

Un système multi-agents pour la fusion de données en analyse d'images

Multiagent System for Data Fusion in Image Processing

par Valéry LEFÈVRE*, Yann POLLET*, Sylvie PHILIPP**, Sylvie BRUNESSAUX*

* MATRA CAP SYSTÈMES, Département Etudes Systèmes et Informatique Avancée,

Parc d'Affaires des Portes, B.P. 613, F-27106 Val de Reuil cedex

** ENSEA, Equipe Traitement des Images et du Signal, 6, avenue du Ponceau F-95014 Cergy Pontoise cedex.

résumé et mots clés

La fusion des résultats en analyse d'images pose de nombreux problèmes de modélisation et de représentation des connaissances parmi lesquels, on peut mentionner :

- la difficulté de définir exhaustivement les conditions d'utilisation des traitements en fonction du contexte,
- la difficulté d'évaluer les résultats des traitements,
- la difficulté d'élaborer des stratégies faisant coopérer plusieurs traitements ou fusionnant des résultats.

Le système ADAGAR (Atelier de Développement d'AGents sur Architecture Répartie) propose des solutions à ces problèmes. ADAGAR résulte de concepts élaborés dans le projet BBI (BlackBoard Image). L'objectif du projet BBI était de concevoir sur une architecture multi-processeurs, un système perceptif d'interprétation d'images issues de sources différentes. L'idée n'était pas d'élaborer de nouveaux opérateurs de traitements d'images, mais de développer une structure d'accueil des traitements existants.

Une des particularités du système réside dans l'expression de connaissances d'estimation a priori (scores d'aptitude). La formalisation de ces connaissances est inspirée de la théorie de Dempster-Shafer. Le formalisme employé permet de représenter l'ignorance sur les conditions d'activation d'un traitement dans un contexte particulier.

Le prototype BBI contient actuellement une trentaine d'agents. Il fonctionne sur des couples d'images aériennes ou satellitaires dans les spectres visible/infrarouge et visible/radar. BBI élabore dynamiquement une ou plusieurs stratégies de recherche, à partir des spécificités des images traitées et des objets à détecter.

Les coopérations entre les traitements au bas niveau, les fusions d'informations et les focalisations d'attention au niveau symbolique sont les principales raisons du bon fonctionnement du prototype.

Intelligence artificielle distribuée, Système multi-agents, Système réparti, Interprétation d'images.

abstract and key words

The data fusion of results in image processing poses many problems of modelling and representation of knowledge, among which one can mention :

- the difficulty to define, exhaustively, the conditions of utilization of processings in function of the context,
- the difficulty to evaluate the results of processings,
- the difficulty to elaborate strategies making cooperate several processings or merging results.

The system ADAGAR (Atelier of Development of AGents on Distributed Architecture) proposes solutions to these problems. ADAGAR results from elaborated concepts in the project BBI (BlackBoard Image). The aim of the project BBI was to conceive on a multi-processors architecture, a perceptive system of interpretation of images issued from different sources. The idea was not to elaborate new operators of image processing, but to develop a reception structure of existent processing.

One of the particularities of the system lies in the expression of a priori valuation knowledge (scores of aptitude). The formalization of these knowledge is inspired of the theory of Dempster-Shafer. The formalism allows to represent the ignorance on conditions of activation of a processing in a particular context.

The prototype BBI contains currently a thirty of agents. It functions on couples of aerial or satellite images in the visible and the infrared spectra or in the visible and the radar spectra. BBI elaborates dynamically one or several strategies of research, from the specificities of the processed images and objects to detect.

The cooperations between processings at the low level, the information fusion and the focus of attention at the symbolic level are the main reasons of the good functioning of the prototype.

Distributed artificial intelligence, Multiagent system, Distributed system, Images interprétation.

1. introduction

Le domaine d'application de l'Intelligence Artificielle Distribuée est actuellement en pleine croissance, que les approches soient de type multi-agents classiques [Bau89, Gar89a] ou de type tableau noir [Gar89b, Gon87, Hay85, Laâ88, Nii86].

La vision par ordinateur est un domaine vaste où la connaissance n'est pas exprimable de manière unique. De plus, elle n'est pas toujours utilisable de la même façon. Par exemple, la connaissance de la forme géométrique d'un objet est beaucoup plus utile pour une recherche dans une image visible que dans une image infrarouge.

Les caractéristiques de cette connaissance et l'hétérogénéité de sa représentation rendent très performants les systèmes multi-agents qui permettent de la hiérarchiser et de la distribuer.

L'objectif du projet BBI (**BlackBoard Image**), contrat de recherche n°90.409 effectué pour le compte de la DRET, fut de concevoir, sur cette base, un système perceptif d'interprétation d'images issues de différentes sources, basé sur des mécanismes de fusion d'informations aussi bien au niveau symbolique, qu'au niveau numérique. L'intérêt d'un tel projet réside, dans notre cas, dans l'établissement d'un système opérationnel de photo-interprétation. Dans un tel système, une aide peut être apportée lors des deux phases suivantes :

- la phase de **Sélection / Détection** qui permet d'extraire des zones d'intérêt dans un grand flot d'images,
- la phase de **Photo-Interprétation Rapide** dont le but est d'identifier rapidement les alarmes.

Toutefois, la robustesse des résultats fournis par ce système doit justifier son utilisation. En complément des connaissances actuelles en analyse et en interprétation d'images, une modélisation de l'expertise du photo-interprète a permis de doter notre système de connaissances supplémentaires, pour déterminer l'information pertinente dans de telles recherches.

La nécessité de puissance de calcul liée au domaine d'application du système ainsi qu'au domaine du traitement d'images nous a conduits à définir des caractéristiques de systèmes pouvant s'exécuter facilement sur des architectures multi-processeurs diverses.

La modélisation que nous proposons pour un tel système multi-agents est inspirée du modèle du blackboard. Ainsi, une des originalités du modèle réside dans un mécanisme de contrôle particulier.

Le prototype BBI a été réalisé sur une station de travail reliée à un réseau de transputers. Son domaine d'application est la détection d'objets dans des couples d'images aériennes multispectrales. La généralisation des concepts de BBI a abouti à **ADAGAR** (**A**telier de **D**éveloppement d'**A**gents sur **A**rchitecture **R**épartie), outil indépendant du domaine d'application.

Dans un premier temps, cet article présente brièvement la philosophie de notre système multi-agents et l'implantation de ces principes sur une architecture distribuée [Lef94]. En particulier, nous détaillons l'utilisation de la théorie de Dempster-Shafer dans notre environnement (aptitude des agents, caractère imprécis des données). Il décrit ensuite l'intérêt d'un tel système du point de vue opérationnel au travers de la présentation de BBI et présente les perspectives d'évolution actuellement envisagées.

2. un système multi-agents sur architecture répartie

2.1. un modèle inspiré du modèle du blackboard

Afin de pouvoir inclure des traitements existants dans une structure d'accueil commune (volonté de réutilisation), plutôt que de tenter de définir des règles de comportement s'appliquant à un grand nombre d'entités coopérant, une organisation multi-agent basée sur le modèle social a été choisie. Ainsi, les agents d'ADAGAR sont cognitifs.

Concernant le choix du mode de communication (partage d'informations ou envoi de messages), la définition d'un compromis visant à intégrer les avantages de l'un et l'autre de ces modèles (sans en conserver les inconvénients) est proposée.

Les avantages d'utilisation du modèle du tableau noir, pour l'établissement d'un système de vision, se situent d'une part, au niveau de la structuration de la connaissance (hiérarchisation possible du tableau noir et des sources de connaissance), d'autre part, au niveau des modes de contrôle utilisables. Ces mécanismes de contrôle permettent, notamment, d'avoir un comportement de recherche descendant, ascendant ou mixte.

