

---

# Projet ORION

## Environnements de résolution de problèmes pour des systèmes autonomes

---

**Localisation :** *Sophia-Antipolis*

**Mots-clés :** intelligence artificielle, système à base de connaissances, pilotage de programmes, interprétation d'image.

### 1 Composition de l'équipe

#### Responsable scientifique

Monique Thonnat, DR Inria

#### Assistante de projet

Agnès Cortell, SAR Inria [ à temps partiel]

#### Personnel Inria

Sabine Moisan, Chargée de Recherche

Véronique Clément, Chargée de Recherche [ à mi-temps]

Patrick Itey, Ingénieur d'étude [ à 40%]

#### Ingénieurs experts

Nicolas Chleq, ingénieur expert

Jean-Christophe Ossola, ingénieur expert, de juin à décembre

#### Chercheurs doctorants

François Brémond, ENS Lyon, université de Nice-Sophia Antipolis

Monica Crubezy, boursier INRIA, université de Nice-Sophia Antipolis

John van den Elst, boursier HCM, université de Nice-Sophia Antipolis, jusqu'à octobre

Maria del Mar Marcos, université de Castillon Espagne

Jean-Christophe Noel, boursier Cifre CRIL et CEA/CESTA, université de Nice Sophia-Antipolis, à partir d'avril

Jean-Christophe Ossola, boursier MESR, université de Nice-Sophia Antipolis, jusqu'à fin mai

Régis Vincent, boursier INRIA, université de Nice Sophia-Antipolis

### Collaborateurs extérieurs

Frank van Harmelen, Docteur, université d'Amsterdam  
Jean-Paul Rigault, Professeur, université de Nice

### Stagiaires

Elliott Brooks, Erasmus, université d'Edimburgh, du 1er avril au 31 août

## 2 Présentation du projet

Le projet Orion (anciennement action Orion) a été créé en juin 1995.

Notre objectif est de concevoir et de développer des techniques et des logiciels pour, d'une part le **pilotage automatique de programmes** et, d'autre part, l'**interprétation automatique** d'images. Nous focalisons nos travaux sur l'étude des connaissances qui interviennent dans ces deux types de problèmes : connaissances sur les programmes et leur utilisation pour le pilotage automatique de programmes et connaissances sur les objets et les scénarios à reconnaître pour l'interprétation automatique d'images. Thématiquement, Orion est une équipe pluridisciplinaire, à la frontière des domaines de la vision par ordinateur, des systèmes à base de connaissances et du génie logiciel. Les techniques que nous utilisons relèvent donc de ces trois disciplines. Les applications qui nous intéressent relèvent principalement du domaine de la vision par ordinateur. Le domaine de recherche d'Orion concerne les systèmes à base de connaissances autonomes. Plus précisément, notre objectif est 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). Parmi les applications qui nécessitent un fort degré d'autonomie, nous avons identifié d'une part les problèmes d'automatisation de l'utilisation de programmes, et d'autre part les problèmes d'interprétation automatique d'images. L'automatisation de l'utilisation de programmes 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), soit qu'il n'en a pas le temps (saturation de ses capacités), soit enfin 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'images. Par la suite nous utiliserons le terme d'**interprétation automatique d'images**.

## 3 Actions de recherches

Afin de faciliter la construction de systèmes à base de connaissances (SBC), nous développons des moteurs indépendants d'une expertise particulière, mais cependant *dédiés* à une classe de problèmes. Les classes de problèmes envisagées sont le pilotage automatique de programmes et l'interprétation d'images. Le développement d'outils dédiés fournit des systèmes bien adaptés à l'expression des connaissances et largement utilisables dans différents domaines d'applications.

Pour construire de tels moteurs il est nécessaire de se baser sur une modélisation des connaissances et des méthodes de résolution de problèmes qui interviennent dans les classes de problèmes étudiées.

### 3.1 Pilotage de programmes

*Participants* : Monica Crubezy, John van den Elst, Frank van Harmelen, Maria del Mar Marcos, Sabine Moisan, Jean-Paul Rigault, Monique Thonnat, Régis Vincent

*Mots-clés* : pilotage de programmes, planification, réutilisation de programmes.

Le *pilotage de programmes* a pour but d'automatiser l'utilisation de logiciels complexes (bibliothèques de traitement d'images par exemple), grâce à des systèmes capables de planifier des traitements et d'en contrôler l'exécution.

Un système à base de connaissances en pilotage de programmes est généralement constitué :

- d'une bibliothèque de programmes exécutables adaptés à un domaine d'application (par exemple : traitement d'images médicales) ,
- d'une base de connaissances recueillant le savoir-faire d'utilisation des programmes de la bibliothèque (choix, ordonnancement, paramétrage, exécution, évaluation des résultats, ré-exécution des programmes) ,
- d'un moteur de pilotage qui choisit, ordonne et exécute les programmes de la bibliothèque en fonction de l'objectif et des données de l'utilisateur et des connaissances contenues dans la base de connaissances ,
- d'une interface permettant à un utilisateur d'exprimer un objectif à atteindre sur des données particulières, de suivre l'exécution des traitements et de visualiser leurs résultats.

La construction d'outils dédiés à une classe de problèmes particulière (ici le pilotage de programmes) offre deux avantages : tout d'abord focaliser les modèles de connaissances fournis par les outils sur les besoins de la classe de problèmes, et ensuite fournir des formalismes unifiés et communs aux bases de connaissances d'une même classe de problèmes. Nous souhaitons aller plus loin et faciliter aussi l'écriture des moteurs eux-mêmes, afin de pouvoir décrire une stratégie de pilotage à un niveau abstrait et de pouvoir facilement changer de stratégie selon les besoins.

#### 3.1.1 Plate-forme de développement de systèmes de pilotage

*Participants* : Sabine Moisan, Jean-Paul Rigault, Régis Vincent

*Mots-clés* : pilotage de programmes, génie logiciel.

Le but de cette plate-forme est de fournir un environnement commun à différents systèmes de pilotage, comportant par exemple un langage de description et des facilités de vérification de bases de connaissances, ou des interfaces graphiques. Grâce à une telle plate-forme, il est possible de tester plusieurs moteurs sur une même base de connaissances ou plusieurs applications avec le même moteur tout en gardant le même environnement de développement.

