

Rapport INRIA 1994 — Programme 3

Environnements Intelligents de résolution de
problèmes pour des systèmes autonomes

ACTION ORION

3 mai 1995

ACTION ORION

Environnements Intelligents de résolution de problèmes pour des systèmes autonomes

Localisation : *Sophia-Antipolis*

Mots-clés : apprentissage automatique (7), base de connaissances (8), classification (10), conception par objets (6), génie logiciel (4), intelligence artificielle (1), intelligence artificielle distribuée (10), interprétation d'image (1, 9, 11), pilotage de programmes (1, 4, 5, 7), planification (4–6), programmation par objets (6), réactivité (5), résolution de problème (4), réutilisation de logiciel (4), séquence d'images (9, 11), système à base de connaissances (1, 7, 10, 11), temps réel (11), traitement d'image (11).

1 Composition de l'équipe

Responsable scientifique

Monique Thonnat, DR Inria

Secrétaire

Françoise Gouttier, TR Inria

Agnès Cortell, SAR Inria, depuis le 24 octobre 1994

Personnel Inria

Sabine Moisan, CR

Ingénieurs experts

Chandra Shekhar, ingénieur expert, jusqu'au 30 novembre 1994
Hany Tolba, ingénieur expert

Chercheurs extérieurs

Frank van Harmelen, Docteur, Université d'Amsterdam
Jean-Paul Rigault, Professeur, Université de Nice Sophia-
Antipolis

Chercheurs doctorants

François Brémond, ENS Lyon, Université de Nice Sophia-
Antipolis
John Van den Elst, boursier HCM, Université de Nice Sophia-
Antipolis
Jean-Christophe Ossola, boursier MESR, Université de Nice
Sophia-Antipolis
Régis Vincent, boursier INRIA, Université de Nice Sophia-
Antipolis

Stagiaires

Patrick Ladugue, stagiaire Mastère CERAM, du 1^{er} avril au
30 septembre 1994
David Palermo, stagiaire Mastère CERAM, du 1^{er} avril au
30 septembre 1994
Maria del Mar Marcos, stagiaire COMETT, en janvier 1994
et de juillet à décembre 1994
Alain Philippe, stagiaire Supelec, de avril à juin 1994

2 Présentation du projet

L'action ORION, créée en octobre 1992, est rattachée au projet SHERPA.

Le domaine de recherche d'ORION concerne les systèmes à base de connaissances autonomes. Plus précisément, ORION a pour objectif de faciliter la construction de systèmes intelligents automatiques et adaptatifs, c'est à dire des systèmes incorporant explicitement une expertise (intelligents), fonctionnant sans intervention humaine pour les prises de décisions (automatiques) et ayant des capacités de réaction vis à vis des changements de leur environnement (adaptatifs). Un domaine d'application privilégié pour de tels systèmes est le traitement de signal

et d'image; il existe également des applications intéressantes en calcul scientifique.

Parmi les applications qui nécessitent un fort degré d'autonomie, nous avons identifié d'une part les problèmes d'automatisation de l'utilisation de logiciels complexes, et d'autre part les problèmes d'interprétation automatique de données.

L'automatisation de l'utilisation de logiciels complexes nécessite de planifier les traitements et de contrôler l'exécution de codes de calcul d'une bibliothèque. Par la suite nous utiliserons le terme **pilotage automatique de programmes** pour désigner ce type de problèmes.

Dans de nombreuses applications l'utilisateur ne peut pas interpréter les résultats des logiciels de traitements de données soit parce qu'il n'en est pas capable (manque de compétence), qu'il n'en a pas le temps (saturation de ses capacités) ou parce qu'il n'est pas présent (systèmes embarqués). Ceci correspond à un sous-problème de la perception, celui de l'interprétation des résultats des traitements d'image ou de signal. Par la suite nous utiliserons le terme d'**interprétation automatique de données**.

En pilotage de programmes nous avons approfondi les points suivants :

- la modélisation de la connaissance pour le pilotage de programmes;
- le pilotage temps-réel de programmes de perception;
- la constitution d'une bibliothèque pour la planification;
- le raffinement de bases de connaissances;
- la vérification de bases de connaissances.

En interprétation de données nous avons démarré les études suivantes :

- l'interprétation de séquences d'images;
- la coopération de systèmes à base de connaissances
- la reconnaissance de scénarios dans une application de vidéo surveillance.