Cependant, bien que des extensions aient été proposées [Occ91] [Vel87] afin de le paralléliser, ce modèle est peu adapté à une architecture matérielle distribuée. De plus, la présence d'un superviseur est un avantage pour le contrôle de l'établissement des stratégies de résolution mais un inconvénient au niveau de l'aspect distribué du système.

Concernant l'exécution parallèle, le modèle basé sur la notion d'acteur s'avère beaucoup plus pertinente. En effet, puisque la connaissance est entièrement distribuée et que la communication s'effectue par envoi de messages, rien ne contraint une communication entre agents distants via un réseau quelconque.

Cependant, ce modèle a, lui aussi, des inconvénients. L'autonomie de chaque agent et la grande souplesse de l'architecture entraînent parfois un comportement global du système moins performant que celui d'un système plus contraint.

A partir de ces constatations, le modèle d'agent établi est fondé, tout d'abord, sur une *communication par envoi de messages*, ensuite, sur une *structuration de la société d'agents* en groupes d'abstraction croissante, enfin, sur un *contrôle distribué* des agents sur leurs homologues de niveau inférieur.

La figure 1 représente où se situe, dans le modèle décrit, l'inspiration issue du modèle du tableau noir. Toutefois, les différentes connexions représentées entre les agents matérialisent des canaux de communication possibles, mais elles ne sont, en aucun cas, des liens statiques prédéfinis (ceci afin de permettre des extensions souples du système).

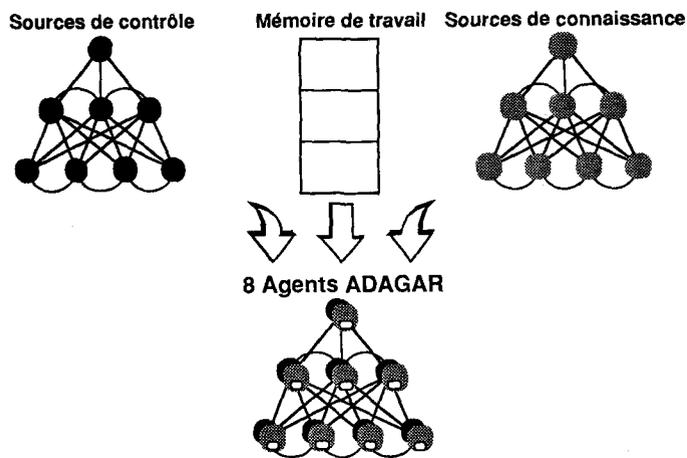


Figure 1. – Modélisation d'un système à partir d'ADAGAR.

Les communications sont basées sur un protocole d'appel d'offres inspiré du Contract Net [Smi80], modifié afin d'éviter une surcharge du flot de communications [Lef93a].

Ce modèle répond aux contraintes imposées au problème en ce qui concerne l'évolutivité, l'exécution parallèle et les possibilités de contrôle des stratégies de résolution. Les paragraphes suivants décrivent, plus en détail, les choix effectués pour satisfaire ces contraintes.

2.2. les agents

La description d'un agent est totalement indépendante de ses collègues. Ainsi, la création d'un nouvel agent ne nécessite aucune connaissance de ses confrères : **les agents ne possèdent initialement aucune accointance** (ceci afin de permettre une évolutivité aisée du système).

Lors d'une exécution, l'enrichissement de ces accointances s'effectue grâce aux échanges successifs entre agents.

Seule, la connaissance du niveau d'abstraction propre à l'agent est nécessaire (ou plutôt, du niveau d'abstraction de la société dans lequel il sera intégré). La structure d'un agent est complexe; ainsi, il peut disposer de buts, de comportements propres, de raisonnements ou autres particularités. Lors d'une exécution, les

agents sont « endormis », seule la réception d'une requête émise par un autre agent peut les « réveiller ». La requête ne peut venir que d'un agent de même niveau ou d'un niveau immédiatement supérieur.

Afin de plus facilement comprendre cette structure, une définition préalable de certains termes utilisés est nécessaire.

2.2.1. définitions

- Le **contexte** est un ensemble de couples (attribut, liste de valeurs) précisant les principales conditions dans lesquelles doit s'effectuer la résolution.

Ainsi, une recherche sur un couple d'images, issues d'un capteur infrarouge (aérien) à basse altitude et d'un capteur radar (satellitaire) à haute altitude, prises à 14 heures avec un temps ensoleillé, a un contexte de la forme :

{(IMAGE(IR RADAR)) (ALTITUDE (BAS HAUT))
(METEO ENSOLEILLE) (HEURE 14.00)}

Les paramètres non cités sont inconnus. Dans notre exemple, le jour précis n'est pas fourni, le contexte contient implicitement l'information :

{(JOUR INCONNU)}

- Le **problème** décrit un besoin particulier. Il est également défini sous la forme d'un ensemble de couples (attribut, liste de valeurs). Une détection de véhicules s'exprime sous la forme :

{(PROBLEME VEHICULE) (OBJECTIF DETECTION)}

- Le **contexte d'exécution** représente l'univers dans lequel les agents doivent fonctionner. Il est formé d'un problème et d'un contexte.

Par exemple, le contexte d'exécution d'une détection de véhicules à partir d'images infrarouge et radar peut alors prendre la forme :

{(IMAGE(IR RADAR)) (ALTITUDE (BAS HAUT))
(METEO ENSOLEILLE) (HEURE 14.00)
(PROBLEME VEHICULE) (OBJECTIF DETECTION)}

- Dans ADAGAR, une **requête** est un message envoyé par un agent à un ou plusieurs autres agents. Elle est formée d'un type de besoin (appel d'offre, demande de sous-traitance ou de données particulières) et d'un contexte d'exécution.

Par exemple, un appel d'offre pour connaître « *Qui peut détecter des véhicules à partir d'un couple d'images IR et Radar...?* » peut prendre la forme :

Type de la requête :

Requête QUI PEUT (appel d'offre);

Contexte d'exécution de la requête :

{(IMAGE (IR RADAR)) (ALTITUDE (BAS HAUT))
(METEO ENSOLEILLE) (HEURE 14.00)
(PROBLEME VEHICULE) (OBJECTIF DETECTION)}

- Les **paramètres d'exécution** sont des variables internes à chacun des agents pouvant être modifiées et ayant une influence

sur le comportement de l'agent. Ils contiennent au minimum tous les paramètres du traitement qu'encapsule l'agent.

Ainsi, un agent de bas niveau tel que « *Fermeture Morphologique* » a au minimum comme paramètres d'exécution les tailles X et Y de l'élément structurant de son traitement.

2.2.2. l'expertise d'un agent

A partir de ces définitions, l'expertise d'un agent s'exprime sous trois formes distinctes, tout d'abord, une expertise d'estimation a priori de la capacité de l'agent à résoudre le problème posé, ensuite, une expertise sur le contrôle des actions que peut effectuer l'agent (paramétrage), et enfin, une expertise d'estimation de la qualité a posteriori d'un résultat calculé.

• **L'estimation a priori de la capacité de l'agent à résoudre le problème posé** étend la notion de condition dans les sources de connaissance classiques de la forme (Condition–Action).

Cette expertise permet à un agent de répondre aux appels d'offre lancés par ses collègues. Elle exprime l'écart entre le contexte optimum de fonctionnement de l'agent et le contexte d'exécution courant. Cette estimation est qualifiée par une échelle de valeur : le score d'aptitude (entre 0 et 1; 0 équivaut à aucune aptitude à résoudre le problème et 1 à une aptitude parfaite).