Nous avons finalisé la première beta version de la plate-forme LAMA (**L**ibrary **A**nd **p**latfor**M** for **p**l**A**nning).

La plate-forme (voir figure 1) comprend une bibliothèque de composants réutilisables pour construire des moteurs de pilotage de programmes, un langage de description de base de connaissances (YAKL), un serveur de messages (LAD), des interfaces graphiques, des outils de vérification de base de connaissances et des mécanismes de récupérations d'erreurs.

*Bibliothèque de composants.* Le moteur d'un système de pilotage implante les caractéristiques d'une stratégie particulière de pilotage. Si la stratégie est figée dans le moteur, il peut être difficile de la modifier. Toute évolution de spécifications, de besoins ou de contraintes nécessite alors de reconfigurer le moteur, voire de le ré-écrire complètement. Nous proposons de pallier cet inconvénient en permettant la construction des moteurs à partir d'une bibliothèque de composants de base. Nous avons porté un effort particulier sur la phase de planification, car c'est une sous-tâche importante du pilotage de

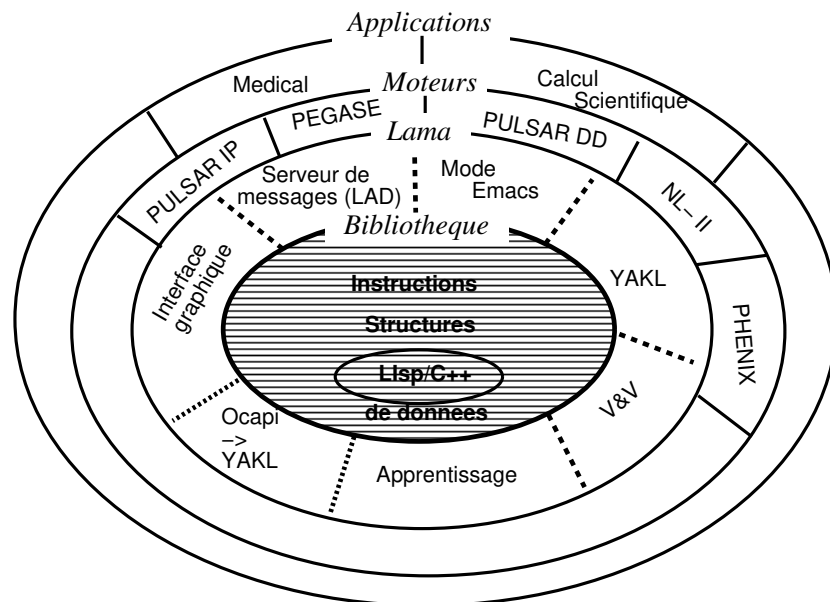


Figure 1: Chaque couche de l'architecture de LAMA correspond à un niveau d'expertise. Le centre (en rayé) est constitué d'une bibliothèque de composants réutilisables pour construire des moteurs. Le niveau LAMA est celui de la plate-forme avec le langage de description de bases de connaissances YAKL, des outils de vérification (V&V), des modes de développement sous Emacs et des outils optionnels comme les interfaces graphiques ou l'apprentissage. Les moteurs de pilotage sont basés sur LAMA et le niveau supérieur est celui des concepteurs de bases de connaissances (Applications).

programmes. L'utilisation d'une telle approche permet de décrire une stratégie de raisonnement à un niveau abstrait, plus simplement et plus rapidement qu'avec un langage de programmation, et de la faire évoluer plus facilement. En effet, le travail de réécriture de moteurs en différents langages ou l'écriture de variantes d'un même moteur, nécessaires pour des tests ou des domaines d'applications différents, sont très fastidieux et coûteux en temps.

La bibliothèque de composants comprend, d'une part, des structures représentant les concepts intervenant en pilotage de programmes et d'autre part des instructions nécessaires à la description des stratégies de raisonnement en pilotage. C'est une bibliothèque de classes représentant les structures de données élémentaires, utilisées par des instructions (instructions de raisonnement), qui peuvent facilement être combinées pour modéliser différents mécanismes de pilotage [143]. Nous avons choisi d'implanter la bibliothèque tout d'abord en Lisp, pour ses facilités de maquettage, puis en C++ pour sa large utilisation et sa portabilité. Cette approche permet une vision unifiée de différents moteurs et fournit une base commode de comparaison entre stratégies de pilotages.

*Langage de description de bases de connaissances*. YAKL (Yet Another Knowledge base Language) permet de décrire le contenu d'une base de connaissances, sans préjuger du langage d'implémentation cible (Lisp ou C++). Ce langage sert à la fois de format de stockage commun à tous les moteurs de la plate-forme et de langage d'écriture ou de consultation de bases de connaissances. YAKL offre une syntaxe proche de la façon de s'exprimer de l'expert pour décrire les opérateurs, les buts, les règles de production qui constituent une base de connaissances. Des outils de vérifications lui sont connectés et le code en langage d'implémentation cible est généré automatiquement.

Un mode Emacs associé a aussi été développé afin de fournir une aide syntaxique à l'écriture de bases de connaissances. Il permet, grâce à des menus, d'écrire plus facilement les descriptions des différentes parties d'une base.

De plus, afin d'assurer un passage plus facile des anciennes bases écrites avec l'outil OCAP vers la nouvelle plate-forme, nous avons écrit un traducteur du format interne d'OCAP vers YAKL. Ce tra-

ducteur, appelable depuis OCAPI, ne fournit pas une traduction complète, en particulier pour les notions inconnues d'OCAPI comme les préconditions ou les effets des opérateurs.

*Serveur de messages LAD.* Un système de pilotage de programmes est un système réparti, constitué de plusieurs processus applicatifs communiquant entre eux, plus ou moins synchronisés les uns avec les autres (interface, moteur, bibliothèque de programmes, etc). Chaque processus doit garder ses caractéristiques propres tout en étant capable de communiquer avec d'autres processus ponctuellement ou de façon prolongée. En particulier les programmes préexistent souvent au système de pilotage, ils fonctionnent sur des machines ou des systèmes particuliers et ils doivent pouvoir continuer à être utilisés de manière autonome dans les mêmes conditions. De plus, un système de pilotage peut être amené à fonctionner comme un sous-système spécialisé d'un système plus général. Par exemple, un système de classification d'objets à partir d'images peut déléguer des tâches de traitement d'images à un système de pilotage.

