

Agents intelligents et gestion de réseaux*

XIII.1 – Introduction

Dans un proche avenir, tous les ordinateurs seront interconnectés. On trouvera des réseaux non seulement dans les grandes compagnies, comme c'est le cas aujourd'hui, mais également dans des bâtiments intelligents et même à domicile, pour piloter des grille-pain ou des machines à laver (17). Comme les autres systèmes, ces réseaux devront être contrôlés et gérés. De plus, il est clair que ces réseaux personnels (e.g. réseaux domotiques) ou les petits réseaux d'entreprise, ne nécessiteront pas les mêmes systèmes de gestion en termes de complexité, de coût ou de ressources. A présent, si toutes les ressources mises en réseau sont gérables (principalement via SNMP), il n'existe qu'un petit nombre de plates-formes de gestion, et toutes ont été conçues pour des réseaux de grande taille. Le paradigme d'agent intelligent semble, dans ce contexte, être une solution pouvant se révéler économique, flexible et adaptable. Même si des centaines de publications ont été faites sur le monde des agents intelligents, seulement un petit nombre d'entre elles traite des applications au monde de la gestion de réseaux. Cet état de fait est en train de changer, et la communauté de l'administration de réseaux prend conscience que « au fur et à mesure que nous avançons dans l'ère de l'information, toute entité qui traite de gestion d'information sans investir sur la technologie agents est peut-être en train de se faire hara-kiri » (30).

Dans ce chapitre, nous présentons l'état de l'art en matière d'utilisation des agents intelligents (AI) en gestion de réseaux. Celui-ci est composé de trois parties.

La première partie est divisée en deux sections : la première passe en revue les principes de la gestion de réseaux ; la seconde présente les caractéristiques des agents intelligents et leurs propriétés.

* Ce texte a été rédigé par Morsy M. CHEIKHROUHOU, Pierre CONTI et Jacques LABETOULLE.

La seconde partie expose l'état de l'art concernant les agents intelligents dans le domaine de la gestion de réseaux. La première section (section XIII.4) traite d'une approche importante pour l'administration de réseaux, appelée la gestion par délégation (MbD ou *Management by Delegation*). Elle décrit les travaux qui ont exploité le concept de MbD pour construire des agents intelligents et discute des améliorations nécessaires liées à la notion d'AI. La section suivante (section XIII.5) s'intéresse à l'utilisation d'agents mobiles pour la gestion de réseaux. Les aspects architecturaux des agents intelligents utilisés en administration de réseaux sont traités dans la section XIII.6. Après la présentation des approches qui utilisent des architectures agent, une analyse critique des forces et faiblesses est faite, concernant l'application au domaine de la gestion de réseaux. La section XIII.7 présente les techniques de coopération entre agents, et les expériences pratiques en administration de réseaux. La section XIII.8 présente une application d'agent d'interface développée pour la supervision d'un réseau.

Enfin, la troisième partie est constituée de la section XIII.9, qui donne une synthèse générale des principaux aspects abordés, avant la conclusion de la section XIII.10.

XIII.2 – La gestion de réseaux

La gestion de réseaux devra évoluer rapidement, afin de satisfaire les besoins des utilisateurs. Il suffit, pour s'en convaincre, de lire les comptes rendus des grands congrès internationaux (i.e. (19)) sur le sujet. Il en ressort un besoin pressant de solutions flexibles et adaptables en taille. Le paradigme d'agent intelligent, comme cela a été précisé dans l'introduction, semble être une solution prometteuse, même si aujourd'hui cette technique n'a été que très peu explorée. Seulement deux pour cent des produits « agents » et de la recherche sur les agents sont dédiés à la gestion de réseaux, selon *The Agent Organization*. Pourquoi ?

Nous pensons qu'il y a deux raisons à ce faible pourcentage :

- D'abord, la gestion de réseaux est fortement influencée par l'approche OOP (*Object Oriented Programming*), et d'énormes efforts de développement sont toujours en cours dans cette direction. La plupart d'entre eux se concentrent sur CORBA, qui supporte logiquement les techniques OOP utilisées.
- Ensuite, si les approches orientées agent ont clairement un grand potentiel, comme c'était le cas il y a dix ans pour les techniques orientées objet, elles représentent un saut technologique qui n'a pas encore fait preuve de sa maturité.

Dans ce contexte, nous posons la question : est-ce une bonne idée de consacrer du temps et de l'énergie à un nouveau paradigme de développement ? Afin de répondre, nous nous concentrerons tout d'abord sur les objectifs principaux de la gestion de réseaux et essaierons de comprendre les raisons de l'insatisfaction des utilisateurs vis-à-vis des solutions actuelles.

XIII.2.1 – Les objectifs de la gestion de réseaux

En première approche et sans vouloir aller très en profondeur, l'on peut dire que l'utilisateur de réseaux souhaite disposer de communications rapides, sécurisées et fiables, le gestionnaire de réseaux souhaite pouvoir configurer et contrôler les accès et les ressources, et enfin le responsable d'entreprise espère des coûts aussi faibles que possible. Pour aller un peu plus loin, considérons la classification des besoins des utilisateurs de gestion de réseaux, telle que proposée par (45) et que l'on trouvera dans (43) :

- *Contrôle des investissements stratégiques* : les réseaux et les ressources informatiques distribuées sont des éléments vitaux pour la majorité des entreprises. Sans contrôle effectif, ces ressources ne fourniront pas le retour sur investissement nécessaire.
- *Contrôle de la complexité* : la croissance continue du nombre de composants réseau, d'utilisateurs, d'interfaces, de protocoles et de fournisseurs fait craindre aux responsables une perte de contrôle sur ce qui est connecté et sur l'utilisation des ressources.
- *Amélioration des services* : les utilisateurs finaux attendent des services toujours plus efficaces, au fur à mesure de l'accroissement de la quantité d'information accessible et de la distribution des ressources.
- *Prise en compte de besoins variés* : L'information et les ressources de calcul doivent être accessibles à un large spectre d'utilisateurs, à travers des applications variées avec des besoins spécifiques en termes de performance, de disponibilité et de sécurité. La personne en charge de la gestion de réseaux doit veiller à la bonne utilisation des ressources pour satisfaire tous ces besoins.
- *Réduction du temps d'indisponibilité* : en parallèle avec l'accroissement des ressources réseau, la disponibilité doit être aussi proche que possible des 100 %. En plus de la simple redondance, la gestion de réseaux a la charge de veiller sur ce facteur devenu indispensable.
- *Contrôle des coûts* : l'utilisation des ressources doit être contrôlée et gérée, afin de satisfaire les besoins des utilisateurs, et cela à moindre coût.

A cause de leur hétérogénéité (et des raisons données ci-dessus), les réseaux sont difficiles et complexes à gérer. Les systèmes de gestion doivent prendre en compte un nombre croissant de ressources, d'équipements, de protocoles, le tout sur des espaces de plus en plus larges. C'est la raison pour laquelle le groupe de travail ISO/TC 97/SC 21/WG4 a été créé au début des années 80 (35). Le travail a été long et difficile ; c'est pourquoi l'IETF, adoptant un point de vue beaucoup plus pragmatique, a spécifié une approche plus simple, basée sur le protocole SNMP et des MIBs (*Management Information Base*) standards. Ceux-ci ont rapidement conquis le marché, conduisant à des normes de fait. Contrairement à l'OSI (*Open System Interconnection*) l'approche architecturale est sommaire, la rendant inappropriée pour de grands réseaux.

XIII.2.2 – La gestion de réseaux et l’OSI

La gestion de réseaux a été étudiée de longue date par l’OSI, qui a défini une architecture et un ensemble de standards. Ceux-ci sont organisés en trois grands chapitres :

- Les communications pour la gestion de réseaux (CMIS/CMIP) (ISO/IEC 9595...)
- Les fonctions de gestion système (ISO/IEC 10164-*)
- Le modèle d’information de gestion (ISO/IEC 10165-*,10589,...).

Les cinq aires fonctionnelles de gestion

La gestion de réseaux OSI est subdivisée en cinq aires de gestion, qui sont fortement couplées. Des unités fonctionnelles sont définies, couvrant ces aires :

- *La gestion des configurations* : démarrage, arrêt, mise en place, lecture, modification de la configuration
- *La gestion des fautes* : détection, trace, localisation, analyse, prédiction, correction
- *La gestion des performances* : mesures, collecte de données, analyse, adaptation, tests
- *La gestion de la sécurité* : contrôles d’accès, transferts, tests
- *La gestion financière* : gestion des quotas, vérification, planification, facturation

L’OSI définit comment un « système de gestion » peut communiquer et agir directement sur un « système géré », qui peut être lui-même un autre système de gestion en charge d’un autre domaine du réseau ou un agent.

La gestion est vue à travers les cinq aires fonctionnelles, et utilise des « fonctions de gestion » pour réaliser les activités d’administration de réseaux. Ces fonctions utilisent les services du protocole CMISE/CMIP pour accéder à l’information de gestion. Les systèmes gérés sont représentés par des agents (agents CMISE), qui doivent eux agir sur les objets invoqués par le manager ou envoyer des reports d’exception quand c’est nécessaire.

En comparaison avec SNMP, le standard de gestion Internet, les normes OSI sont complexes et lourdes à mettre en œuvre.

