actions menées. Il faut aussi pouvoir observer les informations sur le système global : performances du routage et critères permettant d'estimer la mémoire occupée. Il faut pouvoir faire subir au système des perturbations comme des fautes effaçant les données d'un agent sur ses voisins ou le détruisant afin d'étudier la tolérance aux fautes. Des déplacements doivent pouvoir être introduit afin d'étudier l'impact de la mobilité sur les performances. Enfin, il est important que la plate-forme permette la création de scénarios afin que chaque protocole soit éprouvé avec les mêmes événements aux

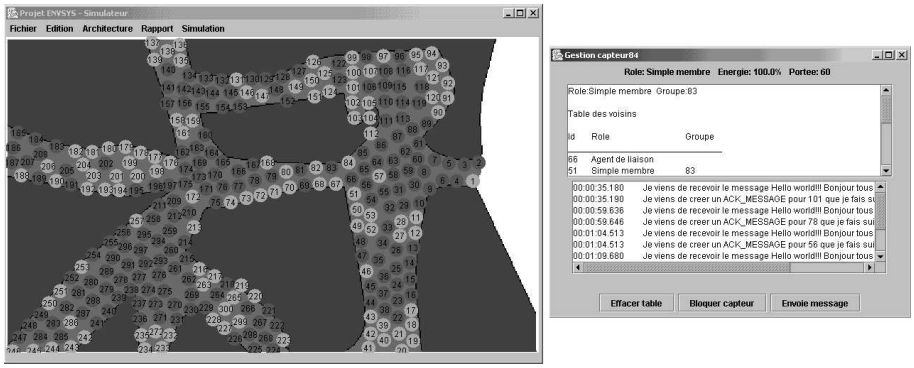


Figure 5. *Captures d'écran du simulateur (grotte instrumentée/espion d'un capteur)*

mêmes instants. La figure 5 montre une capture d'écran du simulateur (on aperçoit la grotte et les capteurs, et un espion de capteur).

5.2. *Résultats de simulation*

Nous avons comparé notre système multiagent avec trois protocoles *ad hoc* très populaires dans la communauté des réseaux *ad hoc*. Ces protocoles sont DSDV et deux variantes de DSR. La première de ces variantes est la version originale de ce protocole (nous l'appellerons DSR) et la seconde une amélioration avec maintenance des routes (notée DSR-ROUTAGE). Cependant, les résultats obtenus par DSDV étant des plus mauvais, ils ne seront pas présentés dans la suite de l'article. Nous n'avons pas comparé CGSR à notre approche, les spécifications techniques de ce protocole restant introuvables.

Nous appellerons dans la suite *rendement du protocole* à instant t le rapport entre le volume utile théorique des différents messages envoyés (en supposant qu'ils aient pris les routes optimales) que divise le volume total transmis par le protocole. Le système qui aura le plus faible rendement aura donc transmis moins de données et donc économisé plus d'énergie. Cette définition du rendement permet de pénaliser correctement les protocoles qui n'utilisent pas les routes optimales. Il est à noter que les protocoles DSR et DSR-ROUTAGE utilisent des routes généralement optimales. Le routage effectué par l'auto-organisation, obtenue avec nos agents capteurs, utilise quand à lui souvent des routes non optimales : en effet, dans notre approche c'est la succession de groupes qui est optimale et non la succession de capteurs.

Illustration du calcul de rendement proposé

Définition : On appelle *volume utile* d'un message le volume qu'occupe les données transportées et *volume de configuration* d'un message le volume des informations

ajoutées pour l'acheminement du message (identifiant destinataire, identifiant de type de message).

Notations : Soit v_u le volume utile d'une trame, V_u le volume utile des messages échangés pour l'acheminement du message de l'émetteur au destinataire. De la même manière nous utiliserons les notations v_c et V_c pour exprimer le volume de configuration ainsi que v_t et V_t pour exprimer le volume total. On a v_t = v_u + v_c et V_t = V_u + V_c + V_o. Soit r le rendement. On a défini r = V_u(RouteOptimale) / V_t(Protocole).