Cette année a vu l'aboutissement de nos travaux sur le projet européen Eureka Prometheus. Le véhicule Prolab2 (Peugeot 605) a été présenté lors de démonstrations temps-réel sur le circuit automobile de Morfontaine, conjointement aux trente autres véhicules Prometheus. Un nouveau champ d'applications en vidéo surveillance s'est ouvert cette année.

3 Action de recherche

3.1 Pilotage de programmes

3.1.1 Modélisation de la connaissance pour le pilotage de programmes

Participants: John van den Elst, Frank van Harmelen, Monique Thonnat, Sabine Moisan

Nous nous intéressons à la modélisation des connaissances pour le pilotage de programmes. Ce problème comporte deux phases, la planification des programmes pour atteindre un but pré-défini, et le contrôle d'exécution des programmes planifiés. Pour la première phase nous cherchons à définir une représentation de la connaissance permettant de composer une solution (planification). Cette connaissance porte sur les éléments nécessaires pour sélectionner et ordonnancer les programmes d'une bibliothèque de programmes, initialiser leurs arguments et paramètres (spécialisation) et les intégrer. Dans notre domaine (le domaine des systèmes autonomes de perception) le contrôle d'exécution est aussi un point d'intérêt important. Nous cherchons à définir la connaissance nécessaire pour observer et évaluer des résultats intermédiaires, et si il y a un problème, adapter les composants (une spécialisation alternative par exemple) ou re-planifier une autre solution. Nous avons étudié ces problèmes sous quatre points de vue : le système existant OCAPI, le domaine de la réutilisation de composants logiciels (génie logiciel), l'étude d'autres approches pour ce problème et la planification.

Point de départ: OCAPI Notre point de départ a été OCAPI (Outil de Contrôle Automatique de Procédures Images). Nous avons proposé une description fonctionnelle des connaissances contenues dans les concepts et les mécanismes d'OCAPI, en utilisant le modèle d'expertise de KADS et la notation CML (Conceptual Modeling Language). Ces travaux sont présentés dans [8].

Réutilisation de composants logiciels Dans le domaine du génie logiciel le thème des composants réutilisables a des liens étroits avec la planification de programmes. Ces liens concernent uniquement la phase de planification. Nous avons pu comparer notre modèle avec les modèles décrits par Biggerstaff, Krueger ou Weide pour les points suivants : la *représentation à différents niveaux d'abstraction des données* qui sont

manipulées par les composants, la *description* des composants, la *structuration* des composants dans une hiérarchie basée sur leur fonctionnalité et le niveau d'abstraction des paramètres et les *sélection*, *spécialisation*, et *intégration* des composants.

Approches existantes Le troisième point de recherche a été l'étude des autres approches du problème de supervision de programmes, en particulier SCARP (Willamowski), TRAM (Chaillot), VSDE (Bodington) et IAMI (Forte). Ces études ont montré la plus ou moins grande dépendance du modèle vis à vis du domaine d'application.

3.1.2 Pilotage temps-réel de programmes de perception

Participants : Chandra Shekhar, Monique Thonnat, Sabine Moisan

Le but de ce travail était le développement de Planete, un générateur de systèmes à base de connaissances dédié à la supervision de programmes en temps-réel. Un système à base de connaissances a été réalisé, pour la détection d'obstacles, dans le cadre du projet Eureka PROMETHEUS. Planete est une implémentation temps-réel embarquée de l'architecture proposée dans [1]. Ce travail a été réalisé en collaboration avec David Hutber du projet PASTIS, qui a développé un module de fusion multi-capteur pour la même application. Le groupe PROART France, du projet PROMETHEUS a réalisé un véhicule démonstrateur, nommé Prolab2. Ce véhicule est équipé d'un système de guidage embarqué (ou copilote), afin d'aider le conducteur à éviter les obstacles dans des situations de conduite réalistes. Le copilote est un système temps-réel intelligent qui traite les informations issues des divers capteurs montés sur le véhicule (5 caméras CCD, 1 caméra stéréo linéaire et un télémètre couplé avec une caméra, ainsi que différents capteurs proprioceptifs). Le copilote utilise ces informations pour interpréter la situation du véhicule et communique cette analyse au conducteur, via une interface. Afin de fournir une interprétation fiable et utile dans le contexte complexe d'un trafic routier très évolutif, les traitements bas-niveau des données (réalisées par divers programmes de perception) doivent être adaptés et optimisés, en fonction des différents contextes routiers et des différentes informations disponibles. C'est la tâche du module de supervision de programmes Planete qui est incorporé dans le copilote.

