UN MODELE D'AGENT POUR LA PLANIFICATION DISTRIBUEE

Slim BOUSSETTA, Daniel COHEN, Pavlos MORAÏTIS
{boussetta, cohendan, moraitis}@lamsade.dauphine.fr

LAMSADE
Université PARIS-DAUPHINE
Place du Maréchal De Lattre de Tassigny
75775 PARIS CEDEX 16
Tél: (1) 44 05 46 53

RESUME

Dans cet article, nous présentons un modèle d'agent générique permettant d'assurer des fonctions de planification distribuée dans un contexte multi-agent coopératif. Un concept unique, celui du plan, est utilisé dans le raisonnement de l'agent ainsi que dans ses interactions avec les autres agents du système. Un plan modélise donc les connaissances opératoires (raisonnement), les méta-connaissances (contrôle du raisonnement) et sert de support aux échanges nécessaires à la résolution distribuée d'un problème. Le formalisme de représentation de plans retenu est celui des réseaux de Petri récursifs. Un réseau de Petri ordinaire est composé d'un ensemble de transitions et d'un ensemble de places. Les transitions correspondent aux actions à entreprendre et les places aux différents états du système modélisé. Les réseaux de Petri récursifs permettent de distinguer des transitions raffinables en un nouveau plan, et des transitions élémentaires correspondant aux actions à exécuter. Notre modèle d'agent est composé de quatre modules essentiels assurant des fonctionnalités différentes : le module de planification, le module de manipulation de plans, le module cognitif et le module de communication. Le module de planification choisit, après évaluation, le ou les plans à exécuter ou à déléguer. Le module de manipulation de plans propose au module de planification des primitives de manipulation de plans et suit leur exécution en décidant la parallélisation éventuelle de morceaux de plans. Le module cognitif contient l'expertise de l'agent, les données du domaine et exécute les actions dans les plans. Le module de communication assure la communication inter-agents par envoi de plans. Les différents types de messages nécessaires à la distribution de la planification sont codés dans les plans et interprétés de manière "intelligente" par l'agent récepteur. Les quatre modules de l'agent fonctionnent en parallèle. Notre modèle d'agent est appliqué à la réalisation d'un système multi-agent cognitif de supervision du réseau téléphonique commuté.

MOTS-CLES

Système multi-agent, planification distribuée, réseaux de Petri récursifs, diagnostic réparti

REMERCIEMENTS

Ce travail a été réalisé dans le cadre du contat CNET-LAMSADE CTI 904-15-4371-123. Nous tenons à remercier en particulier François Le Page et Jean-Philippe Coumes (CNET Paris) et Gilles Deflandre (CNET Lannion) pour leur collaboration.

1. INTRODUCTION

Durant de nombreuses années, la recherche en planification s'est surtout intéressée aux environnements mono-agent. Dans ce type d'approche, l'agent évolue dans un domaine statique où il cherche principalement à rassembler des informations, à planifier et à exécuter ses plans en vue d'atteindre ses objectifs (Chaib-draa, 93). Toutefois, cette approche s'est avérée insuffisante dans la mesure où dans le monde réel, la présence de plusieurs agents est inévitable. En effet, dans la plupart des applications réelles chaque agent évolue dans un environnement multi-agent, dans lequel il doit planifier ses activités en tenant compte des interactions avec d'autres agents qui peuvent soit l'aider soit le gêner (Laruelle, 94).

Dans cet article, nous nous intéressons à un aspect particulier de la résolution distribuée de problèmes qui est la planification multi-agent distribuée. Nous présentons un modèle d'agent générique permettant d'assurer des fonctions de planification distribuée dans un contexte multi-agent coopératif. Un concept unique, celui du plan, est utilisé dans le raisonnement de l'agent ainsi que dans ses interactions avec les autres agents du système. Un plan modélise donc les connaissances opératoires (raisonnement), les méta-connaissances (contrôle du raisonnement) et sert de support aux échanges nécessaires à la résolution distribuée d'un problème. Le formalisme de représentation de plans retenu est celui des réseaux de Petri récursifs.

Dans la première section, nous présentons certaines approches de planification distribuée parmi les plus représentatives. Nous introduisons ensuite l'approche de planification distribuée par réseaux de Petri récursifs. Elle est utilisée par le modèle d'agent présenté dans la section suivante. Nous donnons enfin un exemple de fonctionnement de l'agent à travers une application de surveillance du réseau téléphonique.