Cas pratique : Le capteur s_1 veut transmettre une température à s_4. La température est codée sur 4 octets mais la trame totale envoyée prends 10 octets. On a v_u = 4 octets et v_c = 6 octets soit v_u = 10 octets.

Le protocole DSR : La route proposée est la route optimale (c'est quasiment toujours le cas) qui correspond à la succession de capteurs (s_1,s_2,s_3,s_4). DSR a perdu 50 octets pour trouver cette route. Son rendement après l'envoi d'un message sera r_{dsr} = (4 * 4) / (50 + 4*(4+6)) = 17.7%. Si on envoie deux messages on aura pour le protocole DSR un r_{dsr} = (2*(4 * 4)) / [2*50 + 2*(4*(4+6))] = 17.7% tandis que pour le protocole DSR_ROUTAGE nous obtiendrons $r_{dsrRoutage}$ = (2*(4 * 4)) / [50 + 2*(4*(4+6))] = 22.8%.

Notre approche : La succession de capteurs proposée est (s_1,s_5,s_6,s_3,s_4). Admettons que l'organisation ait nécessité 40 octets pour se mettre en place et 20 octets pour trouver le destinataire (le volume de la requête de recherche prend moins de place). Après l'émission du message on aura transmis V_t = (4*4) / [40 + 20 + (5*(4+6)] soit 14.5%. Après deux messages on aura V_t = 2*(4 * 4) / [40 + 20 + 2*(5*(4+6)] = 20%.

Il est important de prendre en compte qu'avec notre approche tous les messages sont acquittés contrairement au protocole DSR. L'acquittement des messages intervient dans le maintien de la cohérence de l'auto-organisation au même titre que l'écoute indiscrète.

5.2.1. *1er cas d'utilisation : le projet ENVSYS*

Pour ce cas d'étude nous nous placerons dans le contexte du projet EnvSys c'est-à-dire celui d'un protocole unidirectionnel. En effet, les capteurs veulent effectuer des mesures et les faire parvenir à la station de travail qui collecte les informations. Pour cette expérience toutes les cinq secondes deux capteurs transmettent leurs données. Ces messages contiennent un octet qui décrit le type de mesure et quatre octets qui codent l'information mesurée. Les mêmes scénarios sont appliqués aux protocoles. Ces mesures ont été effectuées pour 50, 100, 150, 200, 250 et 300 capteurs. Les rendements mesurés pour ces différentes configurations sont représentés en figure 6.

Nous pouvons voir que DSR a un bien moins bon rendement que DSR-ROUTAGE. Dans ce cas d'utilisation DSR-ROUTAGE se comporte bien car ou bout d'un certain temps, aux perturbations près, toutes les routes auront été mémorisées vu que nous sommes dans un cas unidirectionnel. L'avantage de l'approche multiagent dans ce cas d'utilisation est qu'elle permet rapidement d'atteindre un bon rendement. On peut à

juste titre supposer que dans un cas d'utilisation réelle (fréquence d'envoi des mesures beaucoup plus faible) les capteurs auraient épuisé leur énergie avant les capteurs ASTRO. Le rendement de DSR-ROUTAGE n'aurait d'ailleurs ainsi jamais été supérieur à celui de l'approche multiagent que nous avons menée. Quand la simulation commence les agents s'auto-organisent. Pour auto-organiser (50, 100, 150, 200, 250, 300) capteurs il aura fallu échanger entre capteurs un total cumulé de (3 800, 6 950, 13 900, 16 220, 23 870, 24 790) octets. Ce volume n'est donné qu'à titre indicatif car d'une auto-organisation à l'autre les volumes sont rarement les mêmes. On constate aussi que le rendement de notre solution varie peu en fonction du nombre de capteurs. La raison est que notre environnement provoque essentiellement une augmentation de la densité de capteurs par groupe et non du nombre de groupes lui même.