Ce score d'aptitude permet de rendre l'estimation d'un agent compréhensible (et utilisable) par tous les autres agents sans que ces derniers n'aient besoin d'informations complémentaires.

• **Le contrôle des paramètres d'exécution** rend l'agent autonome. Il dirige entièrement ses actions grâce à des pondérations de ses paramètres d'exécution. Ce contrôle permet ainsi à un agent de modifier son propre comportement. Cette expertise concerne également la modification de ces paramètres en vue d'une nouvelle exécution pour améliorer un résultat (notion de bouclage interne).

• **L'analyse a posteriori de la qualité d'un résultat** décrit comment un agent interprète les données qu'il vient de calculer.

Elle s'exprime sous la forme d'un score de réussite. Comparée aux deux premières formes d'expertise, cette analyse est plus difficile à déterminer. Parfois, elle n'est pas exprimable au niveau de l'agent mais à un niveau plus abstrait. Dans ces cas, l'agent envoie un *compte rendu d'action* plutôt qu'une réelle analyse (là aussi qualifié par le score de réussite).

L'échelle de valeur exposée au travers du score de réussite (entre 0 et 1; 0 équivaut à un résultat estimé très mauvais et 1 à un très bon résultat; ceci par rapport à la connaissance propre à l'agent) permet aux différents agents d'interpréter (ou de tenter d'interpréter) le résultat d'un de leurs collègues.

Il est important de comprendre qu'il n'y a aucun lien entre le score d'aptitude et le score de réussite d'un agent.

En fait, ces scores d'aptitude et de réussite permettent aux agents de pouvoir comprendre (et être compris par) leurs homologues.

Ainsi, lorsqu'un agent estime qu'il n'a pas obtenu de bons résultats en réponse au problème posé, il renvoie un score de

réussite faible à l'émetteur alors que son score d'aptitude était satisfaisant. Toutefois, ce dernier (situé à un niveau plus haut) peut être satisfait par ces résultats au vue de ce qu'il désire en faire.

2.2.3. la modélisation du contrôle d'ADAGAR

Dans ADAGAR, le contrôle des agents entre eux, est exprimé sous deux aspects **les filtres de préconditions** f_{ki} (c'est-à-dire l'estimation a priori) et **les filtres de coopération** g_{ki} (permettant à un agent de coopérer et de contrôler ses collègues).

Soient :

- $P = \{P_1, \dots, P_n\}$: l'ensemble P des n problèmes posés;
- $A_k = \{A_{k1}, \dots, A_{ks}\}$: l'ensemble A_k des s agents du niveau k ;
- $C_1 \times C_2 \times \dots \times C_p$: l'ensemble des attributs possibles pour chacun des « p » caractéristiques du contexte d'exécution
- Cr : l'ensemble des critères de choix de sous-traitants

Pour chaque agent A_{ki} , **la fonction** f_{ki} détermine l'aptitude r_{ki} de l'agent à résoudre un problème donné :

$$f_{ki} : P \times C_1 \times C_2 \times \dots \times C_p \times ([0, 1]^S)^m \rightarrow [0, 1]$$

$$(P_i, c_1, c_2, \dots, c_p, r_{11}, \dots, r_{s1}, r_{21}, \dots, r_{sm}) \rightarrow r_{ki}$$

où P_i est le problème posé, les c_i les valeurs des caractéristiques du contexte d'exécution, les r_{ij} sont les scores d'aptitude « des agents auxquels l'agent A_{ki} a soumis ses m sous-problèmes pour résoudre P_i » et r_{ki} représente le score d'aptitude de l'agent. Il est important de remarquer que cette fonction dépend des aptitudes d'autres agents pour son estimation propre.

Ce score r_{ki} est, en fait, la probabilité que l'agent puisse répondre au contexte posé.

De plus, pour chaque agent A_{ki} **la fonction** g_{ki} représente l'heuristique de l'agent pour déterminer quels collègues il va déclencher (lors d'une requête « Toi Réalise ») pour résoudre son sous-problème « numéro j » :

$$g_{ki} : Cr \times P \times C_1 \times C_2 \times \dots \times C_p \times [0, 1]^s \rightarrow B^s$$

$$(C_i, P_i, c_1, c_2, \dots, c_p, r_{1j}, \dots, r_{sj}) \rightarrow (b_1, \dots, b_s)$$

où C_i est le critère de choix utilisé (qui peut être la rapidité, la précision, ...), P_i , c_i et r_{ij} ont la même signification que précédemment et les booléens b_i indiquent le déclenchement des différents agents ($b_i = 1$ signifie l'agent i est déclenché). Notons que l'agent contrôlant le déclenchement (l'émetteur de la requête) ne tient pas uniquement compte des scores d'aptitude mais également du contexte d'exécution de la requête et du critère de choix. Cela signifie que sa connaissance propre peut le conduire à déclencher un agent qui n'a pas nécessairement le meilleur score d'aptitude (voir chapitre 4).

2.3. la formalisation de la connaissance

Dans l'estimation de l'aptitude, il s'agit de rendre compte pour chaque agent de sa capacité a priori à résoudre un problème donné dans des conditions données. L'ensemble de ces informations est contenu dans le contexte d'exécution. Une approche peut alors consister à modéliser de manière subjective l'aptitude a priori comme une fonction associant à toute valeur de contexte une valeur d'aptitude issue d'un recueil d'expertise, comprise entre 0 (aucune aptitude dans ce contexte) et 1 (aptitude parfaite).

Un premier constat fondamental, et largement confirmé par l'expérience, réside dans le caractère essentiellement non déterministe des conditions d'utilisation d'un traitement d'images (liées, notamment, au bruit). Le contexte d'exécution n'est en effet qu'une schématisation de l'information réellement présente dans l'image, sur laquelle est appliqué le traitement. Aussi un résultat de traitement possède un caractère aléatoire dû à la difficulté de décrire parfaitement une image : ignorance de certains paramètres de la prise de vue, difficulté à quantifier l'information présente sur l'image (type de bruit, présence de texture, ...).

Il faut également noter que la *connaissance* que l'on a des traitements est *incomplète* : il est impossible de rendre compte de manière exhaustive de la pertinence d'un traitement donné dans l'ensemble des contextes possibles. Ainsi, dans de nombreux cas, on ne possède que des indications sur les domaines possibles d'aptitude. **Une simple probabilité est donc insuffisante pour modéliser la connaissance relative à l'aptitude.**

Précisons enfin qu'un contexte d'exécution peut lui-même comporter une description incomplète de l'image et du problème à traiter. Certains attributs, significatifs pour l'aptitude d'un traitement, et non présents dans le contexte, doivent alors être interprétés comme des valeurs inconnues. Ceci nous amènera également dans ce cas à rechercher non l'aptitude mais les valeurs éventuelles de celle-ci dans l'état donné de la connaissance.

Parmi les différentes approches envisageables (raisonnement bayésien, approches probabiliste, floue, etc.), les remarques précédentes ont conduit à utiliser un formalisme inspiré de la théorie de Dempster-Shafer [Dem68] [Sha76] pour la modélisation. **Celle-ci permet en effet de représenter à la fois la nature probabiliste du phénomène considéré [Pit90], et la connaissance imparfaite que l'on a de celui-ci [Hât91].** La connaissance relative à l'aptitude d'un traitement est ainsi caractérisée dans notre approche par deux fonctions associant à toute valeur de contexte une *crédibilité* et une *plausibilité*. Le filtre de préconditions présenté au paragraphe 2.2.3 retourne alors un intervalle $[Cr, Pl]$.