2. PLANIFICATION MULTI-AGENT DISTRIBUEE

Dans l'approche de planification multi-agent, les agents élaborent un plan multi-agent qui spécifie leurs actions futures et leurs interactions. La planification multi-agent est un moyen d'arriver à une meilleure coordination en orientant le comportement des agents vers des objectifs communs. La génération de plans exige un raisonnement sur la manière dont les actions des différents agents peuvent interférer et exige par conséquent des représentations d'actions parallèles. La synchronisation des plans peut être effectuée à différents points de la résolution d'un problème. La planification multi-agent peut être effectuée par un ou plusieurs agents. Ainsi dans une planification multi-agent centralisée, un seul agent génère des plans pour des agents multiples. Dans une planification distribuée, les activités de planification sont réparties entre les agents. Dans le paragraphe suivant nous présentons certaines approches parmi les plus représentatives dans la planification distribuée.

2.1. Approches classiques

La Planification Partielle Globale (Partial Global Planning, PGP) (Durfee, 89), est une approche flexible de coordination qui n'assure aucune distribution particulière des sous-problèmes, de l'expertise ou d'autres ressources. Elle laisse au contraire les agents coordonner leurs actions et interactions entre groupes d'agents et la façon dont les activités locales sont affectées. Dans une planification globale partielle nous distinguons trois types de plans: des plans locaux, des plans-agent (node-plans) et des plans partiels globaux (partial global plans, PGPs) qui

s'appellent ainsi car ils spécifient comment différentes parties du plan du réseau réalisent des objectifs plus globaux. Un PGP synthétise le travail coordonné de plusieurs agents vers un objectif global. Il constitue donc une structure représentant l'activité de coordination entre agents en terme de buts, actions, interactions et relations. La Planification Partielle Globale intègre des concepts organisationnels à travers deux types d'organisation. La première spécifie les règles de résolution de problèmes à long terme et les responsabilités des agents (organisation niveau-domaine). La seconde, appelée organisation méta-niveau, fournit aux agents une structure pour décider comment résoudre les problèmes de coordination. Chaque agent a un planificateur partiel global (PGPlanner), qui bâtit un plan-agent à partir de chaque plan local. Guidés par l'organisation méta-niveau, les agents échangent PGPs et plans-agent afin qu'un ou plusieurs d'entre eux puisse développer des PGPs plus englobants.

Koo (Koo, 88) s'est intéressé à la synchronisation des activités fondées sur les plans. La synchronisation est atteinte d'une manière distribuée par communication entre les agents. La planification se fonde sur des plans de type graphe PERT. Un plan est un réseau d'états, dans lequel un état est un ensemble de prédicats qui décrit le monde à un instant donné. Un arc est l'action qui change la valeur de certains prédicats dans un état donné. Un plan possède les attributs suivants: un état initial, un état final, une durée qui est la somme des durées d'un chemin. Les opérateurs (actions) sont représentés par une structure contenant: un nom, un ensemble de pré-conditions, un ensemble de post-conditions et une durée. Les états sont représentés comme ayant des prédécesseurs, des successeurs, temps au plus tôt, temps au plus tard et un statut. Koo considère seulement un ensemble restreint d'interactions possible entre plans d'agents autonomes. Les seules interactions, que Koo considère sont fondées sur des contraintes de précédence. Un agent sollicité par un autre en vue d'exécuter une nouvelle tâche doit être capable de savoir s'il peut rajouter une nouvelle action dans son plan. La coordination repose donc sur l'insertion de nouvelles tâches dans un graphe PERT.

Dans l'approche von Martial (von Martial, 92), le résultat du processus de coordination est une réconciliation des plans individuels qui respectent les relations négatives ou positives des plans. Les relations entre plans sont dites positives si un bénéfice peut être tiré de leur combinaison et si elles permettent la production d'un plan global plus efficace. Les relations sont dites négatives lorsqu'elles sont susceptibles d'empêcher l'un des deux plans ou les deux d'être exécutés et peuvent provoquer des conflits entre actions. Ainsi chaque agent est obligé d'adapter son plan en raison même de l'existence d'autres agents et de leurs plans. Le processus de coordination est une combinaison de planification et d'exécution. Du fait des interactions asynchrones entre agents, le but n'est pas d'obtenir un ensemble optimal de plans coordonnés mais d'aider le plus possible les agents à améliorer leurs plans. Les agents peuvent avoir plusieurs rôles dans le processus de coordination. Un *planificateur* est un agent qui établit un plan et qui diffuse ses intentions aux autres agents. Un *exécuteur* est un agent qui exécute une action. Cet agent ne négocie pas les plans. Un agent est un *coordinateur* s'il propose une solution pour une relation entre plans. Les agents peuvent avoir un ou plusieurs rôles. S'il existe un agent auquel le seul rôle de coordinateur a été attribué, cet agent est appelé *médiateur*. La négociation est effectuée par l'intermédiaire du médiateur. Ainsi le médiateur est le seul à raisonner sur la coordination, en détectant des changements éventuels pour la synchronisation des plans, en explorant comment les agents devraient modifier leurs plans et en les informant de ses propositions.