5.2.2. *Second cas d'utilisation : protocoles multidirectionnels*

Pour ce cas d'utilisation nous ne sommes plus dans le cas des protocoles unidirectionnels. L'expérience dure 100 minutes. Toutes les secondes un capteur choisi au hasard transmet des messages (en moyenne 30 octets) à un autre capteur choisi au hasard. La figure 7 montre les résultats obtenus par DSR et ASTRO. Les protocoles DSR-ROUTAGE et DSR ayant quasiment les mêmes rendements nous n'avons représenté qu'un seul graphe pour ces deux différents protocoles.

Dans ce cas de figure nous constatons que notre approche multiagent a un bien meilleur rendement et ce malgré le biais de volume de configuration utilisé pour auto-organiser les agents. Cela est du au fait que, contrairement à DSR, l'inondation avec notre approche est très localisée. Dans le cas de DSR ce sont tous les capteurs qui sont

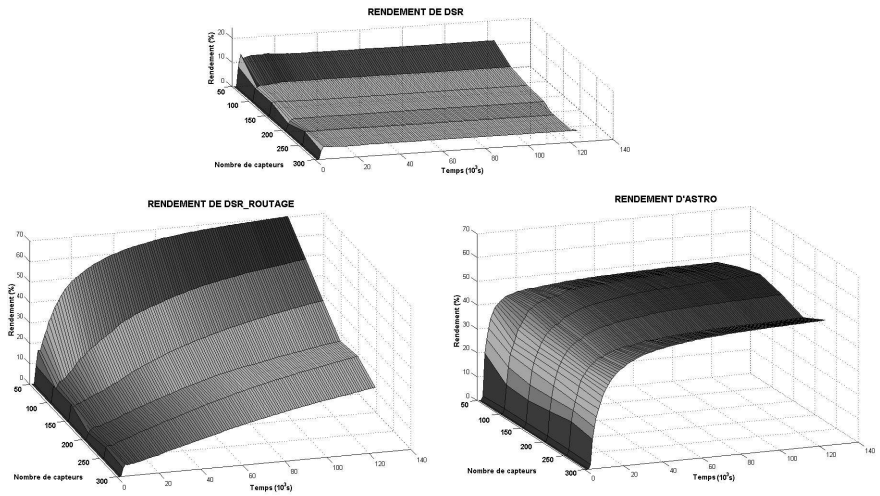


Figure 6. *Performances de DSR, DSR-ROUTAGE et de notre approche multiagent*

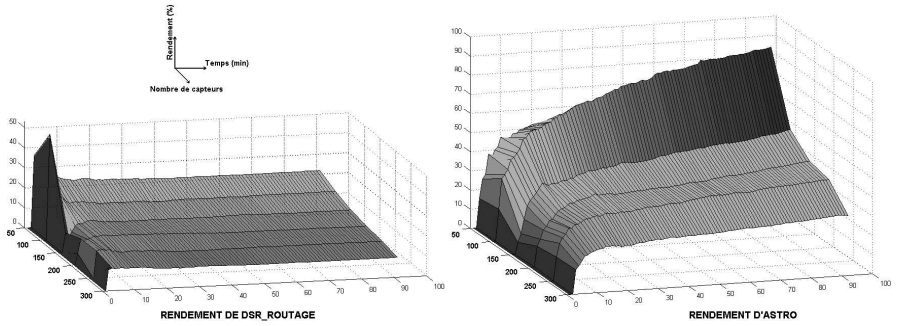


Figure 7. *Performances de DSR et de notre approche multiagent*

sollicités pour répéter les messages alors qu'avec notre solution ce sont au plus, tous les agents représentants et certains des agents de liaison.

5.2.3. *Discussion à propos de ces résultats*

Cette discussion a pour but de présenter une synthèse de nos comparaisons ainsi qu'une analyse du processus auto-organisationnel à l'origine de l'intérêt de notre approche.

Synthèse de nos comparaisons. Nous avons vu que notre approche était particulièrement adaptée à notre problème. Nous avons dressé un comparatif (tableau 5.2.3) de notre protocole avec DSR, DSR-Routage et DSDV dans le contexte d'EnvSys en modulant les notations attribuées par des remarques critiques de notre propre notation.