XIII.2.3 – La gestion de réseaux Internet

La gestion de réseaux Internet s’appuie sur un protocole de communication rustique entre des managers et des agents, pour accéder à des Bases d’Informations de gestion (MIBs). Celles-ci contiennent des variables, gérées au niveau de l’agent. Cet accès simple et direct aux équipements à gérer explique la simplicité de l’architecture de gestion. La contrepartie est la difficulté à gérer de grands réseaux. En

effet, l'usage intensif de la scrutation de variables consomme une grande bande passante ; il faut beaucoup d'échanges pour retrouver les données des MIBs ; la sécurité est sommaire ; les managers ne peuvent pas communiquer entre eux. L'IETF a donc décidé de proposer des versions améliorées de SNMP, pour corriger ces défauts. Les principales extensions des versions V2 et V3 portent sur :

- *Le protocole* : nouvelles primitives *GetBulkRequest* et *Inform Request*
- *Des extensions du système d'information* : nouveaux types de données, tables dynamiques
- *Des capacités de communication de Manager à Manager* : *Inform Request*
- *Un système de nommage* : *SNMP Engine Id*

La seconde amélioration apportée par l'IETF concerne la spécification RMON développée pour l'analyse de segments de RLE (LANs), et de sa version RMON2. SNMP V2 et V3, comme RMON sont un pas vers la gestion décentralisée, sans toutefois la rendre possible à grande échelle. Malgré les défauts énumérés ci-dessus, la solution Internet de gestion de réseaux est de loin la plus répandue, principalement à cause de sa simplicité et de son faible coût de mise en œuvre.

XIII.2.4 – Les systèmes de gestion de réseaux : le marché

Tous les réseaux sont différents. Les divers composants – topologie, médias support, architectures, équipements et logiciels – ont conduit à une explosion des réalisations. Cela explique pourquoi il est pratiquement impossible de trouver des études d'évaluation des solutions de gestion de réseau.

Quelques tentatives ont été faites (14), mais elles restent limitées, et même si le réseau de test est relativement simple, les résultats n'ont pas été très encourageants. Plusieurs produits sont basés sur le produit HP OpenView, complété par des applications de partenaires. D'autres solutions ont été développées par Bull, IBM, Compacq, ...

Dans sa conclusion, I.G. Ghetie (11), qui a fait une étude comparative des produits majeurs sur le marché, observe qu'il y a un manque de coopération et d'intégration entre les applications de gestion de réseaux, même au sein de la même plateforme.

En conclusion sans entrer dans les détails, on peut retenir les points suivants :

- Aucun des produits ne sait gérer facilement des réseaux hétérogènes (à cause de la faiblesse des normes)
- L'adaptation au facteur d'échelle est difficile, à cause de la faiblesse des outils de développement
- Les systèmes sont de gros consommateurs de ressources (e.g. scrutation SNMP)
- Tous les produits sont chers.

Les usagers sont toujours à la recherche d'une plus grande simplicité, d'une meilleure intégration entre des applications plus coopératives, de plates-formes économiques et extensibles.

XIII.2.5 – Les agents intelligents et la gestion de réseaux

Il faut donc repenser la manière de définir l'architecture de gestion de réseaux, en gardant en tête l'idée d'utiliser des agents intelligents. De nouvelles voies ont déjà été ouvertes : par exemple, Aiko Pra (35), qui propose une sorte d'approche prototype, ou D. Stevenson (43), qui part du point de vue *Help Desk* pour définir les besoins en gestion et qui affirme que l'intelligence artificielle doit être appliquée à tous les niveaux de la gestion. L'architecture que nous préconisons est proche de ce dernier point de vue.

Plusieurs études ont été menées autour de la notion de politique de gestion, basées sur des approches objet. Cependant, les prototypes développés sont très influencés par OOP, et semblent restrictifs par rapport au potentiel des agents intelligents. Par exemple, le Département d'Informatique de l'Imperial College à Londres travaille sur des « politiques d'obligation – interdictions », et une infrastructure de gestion de systèmes distribués basée sur la théorie des rôles. L'intérêt de ces études est qu'elles introduisent une architecture de gestion à base d'agents intelligents, tout en maintenant la simplicité d'utilisation du protocole de gestion Internet.

XIII.3 – Principes des agents intelligents

XIII.3.1 – Origines

Il semble certain aujourd'hui que nous nous approchons d'une période où l'on pourra espérer trouver une aide efficace de la part de systèmes logiciels capables de prendre des décisions et d'agir d'une façon autonome, sans supervision permanente. Les agents font partie de ces techniques, même s'ils portent des noms variés : agents logiciels, agents intelligents, agents mobiles ou agents personnels. Alors que de gros efforts sont faits pour le développement des solutions orientées objets, pourquoi cet intérêt croissant pour le monde des agents ? La lecture de la littérature sur le sujet nous donne trois éléments d'explication de ce phénomène :

1. L'Intelligence Artificielle (IA), considérée pendant longtemps comme une activité logicielle marginale, produisant des solutions complexes et difficiles à développer, a été influencée par les approches orientées objets. Les chercheurs impliqués en IA ont compris l'intérêt de travailler sur des entités logicielles plus simples, plus souples et réutilisables. « Un groupe d'agents est plus que la somme des capacités de chacun », selon J. Muller dans (29).

2. Le monde des objets, après l'intégration des environnements réseau et le développement des architectures d'objets distribués, éprouve le besoin d'aller un pas plus loin, vers une certaine « autonomie » et une certaine « mobilité » des objets. De nouveaux langages objet ont été développés, permettant la mobilité, et le vocable « agent » est apparu avec la définition suivante : « un agent est un programme qui agit automatiquement, pour le compte d'une personne ou d'une organisation » (32).
3. Les réseaux utilisent depuis longtemps des logiciels appelés « agents » dans le domaine de l'administration de réseaux, même si la définition n'est pas la même. Les agents SNMP et CMIP sont des modules logiciels qui traitent l'information représentant un ou plusieurs éléments de réseaux, et échangent ces informations à travers les protocoles SNMP ou CMIP. Il y a des expériences visant à remplacer les simples agents SNMP ou CMIP par des substituts intelligents (34), mais sans remettre en cause les grands principes de la gestion de réseaux ce qui ne représente pas une avancée significative. D'autre part, l'expansion des réseaux a réorienté la façon d'utiliser les ordinateurs. Les ressources sont maintenant réparties sur des systèmes différents et le plus souvent hétérogènes, et doivent être partagées. L'information est répartie sur le réseau et a besoin d'être rassemblée pour être traitée. L'architecture classique client/serveur, basée sur un serveur unique, doit être remise en cause. L'approche agent a été prise en compte et est considérée comme LA solution par des sociétés comme Oracle avec son architecture client/agent/serveur.

Nous pensons que, même si une telle évolution n'est pas encore universellement reconnue, tous ces domaines doivent converger, puisqu'ils utilisent les mêmes environnements réseau et que des raisons économiques vont imposer la préservation des investissements.

Jusqu'à présent, les origines multiples des agents ont rendu difficile un consensus sur une bonne définition. Par exemple, lors du *workshop ATAL'96 (Agent Theories, Architectures and Languages)*, un temps important a été passé sur la définition et la taxonomie des Agents. Une définition a été donnée par le titre de la publication de Mario Tokoro « *An agent is an Individual that has Consciousness* ». Mais la conclusion générale est qu'il n'y a pas urgence à définir la notion d'agent d'une manière unique.

Dans cette étude, nous ne chercherons donc pas à entrer dans ce débat, et nous utiliserons le concept d'agent intelligent comme un ensemble de propriétés, et nous nous servirons du mot agent comme une abréviation d'agent intelligent.

XIII.3.2 – Propriétés des agents

Le premier but de cette approche par les propriétés est d'obtenir des agents rencontrés lors de notre recherche. Ce ne sera certes pas suffisant pour mieux définir les concepts, comme nous l'avons vu précédemment. Ceci dit, il est important de bien comprendre quelles propriétés un agent peut posséder, et nous donnons ci-dessous les définitions de ces propriétés.

- *Autonomie*

« Capacité d'agir seul, indépendance : un responsable d'entreprise a une autonomie totale dans son domaine » (traduit librement de Oxford Advanced Learner's Dictionary). L'agent décide lui-même quand et sous quelles conditions il va exécuter quelle action. « Un agent autonome est un système situé dans un environnement, et en faisant partie, qui mesure cet environnement et agit sur lui, dans le temps, en fonction de son propre agenda, et pour prévoir ce qu'il mesurera dans le futur » (7). M. Wooldrige est en désaccord avec la définition d'un agent de Franklin et Graesser's, et dans « Une réponse à Franklin et Graesser » précise que l'autonomie est nécessaire en tant que propriété, mais que le sont également les comportements réactifs, proactifs et sociaux. Ce point de vue soulève la question : « Qu'est-ce que l'autonomie sans comportement proactif, délibératif ou réactif ? ». On peut répondre que les virus, les virus réseau (worms) sont des exemples d'agents autonomes et mobiles.

- *Communication*

L'une des propriétés clés des agents est leur capacité à dialoguer avec un semblable, avec un humain (agents d'interface) ou avec un équipement. Certains langages de communication ont ainsi été définis, parmi lesquels :

- Tableaux noirs : les agents écrivent et lisent sur des espaces (*blackboard*).
- KQML : *Knowledge Query and Manipulation Language* est un langage et un protocole pour échanger des informations et des connaissances à travers des *performatives*.
- KIF : *Knowledge Interchange Format* est un format d'échange de connaissances
- COOL : dialogue structuré, basé sur KQML, utilisé pour coordonner des agents. (2).

- *Collaboration/Coopération*

Les agents sont dits collaboratifs quand ils savent travailler ensemble. Aussi un agent est-il capable de communiquer et négocier avec d'autres. Il a des qualités délibératives et peut coordonner ses actions avec ses semblables. Les agents collaboratifs sont particulièrement intéressants quand une tâche implique plusieurs systèmes à travers le réseau. La négociation est un élément clef pour les agents collaboratifs. La coopération peut exister sans la collaboration, mais la collaboration nécessite la négociation.

- *Délibération*

Connaître les objectifs et les règles, les appliquer sans attente d'instructions supplémentaires. M. Wooldrige and N. Jennings définissent un agent délibératif comme « celui qui contient un modèle de représentation symbolique du monde, et dans lequel sont prises des décisions (...) grâce à un raisonnement logique (ou au moins pseudo-logique), s'appuyant sur des reconnaissances de schémas et des manipulations symboliques » (49).