2.2. Approche par réseaux de Petri récursifs

2.2.1. Modèle des réseaux de Petri récursifs

Un réseau de Petri (RdP) ordinaire (Figure 1) est un graphe bipartie orienté et valué. Les deux types de noeuds sont les places qui représentent des états du système (ressources, processus...) et les transitions qui représentent les événements.

Un réseau de Petri est souvent représenté par un quadruplet N = (P, T, A, W) où

$P = \{p_1, p_2, ..., p_m\}$ est l'ensemble des places,

$T = \{t_1, t_2, ..., t_n\}$ est l'ensemble des transitions,

A est l'ensemble des arcs,

W est la valuation des arcs. C'est une fonction à valeurs positives entières.

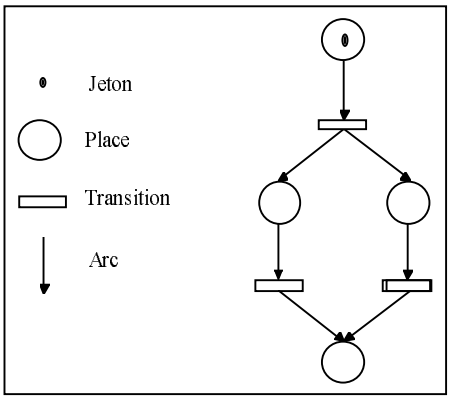

Figure 1 : Exemple de réseau de Petri ordinaire

L'occurrence d'un événement correspond au franchissement d'une transition. L'état courant du système est représenté par un marquage, un nombre de jetons dans chaque place. Le marquage d'un réseau évolue en fonction du franchissement de transitions. Les réseaux de Petri ordinaires sont adaptés à la représentation de la concurrence dans les systèmes distribués mais s'avèrent souvent inadéquats pour représenter les flux d'informations.

Contrairement aux réseaux de Petri ordinaires, les réseaux hiérarchiques de haut niveau (RHHN) (Jensen, 91) permettent la prise en compte des flux d'informations et leur représentation efficace. De plus, la notion de hiérarchie offre la possibilité de masquer des parties du réseau par abstraction et de les démasquer par dépliage en cas de besoin. Le dépliage (raffinement) d'un objet (place ou transition) génère un sous réseau d'un niveau hiérarchique immédiatement inférieur. Toutefois, les RHHN présentent une limitation importante pour la planification dynamique. En effet, ils ne permettent pas de gérer correctement le raffinement dynamique c'est à dire le raffinement du plan en cours d'exécution.

Nous avons donc opté pour un formalisme qui étend les RHHN en vue de leur utilisation en planification dynamique: les Réseaux de Petri Récursifs (El Fallah-Seghrouchni, 95). Selon les auteurs, une place est vue comme un noeud qui modélise une information relative au processus de résolution de problème, ou un état du processus de résolution. Nous associons à une place un domaine symbolisant la nature de l'information. Une transition est un noeud qui modélise une

action du plan de résolution et son franchissement correspond à l'exécution de l'action modélisée.

Trois types d'actions sont distinguées: les actions élémentaires, les actions abstraites et les actions de fin. Une action élémentaire au plus bas niveau de la hiérarchie est irréductible. Une action abstraite est une action décomposable en actions élémentaires. Elle représente une vue abstraite de la résolution et peut être raffinée en un nouveau plan. Les actions de fin permettent de replier correctement un plan en assurant la remontée du contexte d'exécution au niveau précédent.

2.2.2. Application à la planification distribuée

Le formalisme des RdP récursifs offre plusieurs avantages pour la planification multi-agent distribuée. Un plan est un ensemble coordonné d'actions, nous le représentons par un RdP récursif qui fournit une expression naturelle et graphique des synchronisations des activités parallèles que sont les exécutions des tâches des agents. Il permet d'exprimer l'ordonnancement des actions (relation d'ordre causale ou temporelle) des plans de résolution ainsi que le partage d'informations entre agents. De plus, il autorise l'affectation dynamique des tâches et la modification des plans de résolution par fusion ou raffinement.

L'aspect hiérarchique des RdP récursifs permet de considérer un système distribué à différents niveaux d'abstraction et de raffiner des transitions abstraites (une transition peut être une vue d'un sous-réseau) ce que ne permettent pas les graphes PERT par exemple (modèle de Koo). De plus, contrairement à ce qui se passe dans PGP, les propriétés formelles des RdP récursifs autorisent une analyse quantitative et qualitative (validation a priori) des plans de résolution ainsi modélisés ainsi qu'une indépendance par rapport au domaine. Enfin, l'approche que nous adoptons nous permet de choisir dynamiquement entre plusieurs plans concurrents à partir des propriétés structurelles des réseaux de Petri récursifs. Nous évitons ainsi une phase de négociation multiple (cas de l'approche von Martial) coûteuse en temps et dont la convergence vers un accord est incertaine.