C'est pourquoi nous avons mis en place un mécanisme de communications assez général (LAD : LAMA communication Daemon) pour assurer le dialogue entre des systèmes aussi différents qu'un moteur de pilotage, une bibliothèque de programmes, une interface utilisateur, un système d'interprétation, un système d'apprentissage, etc. Dans la continuité des travaux de [133], nous avons développé une première version d'un mécanisme de communications sur le principe des bus logiciels, basé sur des communications par sockets Unix TCP.

L'architecture du mécanisme de communication, de type client/serveur, est constituée d'un serveur de messages et d'un module d'émission/réception de messages par processus client. Le *serveur de messages* est chargé d'aiguiller les communications entre les différents *processus clients* qu'il connaît, mais reste indépendant du fonctionnement interne des processus et de la sémantique des messages qu'ils s'échangent. Les processus clients se déclarent au serveur par un nom et les éventuels services qu'ils peuvent rendre. Le serveur communique avec chaque processus client du système au moyen d'un *module générique d'émission/réception de messages* par client.

LAD a été implanté sous forme d'un démon Unix intégrant le serveur de messages et un serveur World Wide Web. L'intérêt est de pouvoir commander le démon à travers un visualisateur WWW. Il est ainsi possible de consulter les processus (services) connectés, de bannir un processus, de stopper les communications, de relancer le démon directement par le visualisateur.

*Interfaces graphiques.*

Nous avons développé une interface graphique optionnelle pour LAMA, permettant de visualiser une base de connaissances sous forme arborescente. Lors de la conception, nous nous sommes particulièrement intéressés aux aspects de communication homme-machine et d'ergonomie.

Il est possible de consulter les opérateurs à différents niveaux de détail, de suivre le flot des données, d'afficher la base de connaissances selon certains critères. Cette interface permet aussi de lancer une résolution et de visualiser le plan exécuté. Il est également possible de consulter a posteriori les données produites aux différentes étapes du plan.

### 3.1.2 Modélisation des connaissances

*Participants :* John van den Elst, Frank van Harmelen, Monique Thonnat

*Mots-clés :* pilotage de programmes, modélisation des connaissances.

Afin de mieux comprendre la problématique du pilotage de programmes de traitement d'images, d'améliorer la qualité de l'utilisation des techniques de traitement d'images existantes et d'élargir leurs domaines d'application, les connaissances impliquées dans le processus de prise de décision de l'expert doivent être décrites de manière structurée et indépendante de l'implantation. Ceci peut être atteint en appliquant des techniques de modélisation des connaissances. Nous avons étudié dans quelle mesure les compétences sur le pilotage de programmes de traitement d'images peuvent être représentées explicitement. Une description des connaissances de pilotage de programmes de traitement d'images permet le transfert des compétences de l'expert qui a développé la bibliothèque vers des utilisateurs novices.

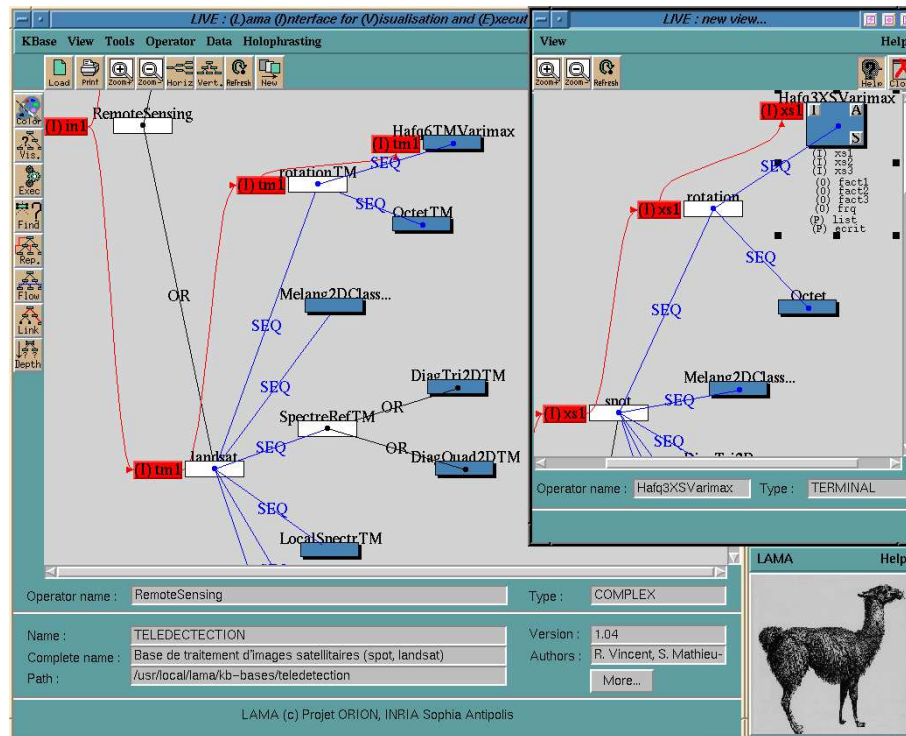


Figure 2: Exemple d'interface de visualisation de base de connaissances.

La modélisation des connaissances est un sujet classique dans le domaine de l'intelligence artificielle, qui implique la description de la nature et de la structure des connaissances spécifiant le comportement de la résolution de problèmes souhaité. Le modèle d'expertise de KADS a montré qu'il est une approche efficace pour ce but. Ce modèle d'expertise fournit une structure des connaissances pré-définie indépendante de l'implantation pour la spécification du comportement de résolution de problèmes, et permet de définir une catégorisation de connaissances nécessaires pour générer ce comportement. En se référant à certaines méthodes de génie logiciel, une telle spécification est appelée une *spécification fonctionnelle*. Nous avons donc utilisé le modèle KADS pour décrire la nature et la structure des connaissances impliquées dans le pilotage de programmes de traitement d'images. Ceci permet une description intuitive et de haut niveau des concepts qui jouent un rôle dans le pilotage de programmes, en utilisant des théories génériques pour différents types de connaissances. De plus, cette modélisation fournit un guide pour la représentation des connaissances de l'expert, propose un cadre pour la vérification et la validation des informations et peut servir de base pour la formalisation et la structure de l'implantation.