• *Mobilité*

Depuis l'apparition de Java, on observe une prolifération des agents mobiles, mais plusieurs types de mobilité sont possibles :

- La mobilité qui permet à un agent de se déplacer d'un système vers un système similaire,
- La mobilité qui permet à un agent de se déplacer vers un système de nature différente,
- La mobilité qui permet à un agent de suspendre son activité sur un système, de se déplacer vers un autre, puis de reprendre son activité,
- La mobilité qui permet à un agent de se déplacer lui-même, plutôt que d'être transporté,
- La mobilité qui est la duplication de l'agent sur un autre système (clonage),
- Le transport de la connaissance de l'agent sur un autre système.

En général, la mobilité est un mélange de ces définitions. Un des problèmes cruciaux de ces technologies est la faiblesse de la sécurité des agents mobiles.

• *Apprentissage*

Il y a au moins deux définitions de l'apprentissage :

- La mauvaise : on dit qu'un agent apprend s'il est capable d'acquérir de la connaissance (des données)
- La bonne : on dit qu'un agent apprend s'il est capable d'utiliser sa nouvelle connaissance pour modifier son comportement.

Malgré le fait qu'apprendre est un facteur important de l'intelligence, peu d'agents sont dotés de la capacité d'apprentissage. La plupart du temps, ils utilisent des règles fixes (précompilées) et des bases de connaissance.

L'objectif de l'apprentissage pour un agent est de pouvoir exécuter des tâches nouvelles, sans avoir à être interrompu. Différentes solutions d'apprentissage ont été étudiées et expérimentées.

Généralisation de la définition : elle consiste à observer son environnement pour en déduire des règles.

• *Proactivité*

« Les actions proactives ont pour objectif de provoquer des changements, plutôt que de simplement réagir à des changements ». Les agents proactifs suivent en général des plans d'action, ou au moins exécutent des règles quand l'environnement atteint des niveaux de seuils prédéfinis ou quand l'évolution de la tendance observée tendrait à outrepasser ces seuils.

Parfois, le terme proactif est utilisé dans le même sens que délibératif. Cependant, un agent est proactif parce qu'il lui a été demandé d'exécuter des tâches proactives, contrairement à des agents délibératifs qui décident eux-mêmes de devenir proactifs.

- *Réactivité*

Faire quelque chose quand un événement se produit, donc réagir à cet événement

- *Sécurité*

C'est la capacité à distinguer les amis des ennemis et à repérer les éléments contaminés.

- *Planification*

L'agent organise les actions à faire au cours de sa vie (par exemple, en fonction de priorités). Pour beaucoup de chercheurs, la planification est la propriété la plus importante pour un agent. La planification est utilisée par les agents délibératifs ou proactifs, en fonction de la connaissance qu'ils observent de l'environnement et des actions possibles qu'ils peuvent lui appliquer.

- *Délégation*

Un agent peut demander à quelqu'un d'autre (un autre agent) de réaliser pour lui un objectif ou une tâche. Cette capacité est importante pour équilibrer les charges, mais également pour agir sur un domaine éloigné.

Conclusion

Beaucoup de débats ont eu lieu autour des définitions données ci-dessus. Comme le soulignent de nombreuses publications, et à cause du manque de consensus, il est de première importance, au sein de tout projet, de bien s'entendre sur les définitions utilisées.

XIII.3.3 – Technologies agent

Le développement d'agents nécessite des langages pour les construire, des protocoles et des langages d'échange pour leur permettre de communiquer, ainsi que des langages pour décrire et transférer les connaissances de l'agent, ses croyances, buts, désirs ou intentions (29).

XIII.3.3.1 – La naissance d'un agent

L'une des idées fortes de notre discussion sur la gestion de réseaux est la capacité qu'un agent doit avoir de pouvoir s'exécuter dans différents environnements hétérogènes. Les agents ont donc besoin d'une infrastructure standard pouvant tourner sur chacun des systèmes hôtes possibles. Ils peuvent alors être développés comme sur une machine unique – la machine virtuelle (VM). Les langages les plus simples pour écrire les agents sont les langages de script, qui peuvent être interprétés (comme le langage *shell* sur Unix). Les langages de ce type les plus utilisés sont *Telescript de General Magic* (remplacé maintenant par *Odyssey* qui est

100 % Java), *Perl* et *Tcl* (*Tool Command Language*) de Sun, et bien sûr, Java, qui est un langage de script orienté objets. Une autre classe de langages provient de la programmation plus traditionnelle, comme LALO (*Agent Oriented Language*), ou des langages d'IA comme *Obliq* de DEC. Selon les langages et outils utilisés pour les créer, les agents naissent avec des « capacités intellectuelles » plus ou moins fortes.

XIII.3.3.2 – Les agents et la communication

Une fois créés, les agents ont besoin de communiquer avec des humains ou des congénères. En fonction des techniques utilisées, différentes solutions sont possibles, s'appuyant généralement, sur TCP/IP. KQML (voir Section XIII.3.2) est le langage le plus utilisé pour encapsuler les requêtes et les réponses entre agents, sur des protocoles comme IIOP, HTTP, et SMTP.

XIII.3.3.3 – Les agents et la connaissance

Un autre niveau de langage est nécessaire pour transporter la connaissance, que nous pourrions comparer à des langues naturelles. L'un des langages les plus utilisés est KIF (*Knowledge Interchange Format*), mais des langages spécifiques sont souvent développés au sein de projets pour des usages spécifiques.

Il est parfois difficile de comprendre que le paradigme des agents intelligents utilise une vision humaine et un discours (anthropomorphisme) pour représenter ce qui n'est que du logiciel. Cette vision est cependant normale et juste, dans le sens où, au stade actuel, on peut utiliser les agents intelligents comme des travailleurs, en leur donnant des responsabilités et de l'autonomie dans leurs actions. Dans les sections suivantes, nous passons en revue des expériences qui utilisent des agents pour résoudre des problèmes de gestion de réseau.

- *Adaptation d'échelle*

Quand les systèmes croissent fortement en taille ou que de nouveaux services sont ajoutés, des agents peuvent prendre en charge le nouvel environnement, en modifiant leurs rôles, leurs croyances et leur organisation.

- *Robustesse*

Une solution totalement distribuée de gestion de réseaux est plus efficace qu'une solution centralisée, car les opérations de gestion peuvent continuer même si une partie du réseau est déconnectée du site central de gestion.

- *Evolutivité*

Un nouvel agent peut facilement remplacer un ancien, sans arrêter l'activité

- *Performance*

Plus près des systèmes à gérer, la gestion distribuée permet une meilleure réactivité et même une proactivité, tout en consommant moins de ressources.

XIII.4 – Les agents intelligents et la gestion par délégation

XIII.4.1 – Préambule

Le paradigme de gestion par délégation (MbD – *Management by Delegation*) a été développé, à l'origine, en dehors du monde des agents. Cependant, nous avons décidé d'aborder cette question ici pour plusieurs raisons :

- Récemment, beaucoup de travaux sur le MbD ont été développés en utilisant des techniques d'agents intelligents,
- Nous pensons que le MbD est un concept riche, renforçant le rôle des agents en gestion de réseaux,
- Certains chercheurs pensent que des agents intelligents peuvent être développés en utilisant les concepts du MbD.

Nous allons donc expliquer les concepts et principes du MbD, les travaux liés dans le domaine des agents intelligents et conclure.

XIII.4.2 – Introduction

Les faiblesses des approches classiques et des protocoles en gestion de réseaux ont été identifiées très tôt. Les données sont distribuées sur des agents qui ont très peu de capacité de calcul, et le traitement des données est centralisé sur les gestionnaires. Il en résulte une grosse charge pour les serveurs, et les opérations de base de la gestion se limitent à des opérations SET et GET qui sont communiquées aux agents. Certains appellent cette approche du *micro-management* (51), entraînant un important trafic de données sur le réseau. Afin de corriger cette situation inconfortable et de définir une architecture plus flexible et adaptable, et dotée de réelles capacités temps réel, il est reconnu que les fonctions et opérations de gestion doivent être exécutées aussi près que possible des objets gérés. La gestion par délégation suit ce principe.

XIII.4.3 – Gestion par délégation

Le concept principal sur lequel repose la gestion par délégation est celui de serveur élastique (13). Un serveur élastique est un serveur dont les fonctionnalités peuvent être étendues et contractualisées dynamiquement durant l'exécution. Un protocole de délégation permet de passer les nouvelles fonctions au serveur élastique et de lui demander de lancer les exécutions. Le protocole permet également de garder le contrôle de chaque instance, d'arrêter ces dernières, de les redémarrer, etc. Les nouvelles fonctionnalités sont décrites dans des scripts, sans restriction sur le langage dans lequel ceux-ci sont écrits.

Appliqués à la gestion de réseaux, les serveurs élastiques se situent à un niveau intermédiaire entre les gestionnaires et les agents (comme les gestionnaires intermédiaires), avec un rôle similaire à des agents proxy (43). Ces serveurs élastiques sont appelés MbD agents (51), MAD agents (*Manager Agent Delegation*) (51), *managing agents* (46), *delegation and flexible agents* (26) ou *elastic agents* (15).

Une autre approche pourrait constituer à faire évoluer les agents classiques SNMP/CMIP vers des agents élastiques. Mais elle n'a pas été retenue, car les manufacturiers préfèrent garder les systèmes actuels en l'état et définir de nouveaux composants ad hoc.

Le gestionnaire utilise le protocole de délégation pour charger les scripts de gestion dans les agents flexibles. Par le même protocole, il demande l'autorisation d'instancier certains scripts délégués et de les exécuter. Ainsi, l'opération de gestion décrite dans le script peut s'exécuter de façon autonome dans le même environnement que pour les objets gérés. Le gestionnaire peut contrôler l'exécution grâce à des primitives spécifiques du protocole.

Par ailleurs, Goldszmidt (13) a appelé ces scripts des « agents délégués ». D'autres recherches explorent le paradigme de MbD dans le cadre de la gestion de réseaux. Par exemple, (20) propose un environnement de script de type tableur pour SNMP. Le langage correspondant permet à un gestionnaire de définir des opérations de calcul que doit exécuter l'agent. Chaque cellule dans le tableur contient une formule de calcul portant sur des données, par exemple, des variables de MIB. Les formules peuvent être insérées, modifiées ou supprimées en fonction des besoins.