Critére	DSDV	DSR	DSR-ROUTAGE	ASTRO
Optimalité des routes (1)	* * * * *	* * * * *	* * * * *	* * *
Tolérance aux pannes (2)	*	* * * * *	* * * * *	* * * *
Rendement du protocole (3)	Lié à la fréquence de rafraîchissement des tables. Mauvais rendement dans tous les cas.	Moyen	Moyen, bon dans le cas de protocoles unidirectionnels	Bon en général, très bon dans le cas de grand réseaux en grappes.
Adéquation pour EnvSys (4)	*	* *	* * *	* * * *

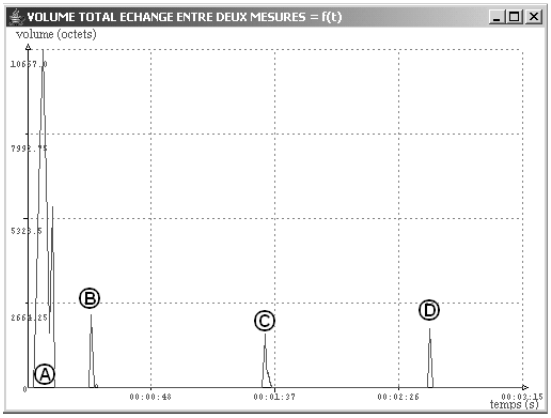
(1) L'optimalité de notre solution est en termes de groupes : elle ne propose pas la meilleure succession de stations mais la meilleure succession de groupes.
(2) La tolérance aux pannes concerne la tolérance à la défaillance d'un lien suite à la mobilité/destruction d'un capteur. Aucune solution ne peut être meilleure que DSR car il est purement réactif : il n'utilise aucune table de mémorisation.
(3) Il est difficile d'apprécier le rendement car il est fonction de divers paramètres tel que le volume utile des messages transportés (vu le caractère non optimal des routes proposées par notre approche, l'envoi de très grands messages pourrait être pénalisant), la fréquence d'émission des messages, du nombre moyen de stations dans l'aire d'émission/réception etc.
(4) Ce critère consiste à mesurer l'adéquation entre les attentes du projet EnvSys (gestion des dépenses énergétiques) : il consiste en un classement de ces 4 solutions.

Tableau 3. *Tableau comparatif des performances des différentes solutions testées*

Nous n'avons pas intégré à notre plate-forme des solutions résultant d'une approche multiagent pour le routage des informations (cf. 3.4). En effet les contextes des expérimentations que nous avons évoquées sont trop différents pour permettre une comparaison qualitative. Les solutions basées sur les agents mobiles tel que (Choudhury *et al.*, 2000) sont trop lourdes pour des capteurs dont les ressources matérielles sont très limitées. La solution proposée dans (Arabshahi *et al.*, 2001) qui n'a été expérimentée que pour des petits réseaux de 30 stations maximum a été conçue pour optimiser le temps d'obtention d'une route et non le critère énergétique.

Evaluation de l'auto-organisation. En termes de résultats, les figures 6 et 7 montrent que notre approche est meilleure dans les cas d'utilisation de EnvSys. La structure de groupes des agents a donc bien permis une réduction du volume des messages qui permettent la localisation des autres agents et donc l'établissement de routes. La *validité* de notre approche peut donc être établie.

Les raisonnements mis en jeu par les agents qui participent au processus d'auto-organisation ne nécessitent pas de primitives très cognitives. Le résultat (les groupes) de ce processus étant facilement identifiable et compréhensible par un observateur extérieur, la *simplicité* du processus peut être constatée.