2.3.1. la théorie de Dempster-Shafer

Cette théorie attribue de la confiance indifféremment à des sous-ensembles aussi bien qu'à des éléments particuliers de l'ensemble des hypothèses d'apparition d'événements. Elle s'exprime à partir

de deux grandeurs, la **crédibilité** et la **plausibilité**; ces deux valeurs pouvant, ensuite, être comparées à la probabilité correspondante. Ces deux grandeurs sont comprises entre 0 et 1.

La crédibilité d'un fait $Cr(F)$ mesure son degré de certitude; sa plausibilité $Pl(F)$ mesure à quel point il peut être nié. On a la relation suivante entre ces deux valeurs et la probabilité correspondante :

$$\bullet Pl(F) \geq P(F) \geq Cr(F)$$

Un fait et son contraire sont également en relation, l'égalité suivante utilise, ainsi, le fait que : « plus F est certain, moins sa négation est probable ».

$$\bullet Cr(F) = 1 - Pl(\neg F)$$

La mesure de crédibilité est établie à partir d'une distribution de probabilité m de l'ensemble E des événements dans l'intervalle $[0, 1]$ définie par :

- $m(\text{Faux}) = 0$; (i)
- $\forall p, p \in E, m(p) \geq 0$; (ii)
- $\sum_{p/p \in E} m(p) = 1$; (iii)

A partir de cette distribution, la crédibilité d'un événement p se calcule en additionnant toutes les masses affectées à p et à ses sous-ensembles. La plausibilité de p vaut 1 moins la crédibilité du complémentaire de p :

- $Cr(p) = \sum_{q/q \rightarrow p} m(q)$ (iv)
- $Pl(p) = 1 - Cr(\neg p)$ (v)

La croyance dans un événement p s'exprime, donc, par un intervalle $[Cr(p), Pl(p)]$ contenu dans l'intervalle $[0, 1]$; la taille de cet intervalle (c'est-à-dire le résultat de $Pl(p) - Cr(p)$) traduit le degré d'incertitude sur la valeur de probabilité.

L'un des intérêts majeurs de cette théorie est la prise en compte de l'ignorance, ce point est abordé dans les paragraphes suivants.

2.3.2. utilisation pour la formalisation des scores d'aptitude

A partir d'un contexte d'exécution, chaque agent doit fournir une estimation de son aptitude à résoudre le problème posé. C'est-à-dire qu'à partir de ses connaissances propres, il doit être capable d'évaluer si le contexte proposé est proche ou non des contextes pour lesquels il fonctionne, en général, de façon satisfaisante. Ce calcul est établi en appliquant son filtre de préconditions (sa fonction f_{ki}) au contexte d'exécution donné.

Pour cela, la modélisation de ce filtre tient compte des conditions d'activation optimales de l'agent.

« Comment s'effectue alors le recueil d'expertise concernant ces conditions d'activation? »

Les attributs influant sur le comportement et le travail d'un agent diffèrent d'un agent à l'autre; de plus, la sensibilité à l'un ou l'autre de ces attributs varie également selon les agents.

- Temps (ensoleillé) 0,10;
- Altitude (bas) 0,15;
- Problème (véhicule),
BON (sous-traitant, rectangle) 0,25;
- Problème (véhicule),
EXISTE (sous-traitant, rectangle) 0,20;
- Objectif (détection) 0,10;
- Nb Capteurs > 1 0,10;
- Incertitude de la connaissance (W) 0,10.

Le filtre de préconditions de cet agent (f_{ki}) est modélisé à partir des sous-ensembles d'événements précédents et de leur masse de probabilité respective.

La présence de l'événement particulier EXISTE (respectivement BON) a pour rôle de vérifier l'existence d'un sous-traitant de qualité non précisée (respectivement d'un bon sous-traitant) au sous-problème posé à savoir une fourniture de rectangles (la réponse de l'agent Véhicule est fonction de ses sous-traitants potentiels).

Remarquons, que **plus ce découpage en sous-espaces est précis et prend en compte un grand nombre de dimensions de l'univers des attributs (voire tous), plus l'intervalle $[Cr, Pl]$ de l'agent est petit**. L'inverse, par exemple, lorsque l'on affecte des masses de probabilité à des sous-espaces événementiels de taille $N - 1$ dans l'univers de dimension N (comme pour les événements Temps, Altitude, Objectif et Nb Capteurs ci-dessus) entraîne une augmentation de l'intervalle $[Cr, Pl]$:
« la plausibilité que l'agent sache répondre est alors grande, cependant, sa crédibilité est faible à cause de l'insuffisance d'informations! ».

Sur l'exemple de l'agent Véhicule donné ci-dessus, avec le contexte d'exécution Ct_1 formé des événements suivants {Temps (gris), Altitude (bas), Problème (véhicule), Objectif (Détection), Nb Capteurs = 1} et l'hypothèse que l'agent Véhicule puisse obtenir un sous-traitant moyen pour son sous-problème de recherche de rectangles (événement EXISTE satisfait) mais pas un bon sous-traitant (événement BON non satisfait), alors l'intervalle d'aptitude de l'agent Véhicule vaut :

$$\bullet Cr_{V\acute{e}hicule}(Ct_1) = 0,15(Altitude) + 0,20(Probl\grave{e}me) + 0,10(Objectif) = 0,45 ;$$

$$\bullet Pl_{V\acute{e}hicule}(Ct_1) = 1 - Cr_{V\acute{e}hicule}(\neg Ct_1) = 1 - (0,10(Temps) + 0,10(Nb\ Capteurs)) = 0,80$$

Pour le contexte Ct_1 , le score d'aptitude retourné par le filtre de préconditions est alors $[0,45 ; 0,80]$.

Notons qu'un contexte d'exécution Ct_2 moins précis tel que, par exemple, { Problème (véhicule), Objectif (Détection), Nb Capteurs = 1} (sans information sur les conditions météorologiques et l'altitude de prise de vue), fait varier la taille de l'intervalle

d'aptitude calculé par le filtre de préconditions :

$$\bullet Cr_{V\acute{e}hicule}(Ct_2) = 0,20(Probl\grave{e}me) + 0,10(Objectif) = 0,30 ;$$

$$\bullet Pl_{V\acute{e}hicule}(Ct_2) = 1 - Cr_{V\acute{e}hicule}(\neg Ct_2) = 1 - 0,10(Nb\ Capteurs) = 0,90.$$

Pour le contexte moins précis Ct_2 , le score d'aptitude retourné par le filtre de préconditions devient $[0,30 ; 0,90]$.

La mise en place du calcul de ces scores d'aptitude n'est pas à la charge du concepteur. En effet, l'environnement ADAGAR contient une classe particulière nommée la classe des prédicats permettant de calculer la crédibilité et la plausibilité d'un agent à résoudre le problème posé à partir de la définition d'un ensemble de prédicats événementiels satisfaits ou non.

Ensuite, à partir des valeurs de crédibilité et de plausibilité calculées, une autre classe, dans ADAGAR, nommée **la classe des contrôleurs** (définie à partir d'un ensemble d'heuristiques) détermine, en prenant en compte les critères imposés par l'émetteur de la requête (tels que, par exemple, « Bonne qualité des résultats » ou « Obtention rapide de ces résultats »), s'il existe un ou plusieurs sous-traitants possibles. L'utilisation de cette seconde classe n'est pas obligatoire, ce protocole décisionnel peut être décrit dans certains agents très spécifiques; néanmoins, elle permet au concepteur de se soustraire à ce choix s'il le désire.