Une autre application est décrite dans (39). Chaque agent contient des objets fonctionnels (leur comportement est lié à des fonctions de gestion) qui utilisent des objets gérés et permettent des communications avec d'autres agents. Ainsi, un gestionnaire se contente de déléguer des squelettes de script pour invoquer des fonctions de gestion. Pour exécuter une instance, l'invocation de la fonction de gestion est liée dynamiquement aux objets fonctionnels. Les mêmes fonctions de gestion sont ainsi mises en œuvre différemment sur les différents agents en fonction des équipements spécifiques sur lesquels elles portent.

D'autres travaux utilisent différents environnements de scripts comme *SQL-like* présenté dans (52), *event-driven scripts* présenté dans (19) et le *Tcl based scripting language* présenté dans (9).

Une autre approche a été retenue par Trommer et Knopka (46), qui ont choisi de doter les agents d'une capacité de calcul de règles. Les règles peuvent être transmises du manager vers l'agent, dotant celui-ci d'un comportement intelligent. Ces agents deviennent des agents flexibles, du fait que des règles peuvent être ajoutées, supprimées ou modifiées dynamiquement.

Se référant aux propriétés et à la définition des agents (49), Mountzia (26) propose de faire évoluer les agents de délégation vers des agents intelligents, en leur donnant des capacités d'autonomie, de réactivité, de proactivité, de mobilité et la capacité d'apprendre. Mais les idées développées en sont encore à un stade préliminaire.

XIII.4.4 – Synthèse et discussion

Les mises en œuvre du concept de gestion par délégation telles que présentées ci-dessus ne correspondent pas totalement à notre définition des agents intelligents. Les propriétés telles que présentées dans la section XIII.3.2 ne sont, en général, pas présentes. Les seules propriétés affirmées sont l'autonomie, la délégation et la coopération. Cependant, les agents délégués ne sont pas autonomes dans le sens où les scripts de gestion doivent préciser ce que l'agent doit faire. Les agents ne peuvent agir en dehors d'un ordre précis. Ils n'ont pas la possibilité de décider par eux-mêmes ce qu'il faut faire, quand et comment le faire. La même critique s'applique à la délégation : seuls des opérations de bas niveau sont déléguées. Enfin, ces agents ne sont pas coopératifs, dans la mesure où ils ne décident pas dynamiquement d'agir d'une manière coordonnée, en fonction d'un objectif global.

Il n'en reste pas moins que le concept de gestion par délégation est puissant et peut être exploité de différentes façons. Il peut servir de support pour concevoir des agents intelligents, cela par deux voies possibles : d'une part, en suivant les suggestions de (26) qui conçoit des agents délégués comme des agents logiciels, et d'autre part, en utilisant la notion de script pour mettre en œuvre la notion de délégation. Cette façon de procéder conduit aux agents mobiles. Dans les deux cas, de gros efforts restent à faire. Il faut passer de la notion de simple délégation de scripts à celle de délégation d'objectifs, qui est d'un niveau nettement plus élevé.

XIII.5 – La gestion de réseaux avec des agents

XIII.5.1 – Généralités

Les agents mobiles (MA) sont capables de sauter d'un nœud vers un autre au cours de leur activité. Ils sont donc capables d'utiliser de l'information extraite au cours de leurs visites sur des sites pour ajuster leur comportement. Ils peuvent déplacer leur activité là où les ressources nécessaires peuvent être trouvées. Dans ce qui suit, nous expliquons comment ces propriétés peuvent être exploitées en gestion de réseaux et nous présentons quelques voies de développement.

XIII.5.2 – Les avantages des agents mobiles

Dans (24), Thomas Magedanz s'est intéressé à l'utilisation des agents mobiles pour la gestion de réseaux. Les MA peuvent encapsuler des scripts de gestion et être envoyés à la demande là où c'est nécessaire. Un agent peut être envoyé dans un domaine réseau pour voyager à travers les éléments, afin de collecter des données de gestion et revenir avec l'information filtrée et traitée. Il s'agit donc d'un substitut pour l'exécution d'opérations de gestion de bas niveau et le traitement de ces données. Ainsi, si l'agent est capable d'extraire l'information utile sur chaque élément pour la rendre concise, il peut rester petit en taille et ainsi économiser de la bande

passante. Il s'agit là de l'argument principal pour justifier l'utilisation des agents mobiles.

D'une manière similaire, un agent mobile peut être utilisé pour transporter une opération de gestion. Au lieu d'invoquer une opération à distance, le gestionnaire peut encapsuler les opérations dans un agent mobile qu'il envoie au bon endroit. Si les opérations nécessitent de nombreux échanges pour la prise de décision, l'agent mobile peut être économique en bande passante, s'il est capable de prendre les décisions localement.

Ces idées ont été exploitées par FTP Software (40). *L'IP Auditor* est une application qui envoie des agents dans le réseau dans le but de collecter des informations sur la configuration et l'état opérationnel des équipements. *L'IP Distributor* est une autre application qui utilise des agents mobiles pour gérer les charges payantes sur le réseau.

Une autre application originale et concrète des MA est présentée dans (1). Des agents mobiles sont utilisés pour contrôler la congestion de trafic dans un réseau à commutation de circuits. Un premier type de MA, appelé agent parent, navigue d'une manière aléatoire entre les nœuds et collecte des informations sur la charge. Ces agents en déduisent une estimation de la charge moyenne sur les nœuds. Ils sont alors capables d'identifier les nœuds qui ont une charge supérieure à la moyenne. Une fois un tel nœud identifié, un deuxième type de MA appelé « agent de charge » (*load-balancing agent*) est créé. Il sait mettre à jour les tables de routage sur les nœuds voisins (grâce à un algorithme d'optimisation), de façon à réduire la charge du nœud gestionné.

Dans une étude très intéressante sur le code mobile, (3) étudie l'avantage d'utiliser des agents mobiles quand l'administrateur de réseau ne peut se connecter que par des moyens non fiables ou trop chers. Il crée *off-line* un agent mobile, se connecte sur le réseau pour le lancer, ferme la connexion et se reconnecte plus tard pour le récupérer avec l'information collectée. Ce principe a été mis en œuvre dans *Astrolog* (37) pour supporter des gestionnaires mobiles et apporter une flexibilité au processus de gestion. La connexion peut même se faire via un GSM.

(24) et (27) suggèrent d'autres applications possibles des MA. Un fournisseur de services ou de réseaux peut envoyer des agents mobiles au point d'accès d'un client pour adapter son équipement terminal. Ces agents servent également à d'autres tâches comme la facturation et la capture des exigences de l'utilisateur. (16) suggère également d'utiliser des MA pour la création immédiate de nouveaux services et l'adaptation aux besoins client.

XIII.5.3 – Discussion

L'utilisation d'agents mobiles pour la gestion de réseaux semble donc prometteuse. Il faut cependant prendre quelques précautions avant de généraliser leur déploiement. En ce qui concerne l'utilisation de la bande passante, il n'est pas évident que le transport d'un agent doté de capacités substantielles soit plus économique que le transport des informations nécessaires vers un serveur central. On

peut, en effet, constater que le traitement des informations est souvent complexe, par exemple, pour des applications de type corrélation d'événements ou détection de fautes. Pour contourner le problème, il faudrait disposer de langages très concis et à haut potentiel descriptif. Cela nous amène à considérer deux options : soit l'environnement dans lequel on fait tourner l'agent est riche et complexe, soit l'agent n'a que des capacités très limitées. La première option conduirait à des solutions coûteuses, tandis que la seconde limite le domaine d'application des agents mobiles. La conception d'un environnement efficace pour assurer la mobilité des agents n'est pas une chose facile. La notion de machine virtuelle Java est un pas dans la bonne direction, mais reste encore éloignée du concept d'agent.

Enfin, l'un des problèmes cruciaux qui reste ouvert par rapport au concept d'agent mobile est celui de la sécurité. Il doit être considéré sous deux aspects : celui de la sécurité des hôtes traversés, et celui de la protection des agents mobiles contre les hôtes hostiles. Si pour le premier aspect quelques réponses ont été données, le deuxième n'est encore que très peu étudié. Pourtant la sécurité est un facteur primordial dans un domaine sensible comme celui de la gestion de réseaux. Seul (40) prétend avoir une solution avec les agents *IP Auditor's* et *IP Distributor* (voir sous-section précédente).

Les systèmes d'agents mobiles, même s'ils sont très attractifs sur le plan de leur approche, doivent donc faire l'objet d'études approfondies avant tout déploiement. Par exemple, utiliser des agents mobiles pour l'exécution de tâches prédéterminées, tel que présenté dans (1), est sans doute élégant mais pas nécessairement optimal. L'utilisation d'un protocole pour échanger l'information et les tables de routages aurait été très efficace.

XIII.6 – Agents réactifs, délibératifs et hybrides en gestion de réseaux

XIII.6.1 – Préambule

En fonction de son architecture, un agent peut être réactif, délibératif ou hybride (49). Alors qu'un agent réactif agit en fonction d'un événement ou d'une situation, un agent délibératif agit en fonction de décisions prises à la suite d'un raisonnement sur la situation de l'environnement. Un agent hybride combine les deux propriétés. Les sous-sections qui suivent traitent de ces trois types d'agents et la dernière en présente une synthèse.

XIII.6.2 - Agents réactifs

L'application présentée dans la section XIII.5 utilise des agents mobiles pour la gestion de la congestion. Les agents utilisés sont de type réactif (1). Ni les agents parents ni les agents de charge n'utilisent de modèle du réseau qu'ils gèrent. Ils se

contentent d'observer l'état du réseau et réagissent en conséquence. L'utilisation d'agents réactifs est particulièrement adaptée à cette application, car les agents mobiles ont besoin d'être petits pour ne pas charger le réseau du fait de leur mobilité. Les agents parents ne réagissent pas tant qu'ils n'ont pas observé un nœud surchargé. La décision de réagir en créant un agent de charge est prise grâce à une simple comparaison entre l'utilisation du nœud visité et la charge moyenne mesurée au cours des déplacements. Les agents de charge qui mettent à jour les tables de routage décident en fonction d'un algorithme qui ne suppose aucun raisonnement. Aucun des agents pris séparément, que ce soit un parent ou un agent de charge, n'est capable de gérer le réseau. Cette application est donc un bon exemple qui montre comment un système multi-agents a des capacités plus fortes que tous les agents réactifs pris individuellement.