(A) Auto-organisation de 300 capteurs, (B) correction d'une incohérence, (C) ajout de 10 capteurs, (D) correction survenue à la suite d'une perturbation (déplacement de capteurs)

Figure 8. *Variation du volume échangé par le processus d'auto-organisation*

En termes d'auto-organisation, la figure 8 permet de constater la *stabilité*. Les structures qui sont établies font preuve d'une constance dans le temps. Elles ne sont remises en cause que lorsqu'une incohérence (événement B) est détectée. La figure permet de voir les envois de messages qui ont permis ces modifications. Le déplacement de capteurs (évènement D) peut entraîner des ré-élections si l'agent qui se déplace est un représentant ou un agent de liaison qui est le seul à permettre la communication entre deux groupes connexes. Dans les autres cas, cette mobilité n'aurait

entraîné aucune autre ré-élection de représentant : elle change seulement la structure interne de certains groupes. Il est à noter que les messages échangés qui concernent l'ajout des 10 capteurs (événement C) n'ont pas affecté la topologie existante. Les agents représentants existants n'ont pas perdu leur mandat afin de privilégier justement cette stabilité du processus. Par contre des groupes ont vu leur nombre de simple membres ou d'agents de liaison augmenter et d'autres groupes sont apparus. La *sensibilité* du processus, que l'on ne peut complètement dissocier du critère de stabilité, a été abordée dans le point précédent. Les états de reconnaissance totale concernent les structures cohérentes. D'éventuelles structures incohérentes peuvent avoir été détectées précédemment (état de reconnaissance partielle) et peuvent être corrigées. Il est à noter que si le processus mis en place avait été trop sensible, il y aurait eu un impact visible sur les rendements des protocoles. *A contrario*, si il avait été trop peu sensible, la tolérance aux pannes aurait été mauvaise. La *convergence* ne peut être que constatée sur les simulations.

6. Conclusions et perspectives

Le système multiagent que nous avons créé a deux importantes caractéristiques. La première est liée au caractère ouvert du système : l'ajout ou la suppression d'éléments ne nécessite pas de reconfiguration manuelle. La tolérance aux pannes y est assurée grâce aux capacités adaptatives de l'auto-organisation. Le dysfonctionnement d'un capteur ne nuit pas à l'intégrité de notre système. La deuxième caractéristique est son homogénéité : l'architecture que nous avons choisie est l'architecture ASTRO. La phase de simulation nous a permis de comparer l'approche multiagent aux approches classiques.

Nous pouvons résumer nos contributions en différents points. La fonctionnalité émergente inférée par l'approche multiagent a rendu le système robuste et flexible. Il semble être bien adaptatif et tolérant aux fautes provoquées par les capteurs qui subissent cet environnement agressif que constituent les grottes inondées.

En termes d'ingénierie, notre solution a été adaptée en un intergiciel par envoi de messages décentralisé. Elle a été encapsulée dans un composant générique et rendu ainsi réutilisable. Il sera manipulé par l'outil associé à notre méthode pour la conception des systèmes multiagents physiques. Il permet ainsi d'envisager différentes applications nécessitant ce type de réseau de communication telles que le diagnostic, la prévention de risque, la fusion de données, l'établissement de modèles prédictifs pour les géophysiciens, le suivi de mouvements, l'informatique médicale.

7. Bibliographie

Agha K. A., Pujolle G., Vivier G., *Réseaux de mobiles et réseaux sans fil*, Eyrolles, 2002.

Akyildiz I. F., Kasimoglu I. H., « Wireless sensor and actor networks : research challenges », *Ad Hoc Networks*, vol. 2, p. 351-367, October, 2004.

Arabshahi P., Gray A., Kassabalidis I., Das A., Marks II R. J., « Adaptive Routing in Wireless Communication Networks using Swarm Intelligence », *Proceedings of the 9th AIAA Int. Communications Satellite Systems Conference*, p. 1-9, April, 2001.

Berger M., Watzke M., Helin H., « Towards a FIPA Approach for Mobile Ad hoc Environments », *Proceeding of the 8th International Conference on Intelligence in next generation Networks*, Bordeaux, France, May, 2003.

Calderoni S., Courdier R., Leman S., Marcenac P., « Construction expérimentale d'un modèle multi-agents », *in* J. Quinqueton, M.-C. Thomas, B. Trousse (eds), *5èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Hermès Science, Nice, p. 109-123, Avril, 1997.

Campo C., Marin A., Garcya-Rubion C., Breuer P. T., « Service Discovery in Pervasive MultiAgent Systems », *AAMAS2002 - Proceedings of the Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices*, ACM, July, 2002.