Nous utilisons donc les RdP récursifs pour représenter un plan de résolution où l'état d'un réseau est déterminé à partir du marquage des places et des transitions activées avec une nouvelle sémantique. Dans l'état actuel de nos travaux, le marquage utilisé est celui des RdP ordinaires. L'aspect coloration des jetons n'est pas encore utilisé. Ainsi, un jeton agit comme un pointeur d'exécution.

3. LE MODELE D'AGENT

3.1. Structure générale

Notre modèle d'agent propose une architecture générique (figure 2). Les agents sont homogènes dans leur structure et se différencient par leurs connaissances de domaine. Un agent est composé de quatre modules: le module de planification, le module de manipulation de plans, le module de communication, et le corps cognitif. Ces quatre modules fonctionnent en parallèle. Un agent peut en effet planifier et exécuter en même temps ce qu'il a déjà planifié.

Seules les *connaissances du domaine* différencient les agents. Celles-ci sont formées des objets du domaine et des plans d'action qui les manipulent. Les plans sont génériques (non instanciés) et regroupés dans une bibliothèque. Ils sont instanciés par le module cognitif avec les

objets du domaine. Cet ensemble constitue le *corps cognitif* (voir paragraphe 3.2). Pour le moment, ces plans sont statiques et fixés à la création de l'agent (pas de faculté d'apprentissage). Pour maintenir la généricité des agents, ceux-ci ne raisonnent pas directement sur les objets du domaine. Le raisonnement est modélisé par les plans génériques.

Les agents du système interagissent entre eux et avec l'environnement. Ces interactions sont gérées par des mécanismes de *contrôle inter-agents*. A l'intérieur d'un même agent, les interactions entre les quatre modules sont gérées par le *contrôle intra-agent*. Les quatre modules partagent une structure de données commune (*tableau noir* ou *blackboard*) contenant le(s) plan(s) en cours d'exécution. Les modules échangent des messages de synchronisation (*notifications*) grâce à des *primitives de communication intra-agent*.

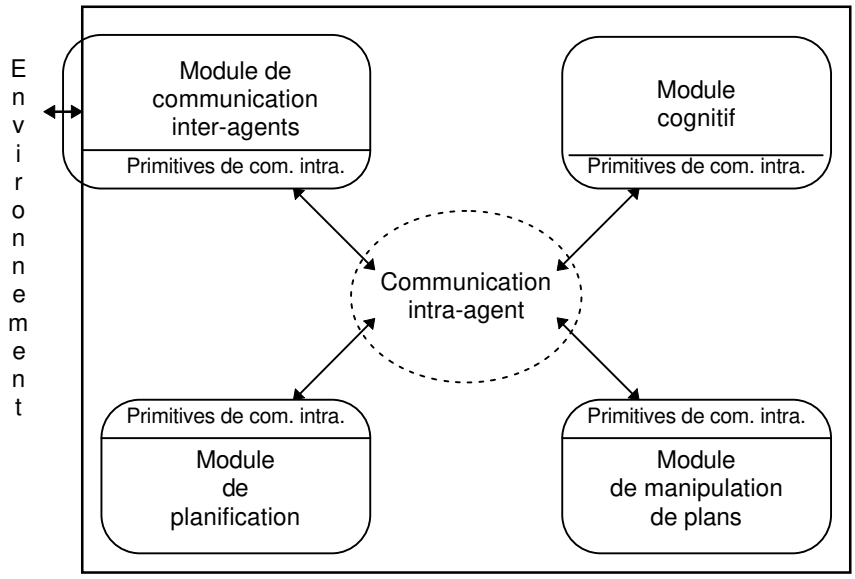

Figure 2: Structure d'un agent

3.2. Le corps cognitif

Le corps cognitif reconnaît des "pattern" à partir de séquences d'événements. Il propose alors au module de planification des plans d'action. Ces plans sont accompagnés d'une fiche d'identité contenant les critères extra-structurels (voir paragraphe 3.3). Le corps cognitif gère également les objets du domaine utilisés lors de l'instanciation (exécution) des plans.

Il est donc composé de :

- Une mémoire tampon : elle recueille les événements émanant de l'environnement,
- Un mécanisme associatif : il sert à reconnaître une situation particulière à partir d'une succession d'événements et propose le(s) plan(s) approprié(s),
- Une bibliothèque de plans : elle concrétise l'expertise relative au domaine d'application,
- Les objets du domaine avec leurs méthodes de manipulation.