Afin de nous permettre d'appliquer et d'évaluer la spécification fonctionnelle du pilotage de programmes de traitement d'images, nous l'avons implantée dans un système de pilotage de programmes. Pour atteindre notre objectif de recherche, le système existant OCAPI a été utilisé comme point de départ. OCAPI (Outil de Contrôle Automatique de Procédures Images) est un noyau de système à base de connaissances dédié au pilotage de programmes, qui a été développé à l'INRIA Sophia Antipolis. Dans un premier temps, nous avons décrit la connaissance contenue dans les concepts et les mécanismes d'OCAPI à un niveau conceptuel en utilisant le *modèle d'expertise de KADS*. En s'appuyant, d'une part sur cette spécification fonctionnelle d'OCAPI et, d'autre part, sur des expériences applicatives, nous avons identifié des informations incomplètes et non pertinentes, telles qu'une représentation des données insuffisamment expressive, un manque de possibilité de réutiliser des programmes, un manque d'information sur la planification dynamique des programmes, des mécanismes d'évaluation et d'ajustement trop localisés, etc.

La spécification fonctionnelle d'OCAPI a été ensuite améliorée et augmentée en se basant sur des techniques de réutilisation de composants logiciels (génie logiciel), des techniques de planification (intelligence artificielle), des systèmes existants de pilotage de programmes, et sur notre expérience pratique sur des applications comme la détection d'obstacles dans des scènes routières, l'imagerie médicale et la classification de galaxies.

Nos travaux ont pour résultat une *spécification fonctionnelle du pilotage de programmes* de traitement d'images.

En appliquant la méthode KADS de *conception par conservation de structure*, ce modèle fonctionnel a été implanté dans un moteur de pilotage de programmes (PULSAR), en utilisant l'architecture de LAMA et le langage d'expression de connaissances YAKL.

Le modèle de connaissances et son implantation ont été illustrés dans le domaine des systèmes de vision autonomes : la détection d'obstacles dans des scènes routières (base PROMETHEE), et le traitement d'images médicales (base AFSIM cf. 3.1.5). Les solutions proposées sont principalement développées pour ces domaines d'application, mais nous avons essayé de les garder les plus génériques possibles.

### 3.1.3 Vérification et Validation

*Participants* : Frank van Harmelen, Maria del Mar Marcos, Sabine Moisan,

*Mots-clés* : pilotage de programmes, validation, vérification, base de connaissances, modélisation des connaissances.

Le problème de la validation et de la vérification de systèmes à base de connaissances (V&V) est crucial pour des systèmes autonomes, afin d'en assurer la robustesse. Nous avons choisi de nous appuyer non seulement sur les propriétés des modes de représentation des connaissances utilisés dans les SBC (schémas et règles), mais aussi sur une métaconnaissance relative aux objets intervenant dans le raisonnement en pilotage de programmes. Nous étudions des systèmes à base de connaissances construits grâce à des environnements de développement dédiés au pilotage et intégrant donc une modélisation fine de cette classe de problèmes. Cette approche nous offre un cadre approprié pour définir des méthodes de validation et de vérification, adaptées à cette classe de problèmes.

Notre but est de fournir aux experts une aide incrémentale pendant la construction d'un système, en intégrant la validation et la vérification dans la méthodologie de développement du système à base de connaissances. Ceci implique des vérifications structurelles, lexicales et syntaxiques aussi bien que sémantiques (sur le rôle des objets ou la validité de leurs relations, par exemple).

Afin de pouvoir identifier les propriétés de la connaissance intéressantes pour la validation et la vérification, nous travaillons aussi sur la modélisation des problèmes de pilotage de programmes. Notre objectif est d'obtenir un modèle suffisamment générique pour pouvoir décrire divers systèmes existants. Ce travail se base d'une part, sur notre propre expérience avec OCAPI et PLANETE, sur les travaux d'autres équipes en pilotage de programmes (VSDE, MVP, ...) ainsi que sur les travaux sur les tâches génériques.

Les facilités de vérification et validation s'appuient sur la plate-forme LAMA. Une question ouverte reste l'adéquation de la granularité des structures et des instructions de LAMA vis-à-vis de nos besoins. Elle doit être appropriée à la vérification des propriétés intéressantes de la connaissance et des stratégies de raisonnement. Dans le but de modéliser le pilotage de programmes, nous avons fait une étude bibliographique de différentes approches pour la modélisation d'autres tâches, comme la planification ou la conception. En effet, la planification est une sous-tâche importante en pilotage de programmes et il existe des rapprochements possibles entre les tâches de pilotage et de conception. Ceci nous a permis d'une part, d'identifier les aspects à considérer au cours de l'analyse d'une tâche, et, d'autre part, d'essayer de tirer parti du travail déjà réalisé pour les tâches de planification et de conception.

Le modèle est actuellement en cours de réalisation. A partir de l'étude bibliographique, nous cherchons à enrichir le modèle de pilotage en ce qui concerne la re-planification et l'exécution. Nous souhaitons obtenir un modèle générique du pilotage de programmes avant d'aller plus loin dans la description et la

formalisation éventuelle des propriétés à vérifier. Cette modélisation pourrait aussi influencer le modèle de LAMA.

### 3.1.4 Moteurs

*Participants* : John van den Elst, Régis Vincent

*Mots-clés* : pilotage de programmes, moteurs, planification.

La bibliothèque de composants de LAMA nous a permis de développer différents types de moteurs de pilotage. Nous avons écrit six moteurs ayant des fonctionnalités spécifiques, dont nous avons développé plusieurs versions. Les tableaux suivants résument les différentes fonctionnalités de chaque moteur.

	NL-II	Pulsar DD	Pulsar IU	Pegase
Type de planification	Non ordonnée	Non ordonnée	Non ordonnée	Ordonnée
Raisonnement dirigé par	les données	les données	les données	les buts
Exécution	Non	Oui	Oui	Oui
Replanification	Non	Non	Oui	Oui
Résolution de contraintes	Non	Non	Non	Non
Réparation	Non	Non	Non	Oui

	Pulsar IP	Phenix
Type de planification	Partiellement ordonnée	Partiellement ordonnée
Raisonnement dirigé par	les buts	les buts
Exécution	Oui	Oui
Replanification	Oui	Oui
Résolution de contraintes	Oui	Oui
Réparation	Non	Non

Les moteurs NL-II, PULSAR DD et PULSAR IU implantent une planification non ordonnée, qui produit des plans où les opérateurs sont exécutés dès que les conditions sont remplies pour qu'ils soient exécutables, sans souci d'ordre. Ces moteurs ont servi de tests pour des problèmes classiques de planification.