2.4. la plate-forme de développement ADAGAR

ADAGAR est l'outil permettant d'une part, de générer de nouveaux agents et d'autre part, de gérer leur exécution (positionnement des agents, communication entre agents,...). Un tel atelier [Lef94] facilite le développement **d'applications modulaires et extensibles**.

Le développeur s'affranchit de l'aspect exécution sur la couche matérielle de ses agents. En effet, ADAGAR gère dynamiquement l'accès aux ressources physiques en fonction des agents s'exécutant. Ces ressources physiques peuvent être hétérogènes ou réparties [Ban91]. La philosophie de ADAGAR permet l'écriture unique d'une nouvelle interface de communication bas niveau (dépendant du type du processeur) pour prendre en compte de nouvelles ressources physiques.

De plus, il est possible de hiérarchiser la société des agents en différents niveaux d'abstraction (voir figure 2). Une fois cette hiérarchisation effectuée, les nouveaux agents d'un niveau donné contrôlent directement ceux du niveau immédiatement inférieur. L'interface homme-machine est modélisée comme un agent du niveau le plus abstrait.

La structure de l'atelier ADAGAR permet de générer de nouveaux agents de plusieurs manières. Un agent peut être développé sans tenir compte des boîtes à outils proposées dans ADAGAR

(tels que les classes de contrôleurs), en utilisant uniquement les bibliothèques de communication permettant de le rendre compréhensible par les agents déjà existants. Par ailleurs, les agents peuvent être implantés en encapsulant des applications externes du type systèmes experts ou interface avec une base de données. La classe des contrôleurs est un de ces outils. Elle contient des contrôleurs génériques qui permettent au développeur de créer de nouveaux agents. C'est utile, par exemple, pour exprimer un contrôle « focalisé sur la qualité des résultats », sur « l'obtention rapide de ces résultats » ou sur « une utilisation maximale des ressources physiques ».

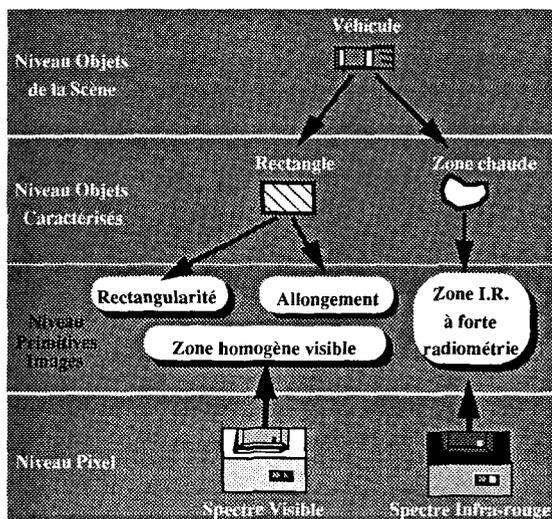


Figure 2. – Hiérarchisation du raisonnement au travers d'une détection de véhicules.

3. Un système de fusion de données pour l'analyse d'images

L'application **BBI (BlackBoard Image)** répond à l'étape de Détection / Sélection de la tâche du photo-interprète [Lef93b]. Son implantation dans un cadre opérationnel est envisagée sous la forme d'un poste de photo-interprétation bi-écran. L'un est réservé au photo-interprète, l'autre est géré par BBI et fonctionne automatiquement et envoyant des alarmes lorsque sa détection ne correspond pas à celle du photo-interprète.

Schématiquement, la résolution d'un problème de détection se traduit par l'action simultanée de deux tâches : **la génération dynamique de chaînes de traitements**, puis, **la prise en compte des résultats obtenus**, qui est effectuée par des mécanismes tels que la **fusion** ou la **focalisation par communication de résultats**.

3.1. les agents de bbi

L'application compte actuellement une trentaine d'agents répartis sur quatre niveaux d'abstraction (figure 2) partant d'un « niveau Objet de la Scène », indépendant de l'image [Mat89] jusqu'au « niveau Pixel » lié aux traitements primaires applicables à une image. Ces niveaux ont été déterminés à partir du dépilage de nombreux scénarios de détection. Un exemple de dépilage est donné en figure 2.

La généralité des agents est liée au niveau auquel ils sont attachés. Plus le niveau est bas, plus les agents sont génériques et, par conséquent, réutilisables.

La modélisation de la connaissance liée au domaine d'application est, elle aussi, originale. En effet, à partir d'une bibliothèque de traitements d'images disponible, le spécialiste en traitement d'images a qualifié chaque traitement, en termes de problème posé, de conditions d'utilisation, de besoins pour bien fonctionner. Chaque traitement a, ainsi, été recensé; ensuite, cette connaissance a permis de définir de façon totalement autonome chacun des agents en l'interfaçant dans ADAGAR.

3.2. élaboration des stratégies

Dans l'application que nous traitons, la génération des chaînes de traitements doit être dynamique parce que la résolution d'un problème est différente selon le contexte d'exécution (les agents intervenant n'étant pas les mêmes).

Un des objectifs du prototype était de montrer l'apport du comportement opportuniste des agents. Différentes stratégies de résolution sont ainsi engendrées dynamiquement et entraînent globalement une amélioration sensible des résultats.

Ainsi, BBI facilite la mise en œuvre :

- de fusion de résultats issus de différentes chaînes de traitements sur une image ou issues de traitements sur différentes images.
- de focalisation d'attention sur des zones d'intérêt déterminées par des traitements préalables, ou des interrogations à des agents spécialisés (recherche de données cartographiques par exemple).
- de coopération entre différents traitements d'images bas niveau en vue d'améliorer, si possible, la segmentation.

3.2.1. fusion de données

Cette stratégie correspond au scénario classique du lancement en parallèle de plusieurs chaînes de traitement. La fusion se fait à différents niveaux, par comparaison des résultats obtenus par chaque chaîne. Dans le cas de la détection de véhicules, au niveau Objets-Caractérisés, par exemple, cette fusion utilise un critère de localisation basé la position du barycentre des points des régions concernées : dans le cas où deux régions, détectées chacune par une chaîne différente, présentent des barycentres proches, une comparaison s'effectue par calcul du recouvrement mutuel des

régions. En cas de réponse favorable, la région intersection est considérée comme valable.

3.2.2. focalisation d'attention

Dans ce cas, ce sont les régions détectées par une première chaîne de traitement qui vont être utilisées pour piloter une seconde par focalisation d'attention sur les résultats obtenus. La seconde chaîne de traitements attend que les résultats de la première lui soit communiqués pour s'exécuter. Ce mécanisme influe essentiellement sur la robustesse de la résolution (les traitements locaux peuvent, en effet, être beaucoup plus adaptés aux vignettes traitées que les traitements globaux).

3.2.3. coopération multi-canaux

Le principe est, lorsque cela est possible, de détecter au préalable les zones les plus homogènes en suivant des stratégies adaptées à la quantité de bruit de l'image. Le parallélisme des agents permet de faire une détection sur de nombreuses petites zones en même temps. Dans le cas de la localisation des cours d'eau, la recherche est orientée par des informations cartographiques lorsque celles-ci sont disponibles. D'une part, elles apportent une précision au niveau de la localisation des cours d'eau. D'autre part, elles permettent de découper l'image globale en un ensemble de petites vignettes englobant les cours d'eau et de diminuer le coût des traitements à effectuer en les traitant en parallèle. Actuellement, ces trois stratégies de résolution sont dynamiquement établies à partir de coopération entre agents.