Nous considérons que chaque événement arrivant à l'agent déclenche un plan. Cependant certains plans sont activés par une succession particulière d'événements (répétition, corrélations...). Pour cela, une structure tampon de stockage d'événements est nécessaire. Ce stockage doit se faire sur une durée significative pour le domaine d'application. L'étendue du

spectre temporel dépend du domaine d'application. Pour reconnaître une situation, on scrute en permanence la structure de recueil d'événements et on tente d'associer une séquence d'événements à un plan d'actions par l'intermédiaire de chroniques.

Selon (Dousson, 94), reconnaître c'est expliquer un ensemble d'événements datés, à l'aide d'un ensemble de schémas temporels (ou chroniques) décrivant comment l'association de plusieurs événements peut produire un nouvel événement et ce en tenant compte d'un ensemble de contraintes sur le domaine. Cette reconnaissance nécessite une représentation temporelle des événements. Modéliser le temps dans un système revient à définir les propriétés temporelles de base du domaine (point, intervalle...) et à associer des informations temporelles aux objets du domaine. Dans le cadre de la reconnaissance de chroniques, la logique à base d'événements apparaît la plus adaptée. Elle permet en effet la persistance par défaut d'informations et sa complexité algorithmique est plus réduite que celle des logiques à base d'intervalles (Dousson, 94).

L'expertise recueillie définit un modèle causal de mauvais fonctionnement. Elle est modélisée par un réseau de Petri récursif. On associe à chacun de ces plans une fiche d'identité. Cette fiche contient les valeurs de critères extra-structurels (par exemple: taux de succès constaté, temps moyen d'exécution, nombre moyen d'agents impliqués...) du plan. Elle rend possible l'évaluation des plans proposés par le module de planification de l'agent. Un ordre de préférence est donc établi à partir de critères structurels calculés par le module de manipulation (paragraphe 3.4) et extra-structurels (la fiche d'identité du plan). Cet ordre de préférence sera utilisé en cas de résultats contradictoires. Des critères contextuels (comme la charge d'un agent ou son état) permettent de déterminer l'opportunité d'une affectation à un agent candidat. L'affectation est réflexive puisqu'un agent peut raffiner un plan avec ses propres plans. Le choix des agents candidats est déterminé grâce aux accointances de l'agent demandeur (voir paragraphe 3.5).

3.3. Le module de planification

Le module de planification définit la stratégie d'exécution de plans de l'agent. Le choix d'un plan se fait indépendamment de la manière dont seront traités ultérieurement les éventuelles actions abstraites qu'il contient. Ce module planifie donc de façon hiérarchique en exploitant le formalisme des réseaux de Petri récursifs.

Le module de planification peut prendre deux types de décisions: lancer des plans, ou arrêter des plans. Le lancement de nouveaux plans est motivé soit par l'arrivée de nouveaux problèmes, soit pour raffiner des sous-plans abstraits. L'arrêt de plans peut être décidé soit pour des plans devenus inutiles, soit pour permettre de lancer des plans plus prioritaires. Il est à noter que plusieurs plans peuvent être exécutés en parallèle par un même agent.

Pour choisir entre plusieurs plans, le module de planification doit d'abord les comparer. Une étape d'évaluation de plans est donc nécessaire. Elle aboutit à l'attribution d'un score à chaque plan candidat. Ce score tient compte des caractéristiques propres aux plans mais aussi de l'état actuel de l'agent et de son environnement. Ce sont les critères: structurels, extra-structurels et contextuels cités précédemment.

Les critères structurels rendent compte de la structure d'un plan. Ils sont calculés à partir des propriétés des réseaux de Petri. Les critères extra-structurels sont indépendants de la représentation par réseaux de Petri et rendent compte de l'intérêt escompté de l'exécution d'un

plan. Ils constituent des estimations fondées sur les exécutions précédentes du plan. Les critères contextuels permettent d'évaluer l'adéquation des plans par rapport à l'état courant du système et reflètent les paramètres de l'environnement. Lorsque les valeurs de ces trois catégories de critères sont connues, il faut les agréger pour arriver à un score global du plan. Pour ce faire, nous utilisons une méthode multicritère permettant de pondérer les critères, s'assurer de leur homogénéité, puis les agréger convenablement. On aboutit à un classement des plans.

3.4. Le module de manipulation de plans

Le module de manipulation gère les plans au niveau de leur structure. Il fournit au module de planification des primitives de manipulation de plans telles que la fusion ou le raffinement. De plus, ce module doit pouvoir évaluer la structure d'un nouveau plan pour fournir la valeur des critères correspondants au module planification. En cas de contradictions structurelles entre plans, c'est le module de manipulation qui se chargera de les éliminer. Le module de manipulation lance l'exécution du plan ou d'une partie d'un plan lorsque les préconditions de déclenchement sont satisfaites, il assure le suivi de l'exécution.