PÉGASE est un moteur de pilotage hiérarchique dont la stratégie est proche de celle du moteur d'OCAPI. Nous avons cependant amélioré les mécanismes de réparation. En utilisant PÉGASE, l'expert peut définir dans la base de connaissances les actions de réparation nécessaires pour corriger un plan en fonction des erreurs détectées.

PHÉNIX et PULSAR IP sont des moteurs hybrides. Ils combinent à la fois une planification hiérarchique et une planification dynamique. Leur intérêt est de pouvoir développer des bases de connaissances impliquant les deux types de planification. Aux plus hauts niveaux d'abstraction, l'expert peut généralement fournir des squelettes de plan partiels pour la résolution de différents buts. A des niveaux inférieurs, il peut souhaiter laisser un planificateur dynamique trouver les meilleurs programmes en fonction des données réelles et des conditions d'exécution.

Il est intéressant de noter qu'en tout, nous avons développé plus de 11 moteurs en un an. Afin de comparer les différents moteurs, nous avons utilisé la même base de connaissances comme test (voir [136]). Nous avons choisi comme critère de comparaison la qualité du plan obtenu, sans tenir compte de la complexité de la recherche, ou des performances en temps.

Tous ces moteurs s'appuyant sur des structures communes, la base de connaissances de test n'a pas eu à être modifiée. Cette technique de conception nous a permis de tester facilement différentes variantes du même type de moteur simplement en modifiant la spécification de sa stratégie et le code correspondant.



### 3.1.5 Bases de connaissances

*Participants* : Monica Crubezy, John van den Elst, Sabine Moisan, Monique ThonnatRégis Vincent

*Mots-clés* : pilotage de programmes, base de connaissances, imagerie médicale, planification, astronomie, télédétection.

La construction de bases de connaissances permet de valider un certain nombre de concepts et de développements réalisés dans le cadre de la plate-forme LAMA, ainsi que de mieux spécifier les besoins dans ce domaine. Les techniques de pilotage de programmes ont été appliquées à des bibliothèques existantes de traitements (de séquences) d'images médicales, d'images satellites ou d'images de galaxies.

*Pilotage de traitement (de séquences) d'images médicales.*

En imagerie médicale les traitements permettent d'extraire des informations qualitatives ou quantitatives estimant des paramètres anatomiques ou des fonctions physiologiques, à partir (de séquences) d'images médicales acquises par différentes techniques (médecine nucléaire, imagerie par résonance magnétique, scanner X). Ces informations permettent une meilleure interprétation des examens d'imagerie médicale en termes de diagnostic ou de thérapie. Cependant la complexité du domaine médical (données et paramètres multiples et complexes, objectifs nombreux) conduit à une diffusion restreinte des traitements d'images existants, faute de système aidant à leur utilisation.

Notre but est de proposer une modélisation d'un système de pilotage de ces programmes intégrant les connaissances nécessaires pour la sélection des (chaînes de) traitements et leur exécution. Un tel système doit permettre d'adapter les traitements d'images médicales à différents contextes, de développer de nouvelles applications et de les mettre à la disposition d'utilisateurs non spécialistes comme les radiologues cliniciens, permettant à ceux-ci de se focaliser sur l'interprétation des résultats.

Nous travaillons sur deux applications, basées sur des approches différentes du traitement d'images médicales.

La première étude est menée en collaboration avec l'unité 66 de l'INSERM (La Pitié Salpêtrière, Paris), sur des programmes d'imagerie médicale fonctionnelle. L'objectif est dans un premier temps de faciliter le travail méthodologique de mise au point de chaînes de traitements d'images médicales en vue d'une utilisation clinique routinière. Cette étude concerne essentiellement des programmes d'Analyse Factorielle de Séquences d'Images Médicales (AFSIM), intéressants car leur utilisation judicieuse fait intervenir de nombreuses connaissances tant méthodologiques que médicales.

Nous travaillons d'une part, sur la modélisation des données, qui doit refléter la richesse et la non exhaustivité du domaine médical, d'autre part, sur le mécanisme de planification et de contrôle d'exécution qui doit être perfectionné pour respecter la démarche naturelle des experts. Nous proposons une modélisation des concepts nécessaires à un outil de pilotage de programmes pour ce type d'application : stratégie de planification hybride, mêlant planification à base de squelettes et planification réactive, et description des données structurée en différents points de vue.

Une base de connaissances de pilotage de l'AFSIM a été développée avec YAKL, en intégrant une partie de ces aspects, sur une application de référence : le suivi de chimiothérapie sur ostéosarcome, qui traite des séquences temporelles d'images médicales. Elle a ainsi été testée avec les moteurs de pilotage PÉGASE, puis PULSAR (cf. 3.1.4) qui planifie les opérateurs en raisonnant à partir de leurs préconditions et de leurs effets sur les données, et dont la stratégie se rapproche de la stratégie hybride souhaitée.

Après une phase de simulation, le pilotage des programmes de l'AFSIM en grandeur réelle est en cours. Dans ce cadre, nous réfléchissons aussi à la manière de piloter l'exécution de programmes conçus pour des environnements différents des nôtres. Cet axe fait intervenir des préoccupations rejoignant celles des applications réparties (cf. LAd) : communication, déclenchement de procédures distantes, synchronisation.

Nous venons de commencer une seconde étude : une application de segmentation du cerveau à partir d'images IRM 3D, en collaboration avec Grégoire Malandain du projet Epidaure de l'INRIA. Il s'agit ici de traitements d'images médicales basés sur les principes de la vision par ordinateur (mor-

phologie mathématique essentiellement), visant à isoler le cerveau anatomique dans les images. Les connaissances médicales utilisées par l'expert ne sont donc pas intégrées dans les programmes, mais interviennent à un haut niveau de raisonnement. L'objectif est de définir jusqu'à quel point, sous quelle forme et à quel niveau, des connaissances médicales sont nécessaires pour piloter ces traitements. De plus, la démarche de l'expert est ici principalement sous forme d'essais/erreurs, à cause de la difficulté pour déterminer les valeurs de paramètres appropriées et à la sensibilité de ceux-ci aux conditions d'acquisition. Il faut donc prendre en compte les mécanismes de raisonnement tels que le retour en arrière et la réparation. Une première version d'une base de connaissances pour le pilotage de ces programmes a été développée et testée avec le moteur PÉGASE (cf. 3.1.4).