Le module de supervision de programmes contient toute la connaissance de résolution de problème correspondant aux modules de perception.

Cette connaissance porte sur la planification, l'ordonnancement et le contrôle d'exécution. En Planete, les éléments de base de la planification sont les *actions*, qui sont des manipulations faites sur les modules de perception. Il y a quatre types d'actions possibles sur les modules : activation/dés-activation, changement de mode de fonctionnement, changement de réglage de paramètre et sélection de données. L'ordonnancement devient non trivial lorsqu'il faut exécuter en parallèle deux ou plusieurs modules, qui utilisent les mêmes ressources. Il y a trois types de conflits d'accès possibles : capteur partagé, programme partagé ou CPU partagé. La connaissance sur le contrôle d'exécution est exprimée sous forme de critères d'initialisation de paramètres, d'évaluation de performance et d'ajustement de paramètres. Pour des raisons d'efficacité dues à la nature temps-réel de l'application, l'évaluation des résultats en Planete est minimale, et réservée à des réglages occasionnel plutôt qu'à une optimisation continue des traitements. Pour les mêmes raisons la base de connaissances et le moteur de Planete ne sont pas complètement indépendants. L'architecture de Planete a été implémentée en langage C++, sous le noyau temps-réel VxWorks. Le matériel utilisé est un processeur Motorola 68040 monté sur une carte MVME167. Dans l'application Prolab2, Planete supervise trois modules de perception : la détection d'obstacles 3-D utilisant le télémètre, la détection d'obstacles par morphologie mathématique et la détection d'obstacles par segmentation de mouvement. Planete et les modules de perception sont conçus pour fonctionner sur des machines séparées (i.e. chacun de ces modules a sa propre architecture matérielle associée). La communication entre Planete et les modules de perception se fait grâce à des zones de mémoire partagées. Un mécanisme d'exclusion mutuelle par sémaphores gère les accès à la mémoire partagée.

Bien que Planete ait été développé dans le contexte d'une application spécifique (le projet Prolab2), ces principes sont généraux et applicables à différentes situations requérant l'utilisation autonome de programmes complexes en temps-réel. Des détails sur la conception et le fonctionnement de Planete sont présentés dans [5, 3, 6].

3.1.3 Bibliothèque pour la planification

Participants : Patrick Laduguie, David Palermo, Sabine Moisan, Jean-Paul Rigault, Régis Vincent

PALMER, Planificateur d'Activités Logicielles, Modulaires et Réutilisable, est une bibliothèque de classes C++ qui propose les structures de base et les instructions élémentaires pour développer des moteurs de planification.

La phase de spécification a consisté en une analyse du domaine de la planification, ce qui a permis d'identifier les entités fondamentales intervenant lors de la planification. La méthode OMT (Object Modeling Technique) et l'atelier de conception objet Paradigm+ ont été utilisés pendant la phase de conception.

Le choix du langage de programmation C++ pour l'implantation a été retenu en raison de sa grande diffusion (équipes de recherche, industrie, enseignement), de sa normalisation par le comité ANSI C++, de contraintes dues à l'implication de l'équipe ORION dans des projets européens et aux possibilités de développement d'applications temps réel.

3.1.4 Raffinement de bases de connaissances

Participants : Régis Vincent, Sabine Moisan, Monique Thonnat

Nous avons réalisé un système d'apprentissage, dont le but est de corriger ou de raffiner les bases de connaissances construites avec Ocapi. Dans un premier temps, nous nous sommes intéressés au raffinement de la connaissance sur la paramétrisation des programmes avant leur exécution. Le raffinement de bases de connaissances consiste à corriger ou à compléter une base. Dans notre cas, nous souhaitons raffiner la base de manière dynamique, c'est à dire d'une utilisation à l'autre. Pour cela, nous avons développé un algorithme particulier combinant l'apprentissage par explications (Explanation-Based Learning) et l'apprentissage par cas (Case-Based Learning) (pour plus de détails voir [9]). En résumé, cet algorithme se base sur le mécanisme de récupération d'erreurs pour apprendre. En effet, dans un système à base de connaissances réalisé avec Ocapi, il existe des mécanismes de récupération d'erreurs qui permettent au système de proposer de nouvelles solutions lorsque des erreurs surviennent. Dans la phase de paramétrisation de programmes le rôle de ce mécanisme est de permettre d'initialiser et de réajuster les valeurs des paramètres d'entrée des programmes. De même, lors de la phase de planification, un mécanisme de retour arrière permet de proposer d'autres choix, en cas de plan incorrect. Dans un premier