Camps V., *Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : application à la recherche d'information dans un système d'information répartie*, Thèse de Doctorat Informatique, Université Paul Sabatier, décembre, 1995.

Chen H., Chakraborty D., Xu L., Joshi A., Finin T., « Service discovery in the future electronic market », *Proceedings of the 17th National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Austin, Texas, USA, p. 1-6, july, 2000.

Chiang C., Wu H. K., Liu W., Gerla M., « Routing in clustered multihop, mobile wireless networks », *Proceedings of IEEE Singapore International Conference on Networks - SICON'97*, Singapore, p. 197-211, september, 1997.

Choudhury R. R., Paul K., Bandyopadhyay S., « Topology discovery in ad-hoc wireless networks using mobile agents », *Proceedings of the 2nd ACM International Workshop on Mobile Agents for Telecommunication Applications*, Springer-Verlag, p. 1-16, September, 2000.

Clausen T., Jacquet P., Laouiti A., Muhlethaler P., Qayyum A., Viennot L., « Optimized Link State Routing Protocol », *IEEE International multitopic conference*, Pakistan, p. 62-68, April, 2001.

Cook D. J., Gmytrasiewcz P., Holder L. B., « Decision-Theoric Cooperative Sensor Planning », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18(10), p. 1013-1023, October, 1996.

Demazeau Y., « La plate-forme PACO et ses applications », *Actes des 2émes Journées du PRC-IA sur les Systèmes Multi-Agents*, Montpellier, France, Decembre, 1993.

Demazeau Y., « From Interactions to Collective Behavior in Agent-Based Systems », *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo France, p. 117-132, 1995.

Drogoul A., *De la Simulation Multi-agents à la Résolution Collective de Problèmes*, Thèse de Doctorat Informatique, Université Paris VI, November, 1993.

Ferber J., Gutknecht O., « A meta-model for the analysis and design of organizations in Multi-Agent Systems », *in* Y. Demazeau (ed.), *Proceedings of the International Conference on Multi-Agent Systems*, IEEE Computer Society, Paris France, p. 128-135, July, 1998.

FIPA, *FIPA ACL Message Representation in Bit-Efficient Specification, Specification number SC00069*, Dec, 2002. http ://www.fipa.org.

Forrest S., *Emergent computation*, The MIT Press, Cambridge, Massachussetts, USA, 1991.

Fox M. S., « An organizational view of distributed systems », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, p. 70-80, January, 1981.

Gasser L., « An Overview of DAI », *in* N. Avouris, L. Gasser (eds), *Distributed Artificial Intelligence : Theory and Praxis*, Kluwer Academic Publishers, Boston, 1992.

Graham N., « The Nicola Mark II : a New Rescue Radio for France », *The CREG Journal*, vol. 38, p. 3-6, December, 1999.

Gray R. S., « Soldiers, Agents and Wireless Networks : A Report on a Military Application », *Proceedings of the Fifth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Manchester, England, April, 2000.

Guichard F., *La réorganisation dynamique dans les systèmes multi-agents*, Thèse de Doctorat Informatique, Université de Savoie, 1996.

Helin H., Laukkanen M., « Performance Analysis of Software Agent Communication in Slow Wireless Networks », *Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN 2002)*, Austin, p. 354-361, Oct, 2002.

Humphrys M., « Towards self-organizing Action Selection », *in* S. Steel (ed.), *14th Workshop of the UK Planning and Scheduling Special Interest Group (SIGPLAN 14)*, 1995.

Iglesias C., Garrijo M., Gonzales J., « A survey of agent Oriented Methodologies », *Proceedings of ATAL 98 - Workshop on Agent Theories, Architectures, and Languages*, vol. LNAI 1555, Springer-Verlag, Paris, France, p. 163-176, July, 1998.

Ishida T., Gesser L., Yokoo M., « Organization Self-Design of Distributed Production Systems », *IEEE Transactions on Knowledge and Data Engineering*, vol. 4(2), p. 123-134, April, 1992.