D'autres recherches portent sur la construction d'infrastructures orientées agents pour la gestion de réseaux et utilisent les agents réactifs à cause de leur simplicité. Dans (44), des agents intelligents sont utilisés pour mettre en œuvre des politiques de gestion distribuées. Dans cette infrastructure, les agents sont composés d'un mécanisme de communication, d'une base de règles, d'une base de solutions et d'un moteur d'inférence. La base de règles contient des règles qu'il faut comparer aux événements observés (correspondant à des anomalies), ainsi que les réactions à avoir. Les réactions ou services sont également décrites comme des règles dans la base de solutions. Ainsi, les règles de réaction sont de la forme `<event-Id, reaction-Id>`. Cette infrastructure a été utilisée pour des applications de gestion comptable.

Une autre infrastructure est décrite dans (38). Une architecture d'agent générique est définie et construite en utilisant April++, une extension orientée objet de April (25). L'agent est organisé en unités qui communiquent (en interne à l'agent) grâce à un mécanisme de tableau noir. Les unités sont organisées verticalement, ce qui signifie qu'elles s'exécutent simultanément. L'information locale à l'agent est centralisée sous forme d'une base d'information. L'agent possède également un médiateur de lien, qui est l'interface avec le composant physique, et une tête qui assure les communications avec les autres agents. De nouvelles fonctions peuvent être ajoutées à l'agent en le dotant de nouvelles unités. Ce modèle a été expérimenté pour la gestion de réseaux multiservices. Pour être complet, cet agent possède également un module de communication avec l'utilisateur, un module de gestion des demandes de services, ainsi qu'un module gérant les fautes.

XIII.6.3 – Agents délibératifs

Seulement deux applications peuvent être considérées comme déployant réellement des agents délibératifs. La première est décrite dans (50). Cet article décrit une infrastructure intéressante appliquée au contrôle d'un service VPN (Réseau Privé Virtuel). Les agents sont utilisés pour automatiser la négociation entre le fournisseur et l'utilisateur en cas de changement des paramètres de service ou de demande de réparation de fautes réseau. L'agent de l'utilisateur a une connaissance de la structure logique du VPN et de son utilisation, tandis que l'agent du fournisseur

connaît la réalisation logique et physique de chaque VPN. Les agents ont donc un modèle du monde sur lequel ils agissent. Plus précisément, un agent utilisateur connaît, pour chaque faisceau (ou lien logique), la capacité et l'utilisation de celle-ci ; de son côté l'agent du fournisseur connaît pour chaque faisceau les liens physiques sur lesquels il est mis en œuvre. Quand, par exemple, arrive une faute réseau, les VPN qui s'appuient sur l'élément défectueux sont affectés. L'agent utilisateur va d'abord demander à l'agent fournisseur de réparer la faute. Très souvent, la réparation immédiate n'est pas possible, et une coopération de type négociation s'établit. L'agent utilisateur essaye de trouver des solutions intermédiaires basées sur la connaissance de l'utilisation des faisceaux, et suggère ces solutions à l'agent fournisseur, en modifiant la structure logique du VPN. L'agent fournisseur coordonne toutes ces solutions et fait les changements nécessaires pour atteindre une configuration acceptable. Les deux types d'agents utilisent des raisonnements logiques basés sur leurs croyances (*beliefs*) relatives aux éléments de réseau.

La seconde application utilisant des agents délibératifs est le projet MANIA (*Managing Awareness in Networks using Intelligent Agents*) (31). Une approche orientée agents est utilisée pour gérer la qualité de service individuelle de chaque utilisateur dans un réseau. Les agents utilisent des catégories mentales (croyances, désirs, buts...). La croyance de l'agent exprime sa perception de l'environnement. La croyance est structurée en plusieurs types d'information : l'état temps réel des ressources (i.e. l'imprimante est très sollicitée en ce moment) ; le comportement historique du réseau (i.e. le service d'impression est toujours très sollicité à une heure donnée) ; les états possibles des services (i.e. la charge maximum que peut supporter une ressource). De plus, l'agent détient des connaissances sur l'utilisateur lui-même et ses besoins en termes de qualité de service.

Les désirs de l'agent sont constitués de deux parties. La première correspond aux requêtes que l'agent ne peut satisfaire (i.e. un utilisateur veut faire une visioconférence, mais il n'y a pas de bande passante disponible). La seconde contient les politiques de motivation (i.e. le gestionnaire de réseau veut motiver l'agent pour donner un certain niveau de priorité aux membres d'un projet, car ils ont une contrainte temporelle).

A la réception d'un contexte d'application (qui décrit les besoins d'un utilisateur en termes de ressources et les qualités de service correspondantes), l'agent les traduit en un ensemble de buts. Par exemple, un but peut être de « monitorer activement le service NSF ». Les buts sont alors traduits en intentions, qui prennent en compte l'état du réseau, les moyens pour atteindre les buts et les contraintes possibles.

XIII.6.4 – Les agents hybrides

Comme dans le cas précédent, deux expériences méritent d'être décrites. La première est présentée dans (12). Elle porte sur un système de multi-agents intelligents chargé de gérer le contrôle d'admission dans un réseau ATM. La gestion centralisée d'un réseau ATM est délicate, car les données vieillissent très vite. La ges-

tion locale n'est pas non plus appropriée, à cause d'un manque de visibilité globale du réseau. Les agents hybrides sont, par contre, parfaitement adaptés. Ils savent combiner une gestion temps réel locale grâce à un comportement réactif, tandis que la partie délibérative s'occupe de la coordination entre agents, afin d'atteindre des objectifs globaux. Le projet est encore en développement et les détails de mise en œuvre incomplets.

La deuxième application est, elle, bien définie. Son architecture s'appuie sur le concept de « *vidid agents* » (47), agents incluant une partie réactive et une partie délibérative. La partie réactive n'est pas codée en dur, et son comportement peut être modifié dynamiquement par la partie délibérative (notion de *Reagents*, définis comme cas particuliers de *vidid agents*, sans capacité de planification).

Les *Reagents* sont utilisés pour réaliser des diagnostics sur des systèmes distribués (6). Le réseau est partitionné en domaines physiques, chaque domaine étant doté d'un agent. Ce dernier contient un modèle de connaissance pour son domaine et une information minimale sur les domaines voisins (au moins, l'adresse de leur agent). Ce système multi-agents applique une version distribuée de la technique *Model Based Diagnosis*. Le principe est que lorsqu'un agent détecte une faute (perte de connexion, par exemple), il commence par une analyse effectuée dans son propre domaine géographique. Cette analyse constitue le comportement délibératif de l'agent, puisqu'il raisonne en fonction de son modèle de connaissance du domaine. S'il ne trouve rien d'anormal localement, il informe les agents de domaines voisins. Ceux-ci réalisent alors le même type d'analyse. S'ils découvrent l'élément en cause, ils informent le premier agent. Un agent informe donc toujours ses voisins et transmet les rapports d'anomalie. Il a un comportement réactif, qui ne suppose aucun raisonnement.

XIII.6.5 – Synthèse et discussion

Il est remarquable qu'il existe peu de publications sur les architectures d'agent. Cette rareté a plusieurs raisons. La première est que la plupart des études publiées s'attachent à décrire des scénarios, sans chercher à décrire des systèmes multi-agents (voir, par exemple, les références (24), (27) (36) et (18)). La seconde raison est que pour les infrastructures supportant les deux architectures, les réalisations sont, en général, à l'état embryonnaire (voir, par exemple, (41)). Une autre raison est que le monde des agents mobiles (relativement développé) s'intéresse rarement aux architectures d'agents intelligents.

Il faut noter que les agents réactifs sont particulièrement adaptés pour les opérations de gestion temps réel. Ils sont très efficaces quand il s'agit d'identifier un scénario de fautes connu, dans un réseau peu étendu. Des approches basées sur l'analyse des symptômes sont souvent suffisantes et efficaces.

D'autre part, les agents délibératifs sont, clairement, mieux adaptés à la résolution de problèmes complexes et au raisonnement logique. Ils ont donc besoin de plus de ressources, et ont des temps de réponse plus longs. Il faut donc les réserver à des tâches complexes, sans contrainte temps réel (comme, par exemple, des analyses d'origine de fautes de sécurité dans des grands réseaux).

Il nous semble que l'idée présentée en (12) d'utiliser des agents hybrides qui combinent les avantages des deux architectures est parmi les plus intéressantes. Le concept de *vivid agent* est sans doute la meilleure solution pour mettre en œuvre cette idée. Celle-ci offre, en effet, une grande flexibilité de conception en permettant des modifications dynamiques du comportement de la couche réactive. Les agents peuvent donc s'améliorer dans le temps, grâce aux mécanismes d'apprentissage de la partie délibérative. Cependant, un résultat similaire peut être obtenu avec un système multi-agents hétérogène, contenant à la fois des agents délibératifs et des agents réactifs, les premiers gouvernant le comportement des seconds (voir (30) pour le développement de cette idée).

XIII.7 – Coopération dans les systèmes multi-agents en gestion de réseaux

La coopération est l'une des propriétés les plus importantes des multi-agents (4). Dans le domaine de la gestion de réseaux, la coopération est la base même des systèmes de gestion distribués. Même si le terme coopération (ou collaboration) n'est pas clairement défini, nous décrirons dans cette section les techniques qui s'y rapportent dans le domaine des systèmes de gestion orientés agents. Nous essaierons d'aller des solutions les plus simples aux solutions les plus élaborées.