temps, nous nous sommes limités à une correction de la connaissance sur la paramétrisation des programmes. L'expert fournit deux types de connaissances pour cette phase : des connaissances sur la façon d'initialiser les paramètres et des connaissances sur la façon d'ajuster ces valeurs, si les valeurs d'initialisation se révèlent incorrectes. Le mécanisme d'ajustement d'Ocapi utilise ces deux types de connaissances et propose de nouvelles valeurs lorsque les valeurs d'initialisation n'ont pas donné de bons résultats. Il est assez difficile pour l'expert d'exprimer les différentes valeurs des paramètres. Souvent il ne connaît que quelques valeurs particulières, les plus fréquentes. Il est extrêmement difficile de prévoir toutes les valeurs possibles, car celles-ci dépendent des données. Dans certains cas, la corrélation est très complexe entre les données et les valeurs des paramètres et l'expert ne sait pas toujours l'exprimer. Cette partie de la connaissance est donc souvent incorrecte et imprécise, et chaque erreur d'initialisation coûte une ou plusieurs nouvelles exécutions du même programme. Lorsque le mécanisme de récupération d'erreurs a fonctionné, le système possède une solution correcte, c'est à dire la ou les valeurs correctes pour les paramètres d'entrée. Grâce à cette solution, le système d'apprentissage peut corriger la base de connaissances.

Réalisation Le système d'apprentissage a été réalisé en Le_Lisp, et fonctionne avec Ocapi. Actuellement nous l'avons testé sur la base de traitement de scène routière PROMETHEE. Ces premiers tests permettent de corriger la base de manière significative, car sur une séquence de 10 images, nous passons de 20 réajustements à 4 et le temps machine passe de 200 secondes à 139 soit un gain de 44%.

3.1.5 Vérification de bases de connaissances

Participants : Maria del Mar Marcos, Sabine Moisan

Les systèmes à base de connaissances générés par les noyaux que nous développons sont des systèmes à représentation hybride, qui combinent des règles de productions et des objets structurés (frames). Au moment de la création d'une base de connaissance par un expert, il faut pouvoir assurer à l'expert un minimum de cohérence dans sa base. C'est le rôle d'un outil de validation et de vérification. Dans un premier temps, un état de l'art en vérification et validation a été réalisé. Puis nous nous sommes intéressés à l'aspect vérification syntaxique pour assurer que

la base en cours d'évolution est correcte. La vérification de bases de connaissances se préoccupe de la détection d'anomalies, en se basant sur la syntaxe et la sémantique des formalismes de représentation. Un vérificateur de cohérence syntaxique des règles et de leurs relations avec les objets a été développé. Nous sommes en cours de spécification de la partie validation sémantique. En particulier nous envisageons pour cela de nous appuyer sur le modèle du pilotage de programmes.

3.2 Interprétation

3.2.1 Interprétation de séquences d'images

Participants : François Brémond, Hany Tolba, Monique Thonnat, Sabine Moisan

Cette année nous avons commencé une nouvelle étude en interprétation d'images concernant des séquences d'images. L'interprétation de séquences d'images a pour objectif, en ce qui nous concerne, de donner un sens à une scène, à partir d'images fournies par une caméra monoculaire et fixe.

Il s'agit tout d'abord d'effectuer le repérage et la classification des objets de la scène, en utilisant des techniques de traitement d'images; puis d'analyser leurs comportements par des méthodes d'intelligence artificielle.