Jamont J. P., Occello M., « Using organizational structures emergence for maintaining functional integrity in embedded systems networks », *2004 IFIP Working Conference on Artificial Intelligence Applications and Innovations*, Kluwer academic publishers, Toulouse, France, Aout, 2004.

Jamont J. P., Occello M., Lagreze A., « A Multiagent System for the instrumentation of an underground hydrographic system », *Proceedings of IEEE International Symposium on Virtual and Intelligent Measurement Systems*, IEEE Instumentation and Measurement Society, Mt Alyeska Resort, AK, USA, p. 20-25, May, 2002.

Jean M. R., « Emergence et SMA », *in* J. Quinqueton, M.-C. Thomas, B. Trousse (eds), *5èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Hermes Science, Nice, p. 323-342, Avril, 1997.

Johnson D. B., Maltz D. A., « Dynamic Source Routing in Ad Hoc Wireless Networks », *in* T. Imielinski, H. Korth (eds), *Mobile Computing*, Kluwer Academic Publishers, p. 153-181, 1996.

Kornman S., *SADE : un Système Réflexif de Surveillance à Base de Connaissances*, Thèse de Doctorat Informatique, Université Paris 6, 1993.

Langley B., « Discovery of infrastructure in multi-agent systems », *Proceedings of second international joint conference on autonomous agent and multiagent systems*, Melbourne, Australia, 2003.

Lee O.-K., Benford S., « An explorative model for federated trading in distributed computing environments », *Proceedings of the International Conference on Open Ditributed Processing*, Brisbane, Australia, p. 179-189, February, 1995.

Legras F., « Écoute flottante et communications locales pour la formation de groupes », *JFSMA - Journée Francophones sur les Systèmes Multi-Agents*, Hermès, p. 31-43, 2002.

Lemlouma T., *Le routage dans les réseaux ad-hoc*, Mémoire de Master, Université des Sciences et de la Technologie Houari Boumèdiene, Alger, Algérie, September, 2000.

Marcenac P., « Emergence of Behaviors in Natural Phenomena Agent-Simulation », *Complexity International Review*, vol. 3, 1996.

Marcenac P., *Modélisation et Simulation par agents - Application aux systèmes complexes*, Mémoire d'Habilitation à Diriger les Recherches en Informatique, Université de la Réunion, decembre, 1997.

MARCIA, « Auto-organisation := évolution de structure(s) », *PRC-GDR Intelligence artificielle*, Hermes, p. 139-152, 1996.

Milanosic N., Malek M., Davidson A., Milutinovic V., « Routing and Security in Mobile Ad Hoc Networks », *in* D. L. Carver (ed.), *IEEE Computer*, vol. 37, IEEE Computer Society, p. 61-65, February, 2004.

Minar N., Kramer K. H., Maes P., *Cooperating Mobile Agents for Dynamic Network Routing*, Springer-Verlag, chapter 12, 1999. Software Agents for Future Communications Systems.

Müller J.-P., Van Dyke Parunak H., « Vers une méthodologie de conception de systèmes multi-agents de réolution de problémes par émergence », *Actes des Journées Francophones en Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, Hermès, Pont-à-Mousson, France, Novembre, 1998.

Occello M., Demazeau Y., Baeijs C., « Designing Organized Agents for Cooperation in a Real Time Context », *in* A. Drogoul, M. Tambe, J. Singh (eds), *Collective Robotics*, vol. LNCS/LNAI 1456, Springer-Verlag, p. 25-73, March, 1998.

Occello M., Koning J. L., Baeijs C., « Conception de systèmes multi-agents : quelques éléments de réflexion méthodologique », *Technique et science informatiques*, vol. 20, n° 2, p. 233-263, 2001.

Perkins C. E., Royer E. M., Das S., « Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers », *Proceedings of ACM Conference on Communication Architectures, Protocols and Applications*, ACM, p. 234-244, August, 1994.

Petriu E. M., Patry G. G., Whalen T. E., Al-Dhaher A., Groza Z., « Intelligent Robotic Sensor Agents for Enviroment Monitoring », *Proceedings of the IEEE International Symposium on Virtual and Intelligent Measurement Systems*, p. 14-19, May, 2002.