3.5. Le module de communication

Deux types de communication coexistent dans le système: communication inter-agents et communication intra-agent (Moraïtis, 94). La communication inter-agents est à la charge du module de communication et est effectuée par échange messages. Ce module ne traite donc que les échanges entre l'agent et son environnement. Il utilise pour cela un module d'accointances structuré en une liste des agents connus avec leurs compétences (l'ensemble des transitions qu'ils sont capables de raffiner). La communication peut être de différents types. Elle peut être sélective, si l'agent confie ou délègue une tâche à un agent particulier (mode POINT TO POINT), ou diffusée si l'agent souhaite envoyer un message à plusieurs agents (mode MULTICAST) ou à tous les agents (mode BROADCAST). L'ensemble de protocoles physiques choisi pour la communication est TCP/IP (Hunt, 93).

Nous définissons trois types de message (Gostan, 95) échangés entre agents:

- REQUEST : demande d'un agent à un autre agent d'exécuter un plan. Cette communication n'attend pas de réponse,
- INFORM : transmission d'une information à un autre agent,
- REQUEST TO INFORM : requête d'un agent à un autre agent. L'agent à l'initiative de la communication a besoin d'une information ou de déléguer une partie de l'exécution d'un plan à un autre agent, puis intégrer la réponse à son propre travail.

Le module de communication garde une trace des messages qu'il envoie. Ceci permet à l'agent de gérer les messages en attente de réponse et au module de communication de surveiller les temps de réponse à ses requêtes. La trace d'un message contient l'identifiant unique généré au moment de l'émission du message, la date de l'envoi, le module d'origine et un script d'arrivée. Ce script permet à un module de modifier le traitement qu'il souhaite faire de la réponse à l'un de ses messages même après que ce dernier ait été envoyé.

Le module de communication échange uniquement des plans. Toute information est encapsulée dans un plan pour pouvoir être transmise. Le module ne gère pas la communication intra-agent, elle est à la charge de chaque module.

4. APPLICATION

Dans cette section, nous présentons un exemple de fonctionnement de notre modèle d'agent. Le domaine d'application est celui de la surveillance du réseau téléphonique commuté. Parmi l'expertise recueillie, nous avons retenu un plan de surveillance illustrant l'utilisation des réseaux de Petri récursifs.

4.1. Exemple de plan

Dans l'application présentée, chaque agent exécute en permanence un plan "Surveiller". Ce plan général est raffiné à l'arrivée d'une alarme. La figure 3 donne un exemple de raffinement possible pour l'alarme "A1" par le plan "P1".