L'objectif de cette recherche est l'élaboration d'un système à base de connaissances générique, visant à l'interprétation de séquences d'images dans un environnement dynamique et non structuré. Le cadre de ce sujet est spécialement axé sur l'étude des comportements d'objets non rigides et plus particulièrement sur ceux ayant trait aux activités humaines. Le fait de devoir traiter des objets non rigides impose d'obtenir des données de traitement d'images de bonne qualité. Dans ce but, nous allons utiliser les informations relatives au contexte (ou connaissances a priori) de la scène. La modélisation de ce contexte, à l'aide par exemple d'une représentation analogique¹ de l'espace, est un premier point de recherche. En second lieu, la nature des objets étudiés (objets de forme complexe) implique de gérer des données incomplètes, incertaines et parfois erronées.

¹Dans ce texte, le terme analogique est utilisé lorsque la représentation du monde conserve la structure inhérente du monde représenté (voir Neumann 90, Howarth 92)

Leur abstraction en données symboliques permet d'établir les relations spatio-temporelles nécessaires à la reconnaissance des comportements.

Les résultats actuels sont d'une part un état de l'art sur ce domaine et d'autre part un premier modèle d'interprétation.

Le domaine d'applications privilégié est celui de la vidéo surveillance.

3.2.2 Coopération de systèmes à base de connaissances

Participants : Jean-Christophe Ossola, Monique Thonnat

La résolution de manière autonome d'un problème de reconnaissance d'objet naturel (zooplancton, galaxie...) se révèle complexe de part la double nature des connaissances requises ; la vision pour le pilotage d'algorithmes de traitement d'images, et le domaine de l'objet à reconnaître pour l'interprétation.

Classic et Ocapı sont deux générateurs de systèmes à base de connaissances évolués, développés dans notre équipe. Ils sont dédiés à des tâches spécifiques, pilotage d'algorithmes pour l'un et interprétation pour l'autre. Nous nous sommes orientés vers une coopération de ces deux générateurs pour définir et tester une architecture distribuée. Après avoir utilisé successivement Sophtalk et Chooe, deux langages de communication, nous avons arrêté notre choix sur celui qui nous semblait le plus approprié à nos besoins : Chooe. Ce langage nous a permis de réaliser une architecture distribuée bien qu'Ocapı et Classic aient été conçus sans perspective de communication. Le choix d'un nouveau langage de communication (Chooe) pour la réalisation de la connection entre les deux systèmes se révèle fort intéressant, du fait de sa simplicité d'utilisation, mais surtout du fait des capacités supplémentaires qu'il apporte en standard au système, notamment :

- le choix entre plusieurs modes de communication ;
- la possibilité de création de processus distants ;
- la gestion d'objets informatiques structurés.

Une première application sur des images de zooplanctons a permis de tester l'architecture ainsi définie. Cette application a mis en jeu deux systèmes à bases de connaissances (SBC). Le premier, que nous avons entièrement conçu, est dédié au traitement d'images de zooplanctons (Prozoo). Le second, qui avait été développé par M.H. Gandelin et que nous avons modifié, effectue le classement de zooplanctons (Clazoo). Il

existait dans Clazoo des appels à des procédures de traitement d'images. Ces appels ont été avantageusement remplacés par une coopération avec Prozoo.

Nous avons ensuite testé cette architecture sur des images de galaxies. Dans un premier temps, nous avons augmenté les capacités d'un SBC existant, développé sous Ocapi et nommé Progal. Ce SBC réalise le traitement nécessaire à l'extraction de paramètres pertinents pour le classement de galaxies. Dans un second temps, nous avons effectué la connection effective de Progal avec le SBC développé sous Classic appelé Sygal qui opère le classement de galaxies. La reconnaissance des galaxies se fait dorénavant de manière complètement automatique depuis le traitement des images et l'extraction de paramètres jusqu'au classement proprement dit. Comme perspectives, nous souhaitons valider l'architecture proposée pour l'application sur le classement de galaxies. Notre prochain objectif est d'augmenter le degré d'interactivité du système en modifiant davantage les SBC actuels. Pour cela, de nouveaux traitements doivent être développés. Ces traitements permettront l'extraction de paramètres spécifiques à une classe d'objet. Certains objets qui étaient jusqu'à présent impossibles à classer vont pouvoir l'être grâce à la capacité de communication entre des systèmes spécialisés en traitement d'image et en classement. De plus, la reconnaissance des galaxies gagnera en précision.