Deux d'entre elles se gênent en compétition pour répondre à une détection de véhicules :

- **fusion haut niveau** : une recherche est effectuée sur chaque image, puis une fusion se fait pour essayer de mettre en correspondance des résultats obtenus des différentes sources.
- **focalisation par communication de résultats** : de façon heuristique, il est plus robuste de chercher l'information dans une image infrarouge, peu précise pour la localisation, pour ensuite, à partir de zones pertinentes, effectuer une recherche locale en imagerie visible.

La troisième se crée lors d'une détection de ponts :

- **coopération multi-canaux** : les meilleures sources d'informations selon le contexte sont sélectionnées. Cette stratégie présente encore plus d'intérêt lorsque l'on travaille sur des triplets d'images visible/IR/radar afin de choisir le meilleur canal optique.

Une mise en place dynamique de cette coopération multi-canaux, en réponse à une recherche de ponts, est présentée en figure 3. L'agent Pont essaie tout d'abord de focaliser la recherche à partir de données cartographiques a priori (demande de recherche de fleuves). Ensuite, l'agent ZoneHomogèneMultiCanaux détermine un masque des fleuves de l'image indépendamment de la bande spectrale où ils ont été trouvés. Au niveau Primitives-Images, la recherche de ces régions est dirigée par l'agent RégionsSombres

qui traite des régions issues de l'image visible, de l'image radar ou des deux (fusion des résultats) selon les conditions météorologiques. Après la détermination de ces masques de fleuves, une focalisation d'attention de l'agent RégionsBrillantes permet de détecter les voies de franchissement (ponts et écluses), là aussi grâce à une fusion des régions brillantes obtenues et d'informations caractéristiques extraites de l'image radar.

Les coopérations entre les traitements au bas niveau, les fusions d'informations et les focalisations d'attention au niveau symbolique sont les principales raisons du bon fonctionnement du prototype.

Le prototype, développé sur une station de travail reliée à un réseau de transputers, a été dédié initialement à une application de surveillance du champ de bataille. Dans le cadre plus général de la photo-interprétation, BBI a ensuite été complété (ajout d'agents) afin de permettre la détection d'objets particuliers tels que les voies de franchissement de cours d'eau.

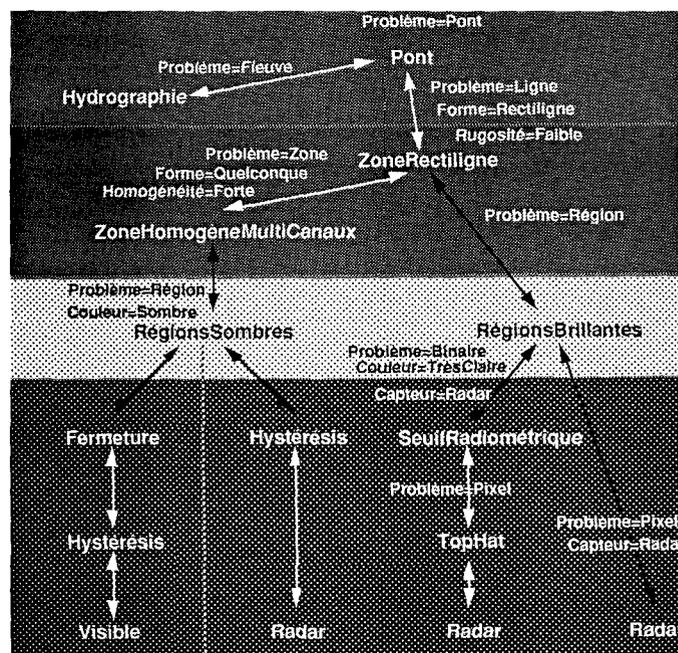


Figure 3. – Instanciation des agents pour la détection de ponts.

La figure 4 et la figure 5 présentent des images écrans du moniteur d'exécution ADAGAR. Dans la figure 4, la détection de véhicules se fait dans un couple d'images aériennes visible (à gauche) et infrarouge (à droite) prises par temps ensoleillé. Dans ce cas, deux stratégies sont suivies afin d'augmenter la robustesse des détections : fusion haut niveau et focalisation.

Dans la figure 5, la détection de ponts s'effectue sur un couple d'images visible (à gauche) et radar (à droite) prises par un temps nuageux.

Un système multi-agents pour la fusion de données

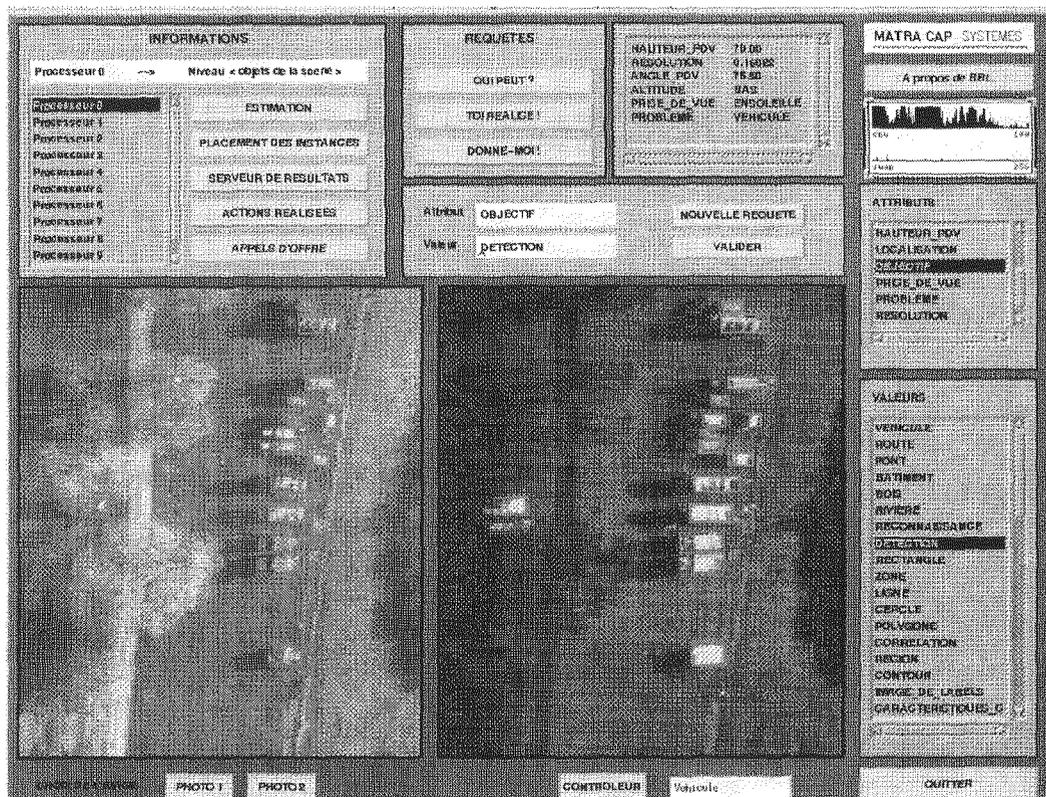


Figure 4. – Détection de véhicules (conditions météorologiques ensoleillées).