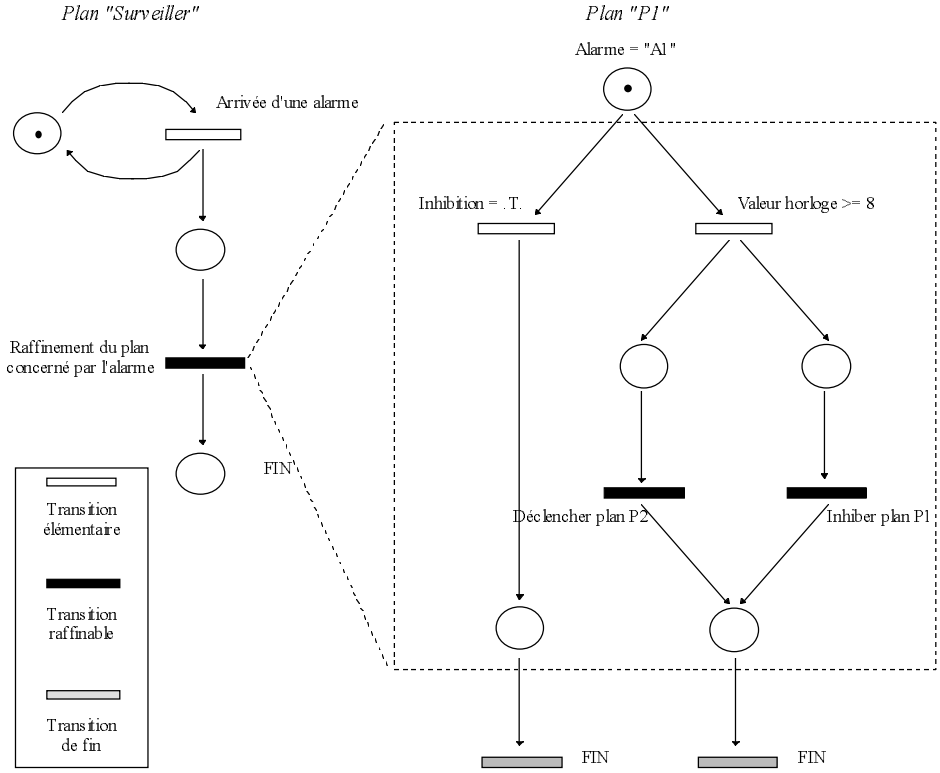

Figure 3: Exemple d'exécution d'un plan

Les transitions élémentaires correspondent aux actions à exécuter par l'agent. Les transitions abstraites ou raffinables correspondent à de nouveaux plans d'actions. Ceux-ci se trouvent chez l'agent lui-même ou chez ses accointances. Les transitions de fin signalent le repliage du plan.

4.2. Planification inter-agents

Comme décrit précédemment, la coopération entre les agents du système se fait par échange de plans. Les échanges de plans se font entre deux agents jouant chacun un des deux rôles suivants: *maître d'oeuvre* ou *exécutant*. Le premier transmet un plan, le second est chargé de l'exécuter. Les situations de coopération possibles sont données dans le tableau 1.

Agent maître d'oeuvre	Agent exécutant
Délégation sans attente de résultat	Raffinement et prise en charge du plan
Délégation avec attente de résultat	Raffinement et renvoi du plan partiellement ou totalement exécuté

Tableau 1: les cas de transmission de plans

4.3. Flux intra-agent

Les quatre modules de l'agent sont indépendants, fonctionnent en parallèle et communiquent de façon asynchrone (figure 4). On peut lancer plusieurs exécutions parallèles de sous-plans indépendants. C'est le module de planification qui détermine les sous-plans vérifiant ces contraintes et ce en tenant compte des critères structurels.

Le suivi de l'exécution se fait à l'aide d'une structure de données partagée (tableau noir) contenant le plan en cours d'exécution. Lorsque les unités d'exécution se terminent, le tableau noir contient le plan avec son marquage final (le résultat de l'exécution).

Les données écrites sur le tableau noir sont lues par les modules de manipulation et de planification. Le premier s'en sert pour enchaîner l'instanciation d'unités d'exécution du même plan. Le second les utilise pour décider des plans ultérieurs à exécuter. Dans certains cas, les données du tableau noir sont transmises à d'autres agents du système, ou à l'opérateur humain, par l'intermédiaire du module de communication.

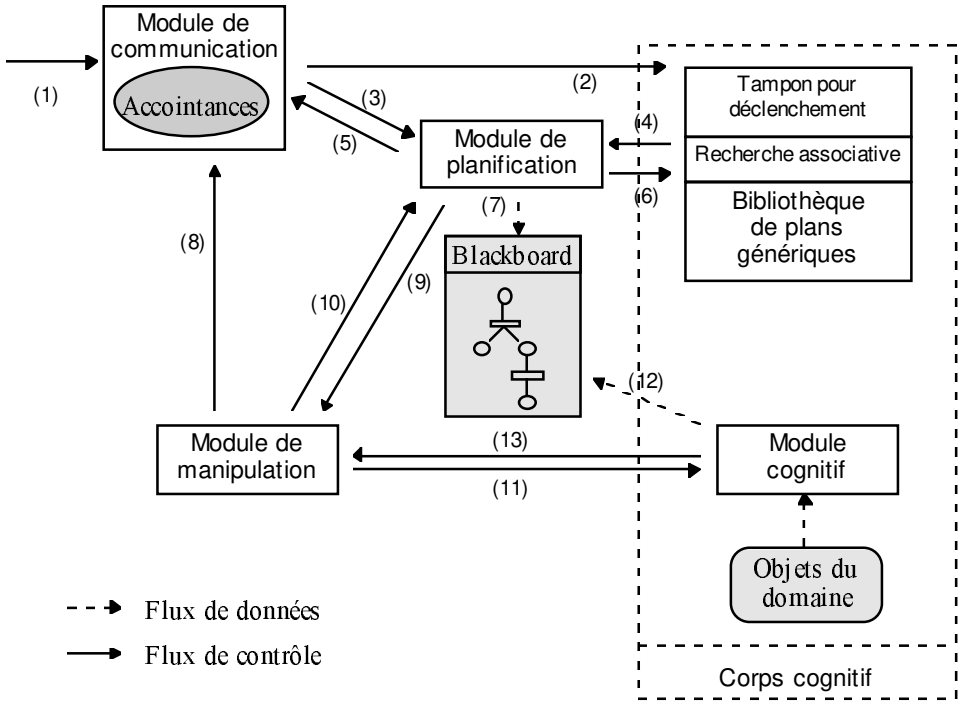

Figure 4: Diagramme des flux intra-agent

La figure 4 donne un aperçu des flux intra-agent. Pour illustrer ces flux, nous traitons ci-dessous le cas de la réception d'une alarme.