#### *Pilotage de traitement d'images satellites.*

La base de connaissances SAR a été développée à l'université de Maryland, USA (Center for Automation Research), pour l'analyse d'images satellites SAR. Il s'agit d'identifier dans une image des objets d'intérêt (cibles, bâtiments, routes et arbres) et de segmenter le reste de l'image en diverses catégories (herbe, eau, etc.). Cette base avait été initialement écrite en utilisant OCAPI et elle a été traduite en YAKL et améliorée pour tirer parti des nouvelles possibilités offertes par LAMA et le moteur PÉGASE.

#### *Pilotage de traitement d'images de galaxies.*

La base de connaissances PROGAL écrite pour l'ancien système OCAPI, a été réécrite pour LAMA. Nous avons pour cela utilisé le traducteur automatique pour transcrire les bases d'OCAPI vers YAKL. Cette opération a aussi été l'occasion de reconcevoir en grande partie la base de connaissances. Nous avons notamment ajouté des étapes optionnelles, simplifié certaines étapes et corrigé les flots de données. Grâce aux outils de vérification de LAMA de nombreuses erreurs présentes dans la base PROGAL-OCAPI ont été détectées. Actuellement la base PROGAL comprend 143 opérateurs et 70 règles.

## 3.2 Interprétation d'images

*Participants* : François Brémond, Nicolas Chleq, Jean-Christophe Ossola, Monique Thonnat, Véronique Clément

*Mots-clés* : interprétation d'image, reconnaissance de forme, reconnaissance de scénarios, séquences d'images.

Nous nous intéressons ici à l'interprétation automatique de données images, celle-ci consistant à attribuer une sémantique à ces données en fonction de modèles prédéfinis. Les données à interpréter sur lesquelles nous avons travaillé cette année sont des images 2D et des séquences d'images. Les résultats de l'interprétation sont la reconnaissance d'objets physiques, d'événements, de situations ou de scénarios.

Dans les sections suivantes, nous présentons nos réalisations concernant :

- le développement d'un système automatisé d'assistance aux opérateurs de vidéosurveillance.
- l'élaboration d'un système générique pour la construction de systèmes spécifiques à un type particulier d'applications de vidéosurveillance.
- le développement de systèmes de reconnaissance automatique d'objets naturels complexes, problème pour lequel nous proposons un environnement logiciel.

### 3.2.1 Interprétation de séquences d'images

*Participants* : Monique Thonnat, Nicolas Chleq, François Brémond

*Mots-clés* : interprétation d'image, séquences d'images, objets mobiles, vidéosurveillance.

L'interprétation de séquences d'images a pour objectif, en ce qui nous concerne, de donner un sens à une scène décrivant des activités humaines, à partir d'images fournies par une caméra couleur, monoculaire et fixe. Cette interprétation de scène repose sur la coopération d'un module de traitement

d'images, d'un module de suivi des objets mobiles et d'un module de reconnaissance du comportement des objets mobiles. Il s'agit, pour le module de traitement d'images, de détecter les régions mobiles sur la séquence d'images. Le module de suivi associe les régions détectées afin de former et de suivre les objets mobiles. La tâche du module de reconnaissance des comportements consiste, à partir de techniques d'intelligence artificielle, à identifier les objets suivis et à reconnaître leur comportement comme constitutif d'un ou plusieurs scénarios prédéfinis. Ces recherches sur l'interprétation de scènes se divisent en deux. Une première partie concerne l'élaboration d'un système générique à base de connaissances d'interprétation de scènes en vidéosurveillance. La seconde partie est menée dans le cadre du projet européen PASSWORDS, dont l'objectif est de développer un système automatisé d'assistance aux opérateurs de vidéosurveillance.

### 3.2.2 Système générique d'interprétation pour la vidéosurveillance

*Participants* : François Brémond, Monique Thonnat,

*Mots-clés* : interprétation d'image, séquences d'images, objets mobiles, vidéosurveillance, suivi d'objets.

L'objectif de cette recherche est l'élaboration d'un système générique d'interprétation de scènes à base de connaissances. Ce système générique a pour but la construction de systèmes spécifiques à un type particulier d'applications de vidéosurveillance.

Ces applications se caractérisent par leur environnement dynamique et non structuré, correspondant à des milieux réels divers et servant de cadre aux activités humaines étudiées. Un des problèmes majeurs de ce type d'applications est le suivi des objets mobiles. Si la trace d'un objet mobile est perdue au cours de la séquence d'images, il n'est plus possible d'étudier son comportement. Cette année nous avons porté notre attention sur ce problème. Nous avons développé un module de suivi d'objets mobiles reposant sur l'apparence des objets mobiles sur l'image, plutôt que sur leur réalité dans la scène. Un objet mobile peut ainsi correspondre à un bruit détecté par les traitements d'images (par exemple un reflet, une ombre portée sur le sol), à un individu entier, à une partie d'un individu (par exemple un de ses membres) ou à un groupe d'individus. Un premier objectif du module de suivi est la prise en compte des problèmes classiques de suivi d'objets mobiles (par exemple occultation, pertes et réparation de traces). Un second objectif est de permettre un suivi de tout type d'objets réels. La solution adoptée consiste à apporter une attention particulière au suivi des êtres humains ou plus généralement aux objets mobiles non rigides se déplaçant lentement. Le module de suivi a été développé en C++.

Nous avons également défini un formalisme de description du comportement des objets mobiles. En collaboration avec Elliot Brooks ([142]), nous avons mis au point un prototype qui, à partir de données synthétisées correspondant à des événements mettant en scène des objets mobiles, détermine leur comportement. Ce prototype a été construit à l'aide du système de manipulation de "frames" développé dans l'équipe. La prochaine étape consiste, en s'appuyant sur ce prototype, à développer un module complet de reconnaissance de comportements.

### 3.2.3 Projet Esprit3 PASSWORDS: reconnaissance de scénarios

*Participants* : Nicolas Chleq, Monique Thonnat

*Mots-clés* : interprétation d'image, reconnaissance de scénarios, vidéosurveillance, reconnaissance d'objets, suivi d'objets.