XIII.7.1 – Coordination sans communication

Comme décrit dans la section XIII.5, le système multi-agents décrit dans (1) s'appuie sur des agents qui ne communiquent pas directement entre eux. Un agent n'est conscient de la présence d'autres agents qu'en observant l'état du système. Chaque fois qu'un agent visite un nœud, il y laisse des informations telles que son identité, son âge et sa date de visite. La lecture de cette information dans chaque nœud traversé informe un agent du nombre d'agents présents dans le réseau. Si ce nombre est trop grand, l'agent le plus jeune met un terme à son action. Il en résulte que, dans certains nœuds déterminés, un processus statique s'exécute en permanence pour vérifier si les agents parents sont toujours vivants. Si un laps de temps s'écoule depuis la visite d'un agent, le processus statique déduit que l'agent est tué, et le régénère.

De plus, quand un agent de charge est créé dans un nœud, il génère un enregistrement avec son identité et sa date de naissance. Après avoir achevé sa mission de mise à jour des tables de routage, il retourne à son nœud d'origine, s'enregistre et se détruit. Ainsi, le parent, lors d'une visite suivante, peut savoir si l'agent de charge est encore vivant ou non. Il peut également savoir s'il a été détruit sans avoir achevé sa mission.

L'absence de communication directe entre agents permet d'éviter un certain nombre de problèmes de cohérence. D'abord, il n'y aura pas d'envoi à un agent qui

est détruit. Ensuite, les agents partagent la même information, sans besoin de mécanismes pour préserver la cohérence. De plus, les agents peuvent coordonner leurs tâches d'une manière très simple. Ces particularités dotent ce système multiagents d'un haut niveau de robustesse, propriété rarement considérée dans d'autres applications.

XIII.7.2 – Une technique primitive de coopération

Dans la section XIII.6, nous avons présenté un système à base d'agents mettant en œuvre une version distribuée du *Model-Base Diagnosis* (6). En cas de faute, l'agent réalise un diagnostic local, et s'il ne trouve aucun élément fautif, il demande à l'agent voisin de faire le diagnostic sur son domaine. On peut considérer ce processus comme une méthode de coopération très simple, et ce pour deux raisons. La première est que le mécanisme de communication est lui-même très primitif. La seconde est que ce mode de coopération est codé en dur, et les agents n'ont aucune décision intelligente à prendre. Cependant, si on se réfère à (4), les agents coopèrent puisqu'ils participent à l'atteinte d'un objectif (même s'il est implicite), qui est d'identifier un élément fautif dans un réseau.

XIII.7.3 – Négociation entre agents auto-intéressés

Les agents auto-intéressés (*self-interested*) n'ont besoin d'aucune connaissance des plans ou objectifs des autres agents. Ils travaillent pour atteindre leurs propres objectifs de la meilleure façon possible, sans tenir compte des autres. Cela ne les empêche pas de faire appel aux services d'autres agents. Une manière naturelle est donc de faire appel à la négociation. En déléguant des tâches à d'autres agents, l'agent cherche ici à maximiser son propre profit. On est donc, dans un certain sens, proche de certains modèles économiques (48).

Les agents auto-intéressés sont particulièrement adaptés à certaines activités d'administration de réseaux, particulièrement pour la gestion de services. Beaucoup de publications dans ce domaine se réfèrent à ce type d'agents. Dans (24), Thomas Magedanz suggère que ces agents sont adaptés aux problèmes d'approvisionnement dans le marché ouvert des services de télécommunications. Un bon exemple est le schéma d'application du FIPA présenté dans (18). L'application définit trois types d'agents : l'agent personnel (PA), l'agent du fournisseur de services (SPA) et l'agent du fournisseur de réseaux (NPA). Ce dernier est responsable de la fourniture des ressources et éléments de réseau nécessaires à la réalisation du service. A partir de ces ressources, le SPA est responsable de la fourniture des services, avec la qualité de service requise. Le PA joue le rôle d'un assistant électronique personnel (PDA) (23) qui aide l'utilisateur pour la définition de ses désirs et les besoins de l'application, en fonction de ses préférences. Il doit négocier les caractéristiques avec différents SPA, afin d'obtenir le meilleur service, avec la meilleure qualité de service et au meilleur coût. Au moment de l'établissement de la communication, le PA doit adapter les ressources locales et les configurer en fonction des caractéristiques de la communication.

Le SPA doit s'informer des besoins de l'utilisateur et identifier les services et leurs paramètres. Ensuite, il négocie avec les NPA pour sélectionner le meilleur réseau, toujours en termes de qualité de service et de coût. Il doit trouver le meilleur compromis entre le fournisseur de réseaux et l'utilisateur.

Enfin, le NPA écoute les demandes des SPA pour les traduire en besoins réseaux, en termes de bande passante, gigue, ... Il doit également négocier avec d'autres NPA dans le cas où la communication implique plusieurs opérateurs de réseaux.

D'autres travaux sur le sujet utilisent des approches similaires. Mike Ruzzo et Ian A. Utting (36) ont également introduit des *User Agents* (UA) pour la définition de services personnalisés, caractérisés par la notion de politique. L'UA doit faire le meilleur choix en négociant avec l'agent du fournisseur, et atteindre un agrément satisfaisant au mieux la politique de l'utilisateur et celle du fournisseur, en termes de préférences et de contraintes. Afin de supporter le nomadisme de l'utilisateur, des agents terminaux sont également introduits. Les UA mobiles négocient avec les agents terminaux pour obtenir l'autorisation d'accès aux points d'accès réseau et établir les communications.

D'autres scénarios suggèrent d'intégrer de nouveaux services dans les agents. Puisque les services sont personnalisés par les PA, (16) propose de nouveaux services avec un environnement ad hoc. Les agents de service détiennent les spécifications des réseaux, et les utilisent pour construire des services. De la même manière, (27) présente un scénario où des agents mobiles sont envoyés dans un environnement sur les machines des services, et ils peuvent offrir de nouveaux services en utilisant les services présents et/ou d'autres agents mobiles. Dans les deux scénarios, les agents, qui procurent de nouveaux services doivent négocier avec les différents agents afin de trouver ceux qui offrent les meilleurs supports pour ces services

Dans (12), une solution de négociation à base d'enchères est suggérée, pour l'affectation des ressources dans les réseaux ATM. Il s'agit d'un protocole simple mais efficace, qui permet de prendre des décisions très vite, entre les utilisateurs qui proposent les plus hauts prix et les fournisseurs qui offrent les plus bas.

Un principe de négociation est également adopté dans (41), pour la gestion de réseaux ATM. Les agents sont organisés en hiérarchies d'autorités. Chaque autorité est responsable d'un certain nombre de ressources, et délivre des « indices de performance ». Ainsi, quand une autorité (en particulier un agent de service dans l'autorité) reçoit une demande usager, elle réserve les ressources nécessaires et décide de quelle autre autorité est la mieux placée pour router la requête. Cette décision est prise en fonction des indices de performance.

XIII.7.4 – Coopération à base de délégation

Par opposition au paradigme de MdB présenté dans la section XIII.4, qui n'agit que par des scripts, nous nous intéressons ici à la délégation entre agents intelligents. Il s'agit d'un niveau plus élevé de délégation car les agents délèguent des objectifs et des motivations au lieu d'opérations de bas niveau.

L'infrastructure d'agent décrite dans (42) (voir section XIII.6.2) utilise des politiques exprimées sous forme de règles logiques. Ces politiques sont traduites en politiques de plus bas niveau et mémorisées dans une base de règles.

Le projet MANIA (voir section XIII.6.3) utilise également une délégation avec un haut niveau d'objectifs. Un agent peut faillir dans l'atteinte d'un objectif s'il lui manque les ressources nécessaires. Il peut alors déléguer l'objectif à un autre agent. Il faut bien noter que seuls les objectifs sont délégués, pas les autres catégories mentales (donc pas les intentions). L'agent récepteur peut avoir un comportement différent de celui qu'aurait pu avoir l'agent demandeur, du fait de son état propre.

XIII.7.5 – Synthèse et discussion

Bien qu'il soit communément admis que la coopération soit très bénéfique aux architectures de gestion de réseaux distribuée, elle n'est que peu exploitée dans les approches à base d'agents. La négociation est une technique particulièrement utile pour les problèmes de construction de services et pour leur gestion. Un modèle de type marché est adapté pour mettre en œuvre des agents qui recherchent la meilleure qualité au meilleur prix. Ils doivent donc négocier pour maximiser le profit.

Les mécanismes de communication sont assez variés suivant les approches. Tandis que dans (6), le protocole de communication n'est pas explicité (simplement parce que le besoin se limite à un simple échange de messages), (36) a fait une distinction entre la syntaxe et la sémantique des messages. Dans (50), deux types de messages (INFORM et REQUEST) sont suffisants, alors que dans (38), les messages sont identifiés par des schémas. La plupart des autres travaux ont opté pour KQML, comme (18) et (42). Dans le projet MANIA (31), les agents utilisent KQML pour la syntaxe, et un langage de type KIF pour échanger des objectifs. Enfin, dans (41), une version étendue de KQML est utilisée. Les primitives ajoutées sont PROPOSE, COUNTER-PROPOSE, ACCEPT, REJECT, PROBLEM et REJECT. Elles sont introduites pour gérer la négociation.

En ce qui concerne la résolution distribuée de problèmes, aucune solution à base d'agents n'existe, qui présenterait une planification globale et une méthode collaborative pour atteindre des objectifs partagés. Si l'on se réfère à (4), un système multiagents coopératif pourrait provenir soit d'agents qui donneraient une priorité à leurs objectifs internes, soit d'agents qui essaieraient d'abord d'atteindre les objectifs globaux. La majorité des systèmes d'agents existants sont du premier type.

XIII.8 – Un agent d'interface

Une application d'agent d'interface pour la supervision d'un réseau est présentée dans (5). L'agent doit traiter un grand nombre d'alarmes et d'événements, en les filtrant et en attachant chaque notification à son contexte.