3.2.3 Passwords : reconnaissances de scénarios dans une application de vidéo surveillance

Participants : Hany Tolba, Monique Thonnat, François Brémond, Sabine Moisan

Cette recherche porte sur le développement d'un outil de raisonnement pour la reconnaissance de scénarios et de situations dangereuses. Ceci est mené dans le cadre du projet européen ESPRIT3, PASSWORDS. Le but du projet PASSWORDS est le développement d'un système générique pour la vidéo surveillance. Ce système doit fournir une assistance aux agents de sécurité afin de reconnaître différents scénarios et différentes situations jugées dangereuses. Les locaux surveillés sont des locaux externes et internes. Comme exemple de situation dangereuse, nous pouvons mentionner la prévention contre le vol de voitures sur un parking de supermarché ou contre le vandalisme sur un quai du métro.

La difficulté des travaux menés dans le cadre de ce projet tient à ce que nous nous intéressons aux comportements humains et aux objets non-rigides qui évoluent dans un environnement dynamique et non structuré. La solution proposée par le consortium du projet est fondée sur une coopération du module de traitement d'image et d'un module d'intelligence artificielle. Notre rôle est la conception et la réalisation du module d'intelligence artificielle. La phase de traitement d'image consiste à extraire les différents indices sur une image captée par une caméra. La phase d'intelligence artificielle consiste à reconnaître les différentes situations et les différentes relations spatio-temporelles entre les objets qui existent dans l'image et qui correspondent à des objets dans le monde réel. L'architecture prévue pour le module I.A. est une architecture de système à base de connaissances temps-réel. Les contraintes d'implantation sont aussi bien des contraintes matérielles que des contraintes logicielles. D'une part, nous allons travailler sur des cartes de traitement de signal (DBV42 de LSI munie de la puce TMS 320C40 de Texas Instruments) et d'autre part nous allons utiliser le langage *C* sous le système d'exploitation temps-réel *Virtuoso*.

4 Actions industrielles

4.1 Esprit PASSWORDS

Participants : Hany Tolba, Monique Thonnat, Sabine Moisan, François Brémond

Depuis avril 1994, l'action *ORION* participe au Projet Esprit3 R&D *PASSWORDS* avec des partenaires scientifiques et industriels italiens, belges et français. Nous intervenons sur la durée totale du projet et dans ses différentes phases de conception et d'implantation. Le but de *PASSWORDS* est le développement d'un système générique de vidéo surveillance de locaux. Le système final fonctionnera en temps réel et aura une architecture parallèle ; son objectif est d'assister un opérateur humain dans la reconnaissance de situations dangereuses. La contribution de l'action *ORION* porte sur l'interprétation de données pour la reconnaissance de scénarios de danger. Deux applications pilotes sont prévues : la surveillance de centres de grande distribution et la surveillance de stations de métro.

Nos partenaires industriels sont : Vigitec (Belgique), SRWT (Belgique), Auchan (France et Italie) et SEPA (Italie). Nous avons également deux autres partenaires scientifiques : CRIF (Belgique) et l'université de Gènes (Italie).

4.2 Eureka PROMETHEUS

Participants : Chandra Shekhar, Monique Thonnat

L'objectif du projet Eureka Prometheus, qui réunit les 14 constructeurs d'automobiles européens et une centaine d'industriels et de laboratoires de recherches (en Allemagne, Italie, Angleterre, Suède et France) est l'amélioration de la sécurité routière. Ce projet a démarré en 1987 et est dans sa dernière phase (1992-1994).

Lors de cette dernière phase nous avons développé avec nos partenaires français le véhicule démonstrateur Prolab2 d'assistance au conducteur par détection des obstacles. Les partenaires sont : Heudiasyc Compiègne, Inria Sophia (Pastis), Irisa Rennes (Temis), LASMEA Clermont-Ferrand, LIFIA Grenoble (Sharp), Ecole des Mines de Paris (CAOR et CMM) et Université de Lille Ce véhicule a été présenté en octobre 94 conjointement aux 30 autres véhicules démonstrateurs de Prometheus. La contribution d'ORION a concerné le pilotage temps-réel des programmes de perception.