Le projet ESPRIT III PASSWORDS (Parallel and realtime Advanced Surveillance System With Operator assistance for Revealing Dangerous Situations) a pour objectif de développer un système automatisé d'assistance aux opérateurs de vidéosurveillance. Construit sur la coopération d'un module de traitement d'images et d'un module d'interprétation basé sur des techniques d'intelligence artificielle, ce système doit reconnaître automatiquement différents scénarios présentant un danger quelconque, afin de les présenter aux opérateurs. L'objectif du module d'interprétation est donc de détecter puis

d'identifier les différents objets intervenant dans les scènes, de construire et de tenir à jour des informations historiques sur ces objets, puis de synthétiser des événements abstraits constitutifs d'un ou plusieurs scénarios suspects parmi ceux qui sont proposés dans une librairie de modèles prédéfinis

Le travail effectué durant cette année a concerné principalement les applications de type "surveillance de parking" où l'angle de vue assez large permet de voir de nombreux objets de nature distincte (voitures, piétons, chariots, etc.) dans un environnement changeant (éclairage, voitures en stationnement, arbres agités par le vent, etc.). Cette gamme d'application a permis d'expérimenter diverses méthodes de récupération du suivi, d'analyse de trajectoire, de reconnaissance de cas typiques (occultation, formation et séparation de groupes, etc.).

### 3.2.4 Développement de systèmes de reconnaissance automatique d'objets naturels complexes

*Participants* : Jean-Christophe Ossola, Monique Thonnat, Véronique Clément

*Mots-clés* : classement automatique, intelligence artificielle distribuée, système à base de connaissances, reconnaissance de forme, reconnaissance d'objets.

Nous nous intéressons à la résolution automatique de problèmes de reconnaissance d'objets naturels à partir d'images monoculaires. Pour cela deux types de compétences sont indispensables : d'une part, l'expertise du domaine d'application des objets considérés qui permet effectivement de reconnaître l'objet d'autre part, l'expertise en traitement d'images qui permet d'extraire de l'image l'information pertinente pour la reconnaissance.

Nous avons précédemment montré quels étaient les bénéfices que l'on pouvait tirer d'une coopération de systèmes à base de connaissances pour la reconnaissance d'objets naturels tels que les galaxies ou les zooplanctons. Néanmoins, la conception de tels systèmes opérationnels requiert une somme considérable de travail pour le développement des bases de connaissances et des algorithmes de traitement d'images nécessaires.

Or, nous constatons qu'un grand nombre de traitements d'images sont communs à plusieurs de nos applications (galaxies, zooplanctons, foraminifères), que ce soient des algorithmes de bas niveau ou des opérations de niveau d'abstraction élevé comme la détection d'une sous-partie de l'objet. L'interprétation s'appuie également sur des paramètres ou des primitives qui peuvent être communs, comme le contour d'un objet ou sa surface.

Pour aider au développement de systèmes de reconnaissance automatique d'objets naturels complexes, nous proposons donc un environnement constitué non seulement de moteurs de systèmes experts mais également (cf figure 3) :

- d'un ensemble de programmes de reconnaissance de formes ainsi que leur description permettant une utilisation automatique,
- d'un vocabulaire de description des formes à reconnaître, associé à des paramètres mesurables par les programmes.

Cet environnement, nommé **CONORS** (**CO**operative **N**atural **O**bject **R**ecognition **S**ystem), s'appuie sur deux systèmes : l'un opérant des tâches de traitement d'images, l'autre prenant en charge l'interprétation des données issues du traitement de l'objet à reconnaître.

**Le système de traitement d'image dans CONORS** . Le système de traitement d'images actuellement utilisé dans l'architecture de CONORS est basé sur l'utilisation du générateur de systèmes à base de connaissances précédemment développé dans notre équipe : OCAP. Il factorise l'ensemble des algorithmes de traitement d'images que nous avons utilisés dans nos applications, et la connaissance sur leur utilisation. L'ensemble de ces connaissances est organisé sous forme de buts abstraits auxquels sont associés un ou plusieurs opérateurs et un contexte d'utilisation. Ce système dispose également de la bibliothèque de programmes dont la syntaxe d'appel est décrite par les opérateurs élémentaires.

**Le système d'interprétation dans CONORS** : Le système d'interprétation utilisé dans l'architecture de CONORS, quant à lui, est basé sur l'utilisation du générateur de systèmes à base de connaissances

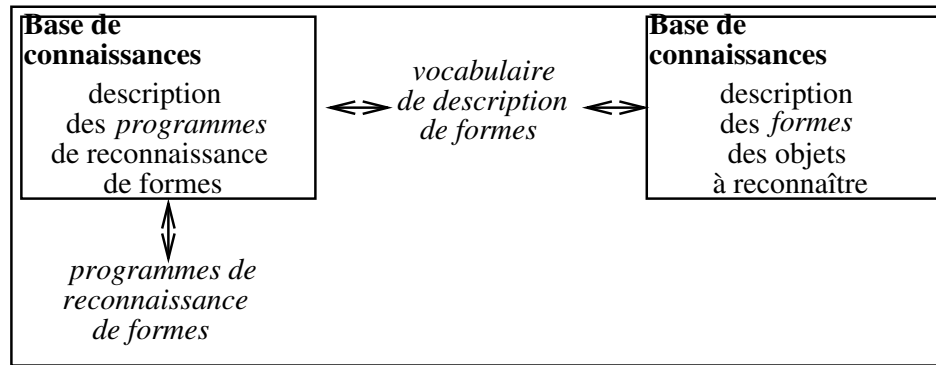


Figure 3: Environnement de développement de systèmes à base de connaissances pour la reconnaissance de formes d'objets naturels complexes

CLASSIC. La connaissance est exprimée à l'aide d'un ensemble de prototypes (8 classes) qui permettent la description d'objets à l'aide de champs (59) auxquels sont associés une ou plusieurs tâches du système de traitement d'images de CONORS. C'est cette connaissance de niveau intermédiaire qui permet l'articulation avec le domaine du traitement d'images et sur laquelle l'expert du domaine d'application s'appuiera pour exprimer ses connaissances de haut niveau.