La connaissance est acquise au fil de l'eau. Utilisant le modèle de chronique (voir (10)), l'agent est capable de faire des raisonnements temporels sur les événements et d'automatiser des tâches de gestion. Une chronique est un ensemble d'événements corrélés, et une action doit être prise quand une séquence est reconnue. Une chronique utilise deux types de formules : la formule « *hold* », qui exprime qu'une valeur est valide sur un intervalle de temps, et la formule « *event* », qui exprime un certain changement de valeur d'attribut.

Un « système de reconnaissance de chroniques » identifie les chroniques et associe à chacune les actions à entreprendre. Un système d'apprentissage évalue les chroniques identifiées, en les comparant à celles déjà connues. Dans un premier temps, les chroniques identifiées sont stockées dans une base des chroniques « non confirmées ». Suite à l'atteinte d'un seuil (ou si l'administrateur le décide), une chronique est transférée dans la base des chroniques confirmées.

Dans cet exemple, l'agent apprend soit par observation, soit en recevant des instructions. Il n'y a donc pas de coopération, mais cet aspect est envisagé dans une extension (voir la conclusion (5)).

XIII.9 – Synthèse générale

Nous synthétisons dans cette section notre vision des approches orientées agents. Nous mettons, en particulier, l'accent sur les propriétés qui restent à étudier plus en profondeur pour l'application de ces techniques en administration de réseaux. Nous donnons, enfin, notre avis sur les tendances qui caractérisent l'introduction des agents dans le monde de la gestion de réseaux.

XIII.9.1 – La vision orientée agents

XIII.9.1.1 – la proactivité

Comme indiqué dans la section XIII.3.2, la proactivité est la capacité d'un agent à anticiper des changements dans un environnement et à agir pour éviter que certaines difficultés ne surviennent. Dans le contexte de la gestion de réseaux, la proactivité consiste à prévoir certaines défaillances (en termes de fautes ou de dégradation de la qualité de service) et de mettre en place des mesures correctives, avant même que les incidents ne se produisent.

La proactivité a été très peu étudiée dans le contexte de la gestion de réseaux. Seules deux approches la mentionnent :

La première (42) qui précise, en parlant de la proactivité que « des politiques peuvent être mises en œuvre par des agents intelligents pour diviser un problème en plusieurs sous-problèmes, et résoudre chacun de ces sous-problèmes... ». Clairement, cette affirmation n'a rien à voir avec la proactivité.

La seconde approche est celle du projet MANIA (33). Le comportement proactif peut être réalisé de deux manières. Dans la première, l'agent étudie les contextes d'application envoyés par les utilisateurs pour déclarer leurs besoins en terme de QoS. Il en déduit les ressources réseaux qui seront sollicitées dans un proche avenir. La seconde manière consiste à apprendre l'évolution du comportement du réseau au fil du temps. Par exemple, l'agent pourrait constater que le serveur NFS est surchargé tous les jours de 9 à 10 heures. Il pourrait donc décider de mettre en route un second serveur juste avant l'heure de surcharge (donc avant le début de la congestion), ou interdire les accès à des utilisateurs de faible priorité.

XIII.9.1.2 – L'apprentissage

L'apprentissage de la connaissance et de l'expertise en administration de réseaux n'est pas de même nature que l'apprentissage du comportement du réseau. Dans le premier cas, il s'agit d'acquérir de nouvelles capacités pour gérer le réseau, par exemple savoir prendre en compte de nouveaux types de problèmes. Le seul exemple de ce type est l'interface d'agents de (5) cité dans la section XIII.8.

Dans le second cas, il s'agit pour l'agent d'améliorer sa connaissance sur le réseau qu'il gère, en vue d'appliquer la même expertise, mais d'une manière plus efficace. Dans la section précédente, nous avons vu quelle sorte d'apprentissage était nécessaire pour réaliser des actions proactives. Dans (1), les agents sont considérés comme acquérant des connaissances sur l'état du réseau afin de déduire la charge moyenne d'utilisation d'un nœud.

XIII.9.1.3 – Robustesse et dégradation progressive

Ces deux propriétés concernent le système multi-agents pris dans son ensemble. La dégradation progressive implique que le système multi-agents ne s'effondre pas brutalement en cas d'une panne d'un de ses composants (30), mais que ses performances diminuent en douceur. La robustesse est la capacité à résister, si des fautes dans l'environnement se produisent (panne réseau ou arrêt de système).

La seule application où ces deux aspects sont considérés est celle de (1). Le mécanisme qui permet de détecter les agents défaillants et de les remplacer a déjà été décrit dans la section XIII.7.1. Rappelons que si un agent parent tombe en panne, cette panne sera détectée et l'agent sera remplacé par un processus statique, dont le but est de maintenir un nombre minimum d'agents opérationnels. En parallèle, les autres agents poursuivent leurs tâches sans que le fonctionnement du système soit interrompu (dégradation progressive). Si un agent de charge tombe, l'incident sera également repéré, et un nouvel agent sera créé. Le système multi-agents a donc la capacité de poursuivre sa mission en cas de panne (robustesse).

XIII.9.2 – Tendances en gestion de réseaux

Le nombre d'acteurs de la gestion de réseaux (constructeurs, opérateurs, fabricants de logiciels...) qui s'intéressent aux agents intelligents est en constante progression. Beaucoup de fournisseurs d'équipements essayent d'incorporer des sys-

tèmes d'AI dans des appareils tels des routeurs ou des commutateurs. L'idée est de doter ces équipements d'un certain niveau d'intelligence, au profit d'une autonomie de gestion et de propriétés d'auto réparation. Par ailleurs, l'intelligence profitera au réseau, par une réduction du trafic et une automatisation du processus de gestion.

D'autres tendances s'intéressent au niveau service. Ce sont principalement les opérateurs qui sont concernés, quand ils cherchent un meilleur niveau de flexibilité dans l'établissement de nouveaux services, et des moyens pour contrôler la qualité du service fourni (et de plus en plus contractualisé avec les utilisateurs).

Les plates-formes de gestion de réseaux introduisent également des technologies agent. Quelques constructeurs les ont annoncées, mais sans toujours préciser les modes d'intégration dans leurs produits. Plusieurs solutions sont envisagées : baser toute la plate-forme sur des agents intelligents ; introduire une infrastructure qui permette à l'utilisateur de définir ses propres agents ; utiliser des agents intelligents préprogrammés. L'objectif poursuivi est de rendre le système plus réactif (ou proactif), plus flexible et de répondre au problème d'évolution de taille.

Dans la pratique, on est encore loin de l'achèvement des recherches nécessaires à l'intégration des architectures d'agents intelligents dans les plates-formes de gestion de réseaux. Un grand nombre de techniques, d'environnements et de langages sont disponibles pour définir des systèmes d'agents et leur permettre de coopérer. Aucune de ces technologies ne fait aujourd'hui l'unanimité. Le besoin, par contre, reste important et ne saurait décroître.

Beaucoup d'efforts portent sur les solutions à base de Java. Elles reposent sur les promesses offertes par la notion de machine virtuelle, et des composants Java qui ont été annoncés et qui devraient bientôt sortir.

XIII.10 – Conclusion

Cette étude a présenté un état de l'art dans le domaine des approches reposant sur les agents intelligents pour la gestion de réseaux. Le lecteur doit rester conscient que ce domaine est considéré par beaucoup d'industriels comme confidentiel, et que, par conséquent, toutes les recherches n'ont pas été publiées.

Les articles qui ont été analysés peuvent être classés en trois catégories. La première décrit des scénarios où les agents sont utilisés pour aider à résoudre certains problèmes de gestion de réseaux. Ils sont en général avares de renseignements sur l'architecture agents utilisée. La seconde classe s'intéresse aux solutions à base d'agents dédiées à un domaine spécifique de la gestion de réseaux. Elles sont en général très détaillées mais ne sont pas utilisables dans le cadre d'une approche intégrée de la gestion de réseaux. La dernière classe décrit des architectures à base d'agents, chacune illustrée d'un exemple d'application liée à une fonction spécifique de gestion de réseaux. Les domaines de gestion les plus couverts sont les fautes, la configuration et les performances. La gestion financière et la sécurité ne sont pratiquement pas étudiées.

L'utilisation des techniques d'agents intelligents apporte aux applications présentées un haut degré d'automatisation et de flexibilité. Le résultat est de libérer l'opérateur humain de certaines tâches de bas niveau et routinières. Celui-ci peut interagir avec l'agent d'une manière abstraite et de haut niveau, en s'exprimant sous forme d'objectifs, de motivations et de politiques. De plus, les agents deviennent conscients de l'état fin du système et de son utilisation. Des actions correctives peuvent être entreprises d'une manière appropriée et dans un laps de temps très court, et des mesures proactives peuvent être prises.

Pour ces raisons, de nombreux organismes ont commencé à investir lourdement dans ces techniques, et participent à des actions de normalisation pour promouvoir leurs travaux⁽¹⁾.

Références

- (1) S. Appleby et S. Steward. « Software agents for control. », dans P. Cochrane et PJT Meathley, editors, *Modelling Future Telecommunication Systems*. Chapman & Hall, 1994.
- (2) Mihai Barbuceanu et Mark S. Fox. « The design of a coordination language for multi-agent systems. », *Intelligent Agents III. Agent Theories, Architectures, and Languages*, pages 341-355. Springer, 1996.
- (3) Mario Baldi, Silvano Gai, et Gian Pietro Picoo. « Exploiting code mobility in decentralized and flexible network management. », *Proceedings of the First International Workshop on Mobile Agents*, Berlin, Germany, April 1997.
Disponible sur <http://www.polito.it/~picco/papers/ma97.ps.gz>.
- (4) J. E. Doran, S. Franklin, N. R. Jennings, et T. J. Norman. « On cooperation in multi-agent systems. », *The Knowledge Engineering Review*, 12(3), 1997.
Disponible sur <http://www.elec.qmw.ac.uk/dai/pubs/fomas.html>.
- (5) Babak Esfandiari, Gilles Deflandre, et Joël Quinqueton. « An interface agent for network supervision. » 1996.
- (6) Peter Fröhlich, Iara de Almeida Móra, Wolfgang Nejdl, et Michael Schroeder. « Diagnostic agents for distributed systems. », *Proceedings of ModelAge97*, Sienna, Italy, January 1997.
Disponible sur <http://www.kbs.uni-hannover.de/paper/96/ma1.ps>.
- (7) Stan Franklin et Art Graesser. « Is it an agent, or just a program?: A taxonomy for autonomous agents. », *Intelligent Agents III. Agent Theories, Architectures, and Languages*. Springer, 1996.