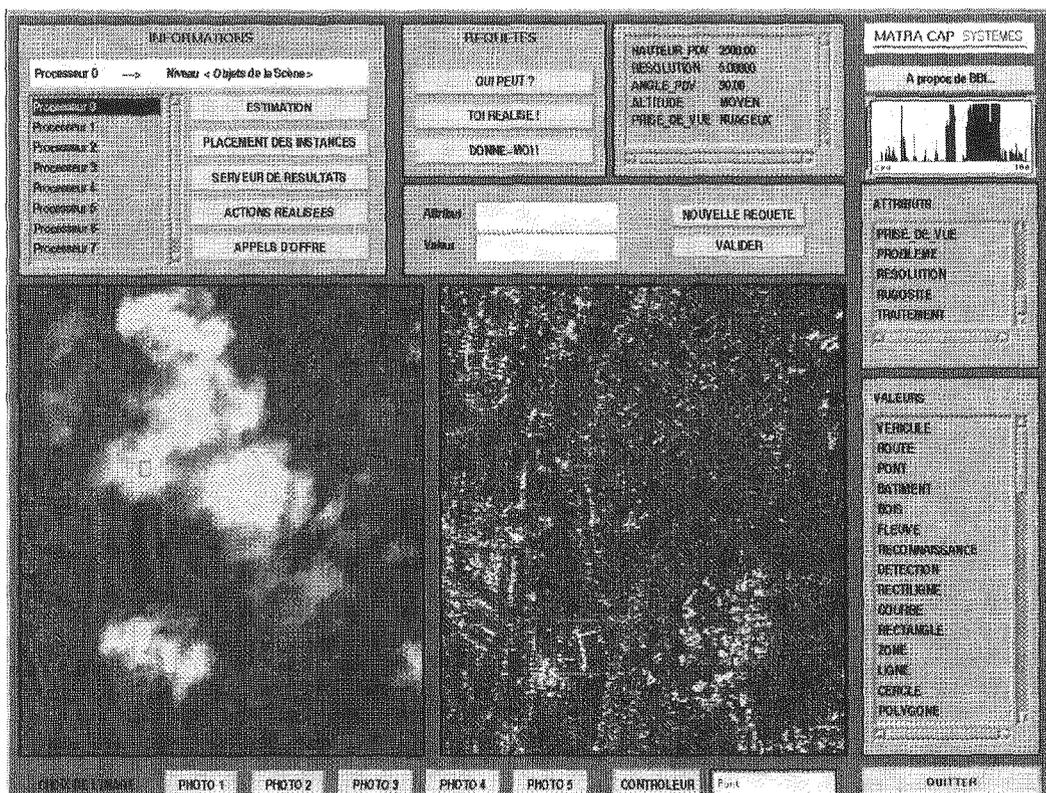


Figure 5. – Détection de ponts (conditions météorologiques nuageuses).

4. perspectives

4.1. le temps réel

ADAGAR est tout à fait adapté aux architectures multiprocesseurs. Il permet ainsi de paralléliser l'application. Mais il ne possède pas de mécanismes temps réel. Les extensions pressenties pour l'atelier ADAGAR concernent notamment les points suivants :

- non monotonie par la remise en cause de résultats partiels,
- réactivité du système à des informations d'entrée asynchrones,
- garantie des temps de réponse (en phase d'appel d'offre et d'exécution) par contrôle du temps de réponse des agents,
- raisonnement adaptatif en privilégiant certains agents par rapport à d'autres en fonction du temps imparti à la résolution du problème.

Nous envisageons de réaliser certaines de ces extensions en intégrant à ADAGAR une partie de nos acquis dans les domaines de l'intelligence artificielle temps réel et des systèmes multi-agents. Nos activités sont plus particulièrement basées sur les outils KASTOR [Bru92a], générateur de systèmes experts temps réel, et ATOME-TR [Bru92b], outil de maquettage d'applications multi-agents temps réel.

L'intégration de KASTOR au niveau des agents de ADAGAR permettra de contrôler les temps de réponse. KASTOR possède en effet un compilateur de règles qui construit un réseau décisionnel à partir d'une base de règles. La conséquence directe de cette transformation est que les temps de réponse du système sont bornés. La conséquence indirecte est que la complétude et la cohérence de la base de règles sont vérifiées, atout non négligeable en environnement critique. KASTOR intègre aussi un mécanisme de raisonnement dans le temps.

Les mécanismes de planification réactive utilisés par ATOME-TR seront également étudiés pour doter ADAGAR de capacités de raisonnement adaptatif.

Grâce à l'ajout de fonctionnalités temps réel, ADAGAR pourra être utilisé sur un spectre d'applications plus vaste (analyse d'images en temps contraint, détection de cibles) et permettra aux développeurs d'applications « temps critique » de profiter pleinement des capacités actuelles.

4.2. apports au niveau traitement d'images

En ce qui concerne l'application du système au traitement d'images, celui-ci apporte des solutions, tant en segmentation qu'en interprétation d'images. Comme pour les recherches présentés en [Clo94], l'approche générique permet de présenter ces travaux

comme une base de qualification et de validation de problèmes généraux en traitement d'images.

Les problèmes actuels sont d'une part, de choisir parmi une panoplie de traitements existants [Coc95], les traitements les plus adaptés à la nature de l'image, au problème à résoudre et d'autre part, d'assurer une coopération entre ces différents traitements.

Le système BBI permet la prise en compte de plusieurs méthodes bas niveau et fournit donc un outil puissant pour la segmentation coopérative :

- enchaînement de filtrage, extraction de contours, fermeture de contours, suppression de petites régions;
- coopération extraction de contours / extraction de régions.

On a vu également qu'il est adapté à la coopération haut-niveau / bas-niveau et on peut, par conséquent, améliorer la segmentation après une phase de détection d'objets ou de focalisation. Le système permet en outre de paramétrer au mieux les traitements.

La coopération entre plusieurs traitements (de même niveau ou de niveaux différents) passe par une fusion de résultats issus ou non de la même image de départ. BBI est un système opérationnel mettant en œuvre une des méthodes proposés par les théoriciens de la fusion et permettant de tester les autres [Blo94].

5. conclusion

L'environnement ADAGAR a été défini à partir des contraintes imposées dans le projet BBI. Son architecture est basée sur un ensemble de concepts originaux inspirés du modèle du blackboard (hiérarchisation des agents, contrôle distribué). Cette approche fournit, dans le domaine de l'aide à la photo-interprétation, un modèle pertinent de coopération mixant la connaissance symbolique apportée par l'expert aux algorithmes existants. Elle étend aussi ses propriétés aux architectures multiprocesseurs afin de répondre aux contraintes de temps de réponse imposées dans un système opérationnel, et aux puissances de calcul élevées dues aux traitements d'images.

L'architecture fonctionnelle d'ADAGAR permet **de distribuer la connaissance** au travers d'agents cognitifs et d'une **organisation hiérarchique** de ces derniers en différents niveaux d'abstraction liés au domaine de l'application. La résolution d'un problème se fait au travers d'une coopération entre agents établie dynamiquement grâce à un mécanisme d'appels d'offre.

Un tel atelier facilite la mise en œuvre de tout mécanisme visant à manipuler des données issues de sources ou de traitements différents quel que soit le niveau d'abstraction (agrégation aux niveaux les plus abstraits et corrélation, coopération, fusion, focalisation, etc).

Le domaine d'application des systèmes multi-agents est très vaste. C'est pourquoi le modèle présenté au travers de ADAGAR possède des caractéristiques permettant une création aisée d'agents et un fonctionnement sur une large classe d'architectures

multi-processeurs. De plus, cette modélisation est orientée traitement et le système fonctionne uniquement grâce à l'expertise recueillie pour chaque traitement de l'application.

Sur la base de ces acquis, les axes de recherches actuellement suivis concernent à court terme, l'augmentation de la typologie des agents et l'enrichissement de la panoplie d'outils fournie au développeur, mais aussi à plus long terme, comme nous l'avons mentionné au paragraphe 4, l'utilisation de cet environnement pour appréhender les problèmes difficiles de traitement d'images à savoir le paramétrage de traitement, l'évaluation et la fusion de résultats.