5 Actions nationales et internationales

5.1 Activités internationales

5.1.1 Coopérations scientifiques

- Nous coopérons avec l'équipe du Prof. Wielinga de l'université d'Amsterdam, sur la modélisation de connaissances pour le pilotage de programmes ; cette coopération est effectuée dans le cadre du coencadrement de la thèse de J. van den Elst par F. van Harmelen et M. Thonnat et fait l'objet d'un financement européen HCM.
- Nous coopérons avec l'équipe du Prof. del Pobil, responsable du groupe IA et Robotique de l'université de Jaume I, à Castellón (Espagne) sous la forme d'accueil d'étudiant, (M. Marcos), dans

le cadre du programme européen COMETT. Nous avons récemment répondu à un appel d'offre d'action intégrée PICASSO sur le thème de la validation et de la vérification de systèmes à base de connaissances autonomes.

5.1.2 Comité de programme

M. Thonnat a fait partie du comité scientifique de la conférence internationale The World of Galaxies II.

5.1.3 Comités de rédaction

M. Thonnat est relecteur des revues : “Pattern Recognition Letters”, “International Journal of Pattern Recognition and Artificial Intelligence” (IJPRAI) et “Astrophysical Letters and Communications”.

5.1.4 Invitations

ORION a invité dans le cadre des séminaires IA, les chercheurs suivants :

- J. Breuker (Univ. Amsterdam) ;
- F. Van Harmelen (Univ. Amsterdam).

5.2 Activités nationales

5.2.1 Coopérations scientifiques

- Coopération avec l'Inra/Geves à Sophia Antipolis (M-H Gandelin) pour l'identification de variétés de roses par systèmes experts et analyse d'images.
- Coopération avec l'Observatoire Nice Côte d'Azur (A. Bijaoui) pour l'analyse et la classification de galaxies.
- Coopération avec l'Observatoire de Paris à Meudon (M. Auvergne) pour le pilotage de codes de calcul scientifique en astrophysique stellaire.

5.2.2 Organisations de séminaires

Cette année nous avons créé des séminaires IA inter-équipes auxquels ont participé ACACIA, ORION et SECOIA.

5.2.3 Invitations

ORION a invité dans le cadre des séminaires IA, plusieurs chercheurs, dont :

- F. Jurie (LASMEA Clermont-Ferrand) ;
- G. Bisson (LIFIA Grenoble) ;
- F. Charpillet (RFIA Nancy).

6 Diffusion des résultats

6.1 Formation

6.1.1 Enseignement universitaire

DEA Informatique et ESSI à l'Université de Nice-Sophia-Antipolis, cours d'Intelligence artificielle (6h, S. Moisan).

DEA. Imagerie et Sciences de l'Univers de l'Université de Nice-Sophia-Antipolis, cours Vision et Intelligence artificielle (10h, M. Thonnat).

DEA Vision et robotique de l'Université de Nice-Sophia-Antipolis, cours d'Analyse de scènes (3h, M. Thonnat).

Mastère Génie Logiciel du Ceram cours Lisp (30h, S. Moisan)

ENSMP Ecole Nationale des Mines de Paris, cours Intelligence artificielle et systèmes experts (30h, S. Moisan).

ENST Ecole Nationale des Telecommunication de Paris, cours Vision et Intelligence artificielle (3h, M. Thonnat).

ISIA-ENSMP Institut Supérieur d'Informatique et d'Automatique, cours Intelligence artificielle et imagerie(3h, M. Thonnat) et cours Intelligence artificielle et Lisp (30h, S. Moisan).

6.1.2 Thèses

M. Thonnat a participé aux jurys de deux thèses en temps que rapporteur et d'une thèse comme examinateur.

6.1.3 Stages

Patrick Laduguie et David Palermo ont fait un stage de 7 mois de fin d'étude dans le cadre du mastère Génie Logiciel du Ceram.

Alain Philippe a effectué un stage de 3 mois de fin d'étude d'élève ingénieur de Supelec.

Maria del Mar Marcos termine son stage préliminaire à la thèse de l'université de Jaume (Castillon, Espagne). Ce stage d'une durée de 7 mois est financé par une bourse COMETT.

6.2 Participation à des manifestations

M. Thonnat a été invitée à la conférence internationale The world of Galaxies II.

7 Publications

Articles et chapitres de livre

- [1] C. SHEKHAR, S. MOISAN, M. THONNAT, «Towards an intelligent problem-solving environment for signal processing», *Mathematics and Computers in Simulation* 36, 1994, p. 347–359.
- [2] S. YU, P. SAINT-MARC, M. THONNAT, M. BERTHOD, «Automatic Identification of Planktonic Foraminifera by Computer Vision», *Journal of Foraminiferal Research*, 1995, en cours de revision.