- Une alarme se produit sur l'application pilotée. C'est le module de communication qui en prend connaissance (1).
- Le module de communication identifie le message, il le stocke avec son identificateur dans le tampon du corps cognitif servant au déclenchement (2).
- La recherche associative propose deux plans de la bibliothèque pour l'alarme (ou les alarmes) en question: P_1 et P_2. Elle en informe le module de planification en lui envoyant également les valeurs des critères extra-structurels correspondants à ces deux plans (4).
- Le module de planification adresse une requête au module de manipulation pour demander des informations sur la structure des plans P_1 et P_2 (9).
- Le module de manipulation envoie au module de planification les valeurs des critères structurels des deux plans dont certains sont fonction de l'alarme A_i (10).
- Disposant des valeurs des critères structurels et extra-structurels, le module de planification évalue les deux plans P_1 et P_2 et décide de ne retenir que le plan P_1. Il le place alors dans le tableau noir (7).
- Le module de manipulation décompose le plan P_1 en unités d'exécution E_1, E_2, ..., E_n. Il gère ensuite le déroulement du plan en adressant des notifications au module cognitif pour lancer les unités d'exécution dont certaines seront déclenchées en parallèle (11).
- Le module cognitif instancie le plan dans le tableau noir avec les objets du domaine (12).
- Le module cognitif adresse une notification de fin d'instanciation au module de manipulation (13).

L'exécution d'un plan peut révéler, lors du raffinement d'action(s) abstraite(s), une nécessité de délégation. Elle est décidée par le module de planification, après consultation des accointances (5). Le module de communication se charge de la préparation des messages, de leur envoi aux agents concernés et assure leur suivi.

Chez l'agent exécutant, le déroulement du plan se fait d'une manière similaire au cas de réception d'une alarme. A la fin de l'exécution, le module de manipulation notifie au module de communication la fin de l'exécution (8) pour que celui-ci communique le marquage final à l'agent maître d'oeuvre.

5. CONCLUSION

Dans cet article, nous avons abordé le problème de la planification distribuée sous un nouvel angle en proposant une approche multi-agent fondée sur l'utilisation des réseaux de Petri récursifs. Le formalisme des réseaux de Petri récursifs offre plusieurs avantages pour la planification multi-agent distribuée. Celui-ci permet d'exprimer l'ordonnancement des plans de résolution ainsi que le partage d'informations entre agents. Il autorise de plus l'affectation dynamique des tâches et la modification collective des plans de résolution par fusion ou raffinement. Nous avons ainsi présenté une architecture d'agent générique capable d'exploiter ce formalisme. Les quatre modules (planification, manipulation, communication et corps cognitif) de l'agent fonctionnent en parallèle et permettent le partage dynamique et l'exécution distribuée de plans entre plusieurs agents autonomes de structure homogène.

BIBLIOGRAPHIE

Chaib-Draa B., Moulin B.
Problématique de la résolution distribuée de problèmes
Analyse de systèmes Vol. XIX, N°1-2, Mars-Juin 1993

Dousson C.
Suivi d'évolution et reconnaissance de chroniques
Thèse de Doctorat de l'Université Paul Sabatier de Toulouse, 1994

Durfee E.H., Lesser V.R., Corkill D.D.
Trends in cooperative Distributed Problem Solving
IEEE Transactions on Khowledge and Data Engineering, Vol. 1, N°1, March 1989

El Fallah-Seghrouchni A., Haddad S.
A formal model for coordinatig plans in multiagent systems
Proceedings of Intelligent Agents Workshop
Augustin Technology Ltd, Oxford Brooks University, UK, November 1995

Gostan O.
Communication dans les systèmes multi-agents
Mémoire de DEA de l'Université Paris Dauphine, 1995

Hunt C.
TCP/IP Network Administration
O'Reilly & Associates, Inc 1993

Jensen K.
High-level Petri Nets, Theory and Application
Springer-Verlag, 1991

Koo C.C.
A distributed model for performance systems: Synchronizing plans among intelligent agents via communication
PhD thesis, Stanford University, Stanford, CA, 1988

Laruelle H., Régnier P., Thiébaux S.
Dossier planification et action
Bulletin de l'AFIA, N°16, Janvier 1994

Moraïtis P.
Paradigme multi-agent et prise de décision distribuée
Thèse de Doctorat, Université Paris Dauphine, 1994

von Martial F.
Coordination plans of autonomous agents
Springer Verlag, 1992