(1) Le contenu de ce chapitre utilise les résultats d'une collaboration d'Eurécom avec Swisscom qui en a assuré un financement partiel. Les auteurs souhaitent également remercier Raul OLIVEIRA pour ses précieux conseils.

- (8) Foundation for Intelligent Physical Agents. Application design test: Network management and provisioning. <http://drogo.csel.it/fipa/spec>, June 1997.
- (9) Garry Grimes et Brian P. Alley. « Intelligent agents for network fault diagnosis and testing. », *Integrated Network Management V : Integrated Management in a virtual World*, pages 232-244, San Diego, California, USA, May 1997. IFIP, Chapman & Hall.
- (10) Malik Ghallab. « Past and future chronicles for supervision and planning. », Jean Paul Haton, editor, *Proceedings of the 14th Int. Avignon Conference*, pages 23-34, Paris, June 1994. EC2 and AFIA.
- (11) I. G. Ghetie. « Networks and Systems Management. Platforms Analysis and Evaluation. », Kluwer Academic Publishers, 1997.
- (12) M. A. Gibney et N. R. Jennings. « Market based multi-agent systems for atm network management. », *Proc. 4th Communications Networks Symposium*, Manchester, UK, 1997.
Disponible sur <http://www.elec.qmw.ac.uk/dai/projects/agentCAC/>.
- (13) Germán Goldszmidt. « Distributed system management via elastic servers. », *IEEE First International Workshop on Systems Management*, pages 31-35, Los Angeles, California, April 1993.
- (14) Christine Gressley. « Network management resources - reviews of network management systems, » July 1996.
Disponible sur <http://tampico.cso.uiuc.edu/~gressley/netmgmt/reviews/> .
- (15) Germán Goldszmidt et Yechiam Yemini. « Distributed Management by Delegation. », *The 15th International Conference on Distributed Computing Systems*. IEEE Computer Society, June 1995.
- (16) Gisli Hjálmtsson et A. Jain. « An agent-based approach to service management - towards service independent network architecture. », *Integrated Network Management V : integrated management in a virtual world*, pages 715-729, San Diego, California, USA, May 1997. IFIP, Chapman & Hall.
- (17) Christian Huitema. « Et Dieu créa l'Internet. » Eyrolles, 1996.
- (18) IFIP. *Integrated Network Management V: integrated management in a virtual world*, San Diego, California, USA, May 1997. Chapman & Hall.
- (19) Richard Kooijman. « Divide and conquer in network management using event-driven network area agents. », *ASCI Conference*, Heijderbos, Nederland, May 1995.
Disponible sur <http://netman.cit.buffalo.edu/Doc/Papers/koo9505.ps>.
- (20) P. Kalyanasundaram, A.S. Sethi, C. Sherwin, et D. Zhu. « A spreadsheet-based scripting environment for snmp. », *Fifth IFIP/IEEE International Symposium on Integrated Network Management IM'97*, volume 5, pages 752-765, San-Diego, California, USA, May 1997. Chapman & Hall.

- (21) S. Labidi et W. Lejouad. « De l'intelligence artificielle distribuée aux systèmes multi-agents ». , Technical report, INRIA Sophia Antipolis, INRIA Sophia Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France, 1993.
- (22) E.C. Lupu et M. Sloman. « Towards a role-based framework for distributed systems management. », *Journal of Network and Systems Management*, 5(1), 1997.
- (23) Pettie Maes. « Agents that reduce work and information overload. », *Communications of the ACM*, 37(7):31-40, 1994.
- (24) Thomas Magedanz. « On the impacts of intelligent agents concepts on future telecommunication environments. », *Third International Conference on Intelligence in Broadband Services and Networks*, Crete, Greece, October 1995.
- (25) F. G. McCabe et K. L. Clark. « April: Agent process interaction language. », Technical report, Dept. of Computing, London, UK, November 1994. Disponible sur <http://www-lp.doc.ic.ac.uk/~klc/>.
- (26) Maria-Athina Mountzia. « Intelligent agents in integrated network and systems management, » 1996.
- (27) T. Magedanz, K. Rothermel, et S. Krause. « Intelligent agents: An emerging technology for next generation telecommunications ? », *INFOCOM' 96*, pages 464-472, USA, March 24-28 1996. IEEE.
- (28) Damian Marriot et Morris Sloman. « Implementation of a management agent for interpreting obligation policies. », *IEEE/IFIP 7th International Workshop on Distributed Systems and Operations Management - DSOM'96*, L'Aquila, Italy, October 1996.
- (29) Jörg P. Muller. « The Design of Intelligent Agents - A Layered Approach. », *LNAI State-of-the-Art Survey*. Springer, Berlin, Germany, 1996.
- (30) Hyacinth S. Nwana. « Software agents: An overview. », *Knowledge Engineering Review*, 11(3):205-244, October/November 1996. Disponible sur <http://www.cs.umbc.edu/agents/introduction/ao/>.
- (31) Raul Oliveira et Jacques Labetoulle. « Intelligent agents: a way to reduce the gap between applications and networks. », J. D. Decotignie, editor, *Proceedings of the First IEEE International Workshop on Factory Communications Systems - WFCS'95*, pages 81-90, Leysin, Switzerland, October 4-6 1995. Disponible sur <http://www.eurecom.fr/~oliveira/wfcs/wfcs.ps.gz>.
- (32) OMG. « Mobile agent facility specification. », Technical report, Crystaliz, Inc., General Magic, Inc., GMD FOKUS, International Business Machine Corporation, June 1997. OMG TC Document.

- (33) Raul Oliveira, Dominique Sidou, et Jacques Labetoulle. « Customized network management based on applications requirements. », Proceedings of the First IEEE International Workshop on Enterprise Networking - ENW '96, Dallas, Texas, USA, June 27 1996.
- (34) Nick Parkyn. « Architecture for intelligent (smart) agents, » June 1995. Disponible sur <http://www.citr.uq.oz.au/>.
- (35) Aiko Pras. « Network Management Architecture. », PhD thesis, Universiteit Twente, Department of Computer Science, February 1995. Disponible sur <http://wwwsnmp.cs.utwente.nl/~pras/thesis.html>.
- (36) Mike Rizzo et Ian A. Utting. « An agent-based model for the provision of advanced telecommunications services. », Proceedings of TINA '95, Melbourne, Australia, 1995.
- (37) Akhil Sahai, Stéphane Billiard, et Christine Morin. « Astrolog: A distributed and dynamic environment for network and system management. » , Proceedings of the 1st European Information Infrastructure User Conference, Germany, February 1997. Disponible sur <http://www.irisa.fr/solidor/doc/pub97.html>.
- (38) Nikolaos Skarmaeas et Keith L. Clark. « Process oriented programming for agent-based network management. », ECAI96 Workshop on Intelligent Agents for Telecommunication Applications (IATA96), Budapest, Hungary, August 12 - 16 1996.
- (39) Motohiro Suzuki, Yoshiaki Kiriha, et Shoichiro Nakai. « Delegation agents: Design and implementation. », Integrated Network Management V : integrated management in a virtual world, volume 5, pages 742-751, San Diego, California, USA, May 1997. IFIP, Chapman & Hall.
- (40) FTP Software. « Ftp software agent technology. » Technical report, FTP Software, <http://www.ftp.com/product/whitepapers/4agent.htm>, 1997.
- (41) Fergal Somers. « Hybrid: Unifying centralised and distributed management for large high-speed networks. », Networks Operation and Maintenance Symposium (NOMS96), Kyoto, 1996 1996. Disponible sur <http://www.broadcom.ie/~fs>.
- (42) P. Steenekamp et J. Roos. « Implementation of distributed systems management policies: A framework for the application of intelligent agent technology. », 2nd International Workshop on Systems Management, Toronto, Ontario, Canada, June 1996. IEEE.
- (43) William Stallings. « SNMP, SNMPv2 and RMON, Practical Network Management. », Addison-Wesley, USA, 1996.
- (44) Douglas W. Stevenson. « Network management: What it is, what it isn't., » 1995. Disponible sur <http://netman.cit.buffalo.edu/Doc/DStevenson>.
- (45) K. Terplan. « Communication Networks Management. », Prentice Hall, 1992.

- (46) Markus Trommer et Robert Konopka. « Distributed network management with dynamic rule-based managing agents. », *Integrated Network Management V : integrated management in a virtual world*, pages 730-741, San Diego, California, USA, May 1997, IFIP, Chapman & Hall.
- (47) Gerd Wagner. « Vivid agents - how they deliberate, how they react, how they are verified. », W. Van de Velde and J.W. Perram, editors, *Agents Breaking Away, Proc. of MAAMAW'96*, 1996.
Disponible sur <http://www.informatik.uni-leipzig.de/~gwagner/>.
- (48) M. P. Wellman. « A market oriented programming environment and its application to distributed multicommodity flow problems. », *Journal of Artificial Intelligence Research*, 1:1-23, 1993.
- (49) Michael Wooldridge et Nicholas R. Jennings. « Intelligent Agents: Theory and Practice. », *Knowledge Engineering Review*, 10(2):115-152, 1995.
- (50) Robert Weihmayer et Ming Tan. « Modelling cooperative agents for customer network control using planning and agent-oriented programming. » 1992.
- (51) Yechiam Yemini, Germán Goldszmidt, et Shaula Yemini. « Network Management by Delegation. », *The Second International Symposium on Integrated Network Management*, pages 95-107, Washington, DC, April 1991.
- (52) Simon Znaty, Michel Lion et Jean-Pierre Hubaux. « Deal: A delegated agent language for developing network management functions. », *First International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology*, 1996.