Picard G., Gleizes M.-P., « The ADELFE Methodology - Designing Adaptive Cooperative Multi-Agent Systems », *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering handbook*, Kluwer Publishing, p. 157-176, 2004.

PLEIAD, « Vers une taxinomie de vocabulaire pour les systèmes multi-agents », *Actes des 1ères Journées Nationales du PRC-IA sur les Systèmes Multi-Agents*, Decembre, 1992.

Pujolle G., *Les Réseaux, 3éme édition mise à jour*, Eyrolles, 2000.

Ratsimor O. V., Chakraborty D., Tolia S., Kushraj D., Kunjithapatham A., Gupta G., Joshi A., Finin T., « Allia : Alliancebased Service Discovery for AdHoc Environments », *ACM Mobile Commerce Workshop*, ACM, Atlanta, USA, September, 2002.

Royer E. M., Perkins C. E., « Multicast Operation of the Ad hoc On-Demand Distance Vector Routing Protocol », *Proceedings of MobiCom*, ACM, p. 207-218, Août, 1999.

Russel S., Norvig P., *Artificial Intelligence : a Modern Approach*, Prantice-Hall, 1995.

Schira O., *Conception d'une couche physique adaptative pour réseau de terrain en vue de l'instrumentation de réseaux hydrographiques souterrains.*, Mémoire de DEA Signal Image Parole Télécom, INPG, Grenoble, 2003.

Smith R. G., « The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver », *IEEE transactions on computers*, vol. C-39 n12, p. 1104-1113, december, 1980.

So Y.-P., Durfee E. H., « Designing tree-structured organizations for computational agent », *Computational and Mathematical Organization Theory*, vol. 2, p. 219-246, 1996.

Steels L., « Cooperation between distributed agents through self-organisation », *in* Y. Demazeau, J.-P. Müller (eds), *Decentralized AI*, Elsevier Science Publishers B.V. (North Holland), p. 175-196, 1990.

Vercouter L., *Conception et mise en oeuvre de systèmes multi-agents ouverts et distribués*, Thèse de Doctorat, Université de Jean Monnet et Ecole des Mines de Saint-Etienne, December, 2000.

Wood M. F., DeLoach S. A., « An overview of the MultiAgent Systems Engineering Methodology », *The First International Workshop on Agent-Oriented Software Engineering*, Springer-Verlag, New-York, USA, p. 207-221, 2000.

Wooldridge M. J., « Intelligent Agents », *in* G. Weiss (ed.), *Multiagent systems : A modern approach to Distributed Artificial Intelligence*, MIT Press, p. 3-51, 1999.

Wooldridge M. J., Jennings N., « Intelligent Agents : Theory and Practice », *Knowledge Enginneering Review*, vol. 10, p. 115-152, January, 1995.

Wooldridge M. J., Jennings N. R., Kinny D., « The GAIA methodology for agent-oriented analysis and design », *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3, Kluwer Academic Publishers, p. 285-312, 2000.

Zhang W., Deng Z., Wang G., Wittenburg L., Xing Z., « Distributed Problem Solving in Sensor Networks », *AAMAS'02*, p. 15-19, July, 2002.

Article reçu le 31 mars 2004
Version révisée le 26 avril 2005

Jean-Paul Jamont *est diplomé de l'Ecole Supérieure d'Ingénieurs d'Annecy. Il est actuellement attaché temporaire d'enseignement et de recherche à l'Université Pierre Mendès-France et chercheur au Laboratoire de conception et intégration de systèmes (LCIS) de l'Institut national polytechnique de Grenoble (INPG) à Valence où il termine sa thèse de doctorat en informatique. Ses travaux portent sur l'utilisation de méthodes multiagents pour la conception de systèmes complexes coopérants embarqués.*

Michel Occello *est Professeur à l'Université Pierre Mendès-France et mène ses recherches au sein du laboratoire LCIS de l'INPG à Valence. Ses travaux portent plus particulièrement sur les systèmes multiagents hybrides et la méthodologie de la conception des systèmes multiagents.*