BIBLIOGRAPHIE

- [Ban91] J. P. Banâtre, M. Banâtre, *Les systèmes distribués : expérience du projet gothic*, InterEditions, Paris, 91.
- [Bau89] O. Baujard, C. Garbay, « KISS : Un système de vision multi-agents », *7ème RFIA*, Paris, 89, 89-98.
- [Blo94] I. Bloch, « Information combination operators for data fusion : a comparative review with classification », *rapport interne ENST 94DO13*.
- [Bru92a] Sy. Brunessaux, « Kastor : techniques et méthode pour le développement de systèmes experts temps réels embarqués », *Génie Logiciel et Systèmes Experts*, n°28, Editions EC2, Septembre 1992.
- [Bru92b] St. Brunessaux, F. Charpillat, J. P. Haton, J. C. Le Mentec, « Une architecture à bases de connaissances multiples & orientée temps réel », *Génie Logiciel et Systèmes Experts*, n°28, Editions EC2, Septembre 1992.
- [Clo94] R. Clouard, « Raisonement incrémental et opportuniste appliqué à la construction dynamique de plans de traitement d'images », *Thèse de doctorat*, Université de Caen, Février 1994.
- [Coc95] J. P. Cocquerez, S. Philipp, « Traitement d'image : filtrage et segmentation », *Editions MASSON 1995* (à paraître).
- [Dem68] A. P. Dempster, « A Generalization of Bayesian Inference », *J. Roy. Statist. Soc., Series B*, vol. 30, pages 205 à 247.
- [Gar89a] C. Garbay, S. Pesty, « MAPS : Un système multi-agents pour la résolution de problèmes », *7ème RFIA*, Paris, 89, p. 355-368.
- [Gar89b] P. Garnesson, G. Giraudon, P. Montesinos, « MESSIE : Un système multi-spécialistes en vision. Application à l'interprétation en imagerie aérienne », *7ème RFIA*, Paris, 89, p. 817-831.
- [Gon87] Y. Gong, J. P. Haton, « Une société de spécialistes à niveaux multiples pour l'interprétation de signaux », *6ème RFIA*, Antibes, 87, p. 245-257.
- [Hât91] J. P. Hâtton, « Le raisonnement en intelligence artificielle - Modèles, techniques et architectures pour les systèmes à bases de connaissance », *Editions InterEditions 1991*.
- [Hay85] B. Hayes-Roth, « A blackboard architecture for control », *Artificial Intelligence*, 26, 85, p. 251-320.
- [Laâ88] H. Laâsri, B. Maître, J. P. Haton, « Organisation, coopération et exploitation des connaissances dans les architectures de blackboard : ATOME », *8èmes journées internationales « Les systèmes experts et leurs applications »*, Avignon, 06/88, p. 371-390.
- [Lef93a] V. Lefèvre, Y. Pollet, « SADGE : Proposition de système pour l'aide à la gestion de l'environnement », *AFCET'93*, Versailles Juin 93.
- [Lef93b] V. Lefèvre, Y. Pollet, « BBI : un système multi-agents d'aide à la photo-interprétation », *13ème journées internationales « Intelligence Artificielle, Systèmes expert et langage naturel »*, Avignon, Mai 1993, p. 473-482.
- [Lef94] V. Lefèvre, « Système multi-agents d'aide à la photo-interprétation sur architecture distribuée », *Thèse de doctorat*, Université Paris XI, Novembre 1994.
- [Mat89] T. Matsuyama, « Expert systems for image processing : knowledge-based composition of image analysis processes », *CVGIP*, 48, 89, pages 22-49.
- [Nii86] H. P. Nii, « Blackboard systems, PART 1 : The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, PART 2 : Blackboard Application Systems and a Knowledge Engineering Perspective, Artificial Intelligence », 7(2), 86, p. 38-53 et, 7(3), 86, p. 82-106.
- [Occ91] M. Occello, C. Chaouiya, M. C. Thomas, « Un modèle d'architecture de tableau noir parallèle pour le pilotage temps réel en robotique mobile », *8ème congrès RFIA*, Lyon, 1991, pages 443 à 448.
- [Pit90] J. Pitrat, « Métaconnaissance - Futur de l'Intelligence Artificielle », *Editions Hermès*, 1990.
- [Sha76] G. Shafer, « A Mathematical Theory of Evidence », *Princeton Univ. Press*, Princeton.
- [Smi80] R. G. Smith, « The Contract-Net protocol : high level communications and control in a distributed problem solver », *IEEE Transactions on computers*, Vol.n C-29, n°12, 1980.
- [Vel87] H. Velthuisen, B. J. Lippolt, J. C. Vonk, « A parallel blackboard architecture », *3rd International Expert Systems Conference*, London 06/1987, pages 487 à 493.

Manuscrit reçu le 19 février 1996.

LES AUTEURS

Valéry LEFEVRE



Valéry Lefèvre est docteur spécialisé en Intelligence Artificielle au sein de la division Préparation de Missions de MATRA CAP Systèmes. Suite à un deuxième cycle en informatique obtenu à l'université de Caen et un D.E.A. d'électronique, option traitement de l'information, à l'université de Paris XI (Orsay), il a soutenu, en 1994, son doctorat d'Electronique, option traitement des images, à l'université d'Orsay, ayant pour thème la réalisation d'un système multi-agents d'aide

à la photo-interprétation sur architecture multi-processeur. Il travaille actuellement sur les problèmes de la déconfliction automatique et de la planification dans les systèmes de préparation de missions.

Yann POLLET



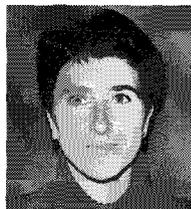
Yann Pollet est responsable du Département Informatique Avancée de la société MATRA CAP Systèmes. Ingénieur de l'Ecole Nationale Supérieure de Techniques Avancées (ENSTA), il rejoint MATRA en 1981 et travaille dans le domaine de la modélisation de systèmes et la simulation. Il est également diplômé de l'IAE et titulaire d'un doctorat en informatique. Présent à MATRA CAP Systèmes depuis 1989, il s'intéresse à l'architecture logicielle des Systèmes d'Information et

de Communication et aux bases de données. Yann Pollet participe à plusieurs projets européens Esprit et Euclid en tant qu'expert technique. Il assure d'autre part un enseignement de bases de données à l'INSA et à la Faculté des Sciences de Rouen.

Sylvie PHILIPP

Sylvie Philipp a enseigné les mathématiques dans le secondaire de 1977 à 1984 après une maîtrise et un CAPES de Mathématiques. Elle a obtenu son doctorat d'informatique de l'Université Paris VI, option traitement des images en 1988 avec une thèse effectuée à E.D.F. portant sur l'analyse de texture appliquée aux radiographies de pièces métalliques. Elle est actuellement professeur et enseigne à l'ENSEA et dans le D.E.A. Traitement des Images et du Signal de l'université de Cergy. Ses activités de recherche se font dans le laboratoire de l'ETIS, à l'ENSEA, et portent sur les traitements bas-niveaux (segmentation et analyse de texture), sur l'interprétation des images et sur les systèmes de vision.

Sylvie BRUNESSAUX



Sylvie Brunessaux est ingénieure spécialisée en Intelligence Artificielle au sein du département Etudes Systèmes et Informatique Avancée de MATRA CAP Systèmes. Elle s'est intéressée à l'intégration et la sûreté de fonctionnement des systèmes experts en environnement temps réel au travers des techniques de compilation de base de règles. Actuellement, elle travaille essentiellement sur l'apport des techniques d'I.A. et plus particulièrement des systèmes multi-agents dans les systèmes de fusion de données.