Communications à des congrès, colloques, etc.

- [3] D. HUTBER, S. MOISAN, C. SHEKHAR, M. THONNAT, «Perception-Interpretation Interfacing for the Prolab2 Road Vehicle», *in: 7th Symposium on Transportation Systems: Theory and Application of Advanced Technology*, Tianjin, China, août 1994.
- [4] S. LIU, M. THONNAT, M. BERTHOD, «Automatic Classification of Planktonic Foraminifera by a Knowledge-based System», *in: The Tenth Conference on Artificial Intelligence for Applications*, IEEE Computer Society Press, p. 358–364, San Antonio, Texas, mars 1994.
- [5] C. SHEKHAR, V. GAUDIN, S. MOISAN, M. THONNAT, «Real-time Supervision of Perception Programs for Prolab2», *in: IMACS International Symposium on Signal Processing, Robotics and Neural Networks*, T. F.

- P. Borne, S. Tzafestas (éd.), p. 11–14, Villeneuve-d'Ascq, France, avril 1994.
- [6] C. SHEKHAR, S. MOISAN, M. THONNAT, «Use of a Real-Time Perception Program Supervisor in a Driving Scenario», *in: Intelligent Vehicle Symposium '94*, Paris, France, octobre 1994.
- [7] L. TRASSOUDAINE, D. HUTBER, P. CHECCHIN, J. ALIZON, J. GALLICE, M. THONNAT, «Building an Environment Map around the Prolab2 Vehicle Using a Controllable Range Sensor», *in: Intelligent Vehicle Symposium '94*, Paris, France, octobre 1994.
- [8] J. VAN DEN ELST, F. VAN HARMELEN, G. SCHREIBER, M. THONNAT, «A functional specification of reusing software components», *in: Sixth International Conference on Software Engineering and Knowledge Engineering*, Knowledge Systems Institute, p. 374–381, juin 1994.
- [9] R. VINCENT, S. MOISAN, M. THONNAT, «Learning as a means to refine a knowledge-based system», *in: Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop*, Hatoyama, Japon, novembre 1994.

8 Abstract

The ORION group is connected to the SHERPA team. The objective of ORION is to facilitate the building of knowledge based problem solving environments for **autonomous systems**. The classes of problems we are interested in are program planning and supervision, and data interpretation. In program planning and supervision we have worked on:

- knowledge modelling for program planning and supervision
- real-time perception program supervision
- knowledge base refinement and validation.

In data interpretation we have started studies on:

- interpretation of sequences of images
- cooperation between knowledge bases
- scenario recognition for video-surveillance applications

Table des matières

1	Composition de l'équipe	1
2	Présentation du projet	2
3	Action de recherche	4
3.1	Pilotage de programmes	4
3.1.1	Modélisation de la connaissance pour le pilotage de programmes	4
3.1.2	Pilotage temps-réel de programmes de perception .	5
3.1.3	Bibliothèque pour la planification	6
3.1.4	Raffinement de bases de connaissances	7
3.1.5	Vérification de bases de connaissances	8
3.2	Interprétation	9
3.2.1	Interprétation de séquences d'images	9
3.2.2	Coopération de systèmes à base de connaissances	10
3.2.3	Passwords : reconnaissances de scénarios dans une application de vidéo surveillance	11
4	Actions industrielles	12
4.1	Esprit PASSWORDS	12
4.2	Eureka PROMETHEUS	13
5	Actions nationales et internationales	13
5.1	Activités internationales	13
5.1.1	Coopérations scientifiques	13
5.1.2	Comité de programme	14
5.1.3	Comités de rédaction	14
5.1.4	Invitations	14
5.2	Activités nationales	14
5.2.1	Coopérations scientifiques	14
5.2.2	Organisations de séminaires	14

5.2.3	Invitations	15
6	Diffusion des résultats	15
6.1	Formation	15
6.1.1	Enseignement universitaire	15
6.1.2	Thèses	15
6.1.3	Stages	16
6.2	Participation à des manifestations	16
7	Publications	16
8	Abstract	17