**L'articulation des composantes de CONORS :** Nous avons défini un lexique de termes qui est utilisé à la fois dans le système d'interprétation pour nommer les champs ci-dessus, et dans le système de traitement d'images pour la description des tâches qu'il est susceptible de résoudre ainsi que pour la description de ses données d'entrée et de sortie. Ce corpus est un *vocabulaire de base* qui permet aux deux systèmes d'échanger des messages dans un langage compréhensible à la fois par un spécialiste en traitement d'images et par un spécialiste d'un domaine d'application quelconque.

La connaissance actuellement contenue dans le système CONORS est basée sur la connaissance commune aux applications précitées (zooplanctons, galaxies). Pour offrir un système minimal de reconnaissance d'objets naturels complexes, se basant sur leur description par des paramètres morphologiques ou radiométriques, nous devons étoffer les bases de connaissances minimales actuelles. La validation de CONORS sera réalisée en développant de nouvelles applications à partir de ces bases minimales (et non pas, comme précédemment, directement à partir des générateurs de systèmes experts), étant bien entendu qu'il n'est pas conçu comme un système figé mais comme un outil extensible voire configurable.

## 4 Actions industrielles

### 4.1 Projet PASSWORDS

*Participants :* Nicolas Chleq, Monique Thonnat

Depuis avril 1994, Orion participe au Projet Esprit3 HPCN R&D Passwords avec des partenaires scientifiques et industriels italiens, belges et français. Nos partenaires industriels sont : Vigitec (Belgique), SRWT (Belgique), Auchan (France et Italie) et SEPA (Italie). Pour plus d'informations voir la rubrique suivante concernant les projets de recherche communautaires.

## 4.2 Michelin

*Participants* : Jean-Christophe Ossola, Monique Thonnat, Véronique Clément

Nous coopérons avec la société Michelin dans le cadre d'un contrat d'une durée de 6 mois (de juin à décembre 1996) pour l'étude de techniques de reconnaissances automatiques de formes complexes appliquées à des pneumatiques.

## 4.3 CEA/Cesta et CRIL

*Participants* : Jean-Christophe Noel

Nous coopérons avec la société Cril de Toulouse et les laboratoire Cesta du CEA à Bordeaux dans le cadre de la thèse Cifre de Jean-Christophe Noel sur le pilotage de codes de simulation.

## 4.4 Visiosport

*Participants* : Nicolas Chleq, Monique Thonnat

Nous coopérons avec la société Visiosport à Sophia Antipolis dans le cadre d'un contrat de conseil pour l'analyse automatique de séquences video de sports collectifs.

# 5 Actions nationales et internationales

## 5.1 Actions nationales

### 5.1.1 Coopérations scientifiques / Groupes de travail

Nous participons au groupe de travail du PRC IA sur la planification (Sabine Moisan). Une réunion de ce PRC sera organisée en décembre à Sophia Antipolis par S. Moisan.

Dans le cadre des PRC, Véronique Clément participe à l'Action Inter-PRC 10.2 "Interaction système / Environnement pour l'Interprétation des signaux et des images" (GDR-PRC ISIS / PRC CHM).

Les membres du projet ont présenté leurs travaux lors des rencontres IA/INRIA qui se sont tenues à l'INRIA Sophia Antipolis les 14 et 15 février 1996.

Nous coopérons avec l'équipe du prof. Di Paola de l'INSERM à Paris pour le pilotage de programmes de traitement de séquences d'images médicales.

## 5.2 Actions internationales

### 5.2.1 Europe de l'ouest

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 et d'une action intégrée PICASSO sur le thème de la validation et de la vérification de systèmes à base de connaissances autonomes.

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.

Le projet participe au projet de recherche communautaire suivant :

- Acronyme: Passwords et Numéro du projet: 8433

- Titre du projet: Parallel and real time Advanced Surveillance System With Operator assistance for Revealing Dangerous Situations
- Programme spécifique du projet : Technologies de l'information
- Modalité du projet : RTD et Domaine : HPCN
- Date de début du projet: 1 avril 1994 - Durée en mois : 35
- Mode de participation du projet Inria: Partenaire
- Nom et adresse du coordonnateur: M. Michel RENARD VIGITEC, rue de la grenouillette 2, 1130 Brussels, Belgique.
- Liste des Partenaires : Vigitec (B), SRWT (B), Auchan (F et I), SEPA (I), CRIF (B), INRIA (F) et l'université de Gênes (I).
- Résumé du projet : le but de PASSWORDS est le développement d'un système générique de vidéosurveillance 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 du projet 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.

## 5.2.2 Etats Unis

Nous coopérons avec C. Shekhar de l'équipe du professeur Chelappa de l'université de Maryland à College Park sur le thème du pilotage de programmes de traitement du signal et d'images.

# 6 Diffusion des résultats

## 6.1 Formation

### 6.1.1 Actions d'enseignement

- DEA Astronomie, imagerie et haute résolution angulaire (Nice) (3h)
- DEA Informatique (Nice) (6h)
- ISIA, École Nationale des Mines de Paris (Sophia Antipolis) (30h)
- École Supérieure des Sciences de l'Ingénieur (Sophia Antipolis) (18h)

### 6.1.2 Thèses soutenues

1. Jean-Christophe Ossola, Coopération de systèmes à base de connaissances pour l'analyse et la reconnaissance d'objets naturels complexes (soutenue le 3 mai 96)
2. John Van den Elst, Modélisation de connaissances pour le pilotage de programmes de traitement d'images (soutenue le 8 octobre 96)

### 6.1.3 Thèses en cours

1. François Brémond, Environnement de résolution de problèmes spécialisé en interprétation
2. Mar Marcos, Vérification et validation de systèmes à base de connaissances en pilotage de programmes
3. Régis Vincent, Étude des mécanismes de gestion des erreurs dans les systèmes à base de connaissances de pilotage de programmes.
4. Monica Crubezy, Pilotage de programmes pour le traitement de séquences d'images médicales
5. Jean-Christophe Noel, Pilotage de programmes pour les codes de simulation en calcul scientifique

### 6.1.4 Stages

Le projet a accueilli Elliot Brooks de l'université d'Edimbourg dans le cadre du programme Erasmus.

## 6.2 Participation à des colloques

Des membres de l'équipe ont participé à des conférences et *workshops* ; on se reportera à la bibliographie pour en avoir la liste.

Dans le cadre de l'Action Inter-PRC 10.2 (GDR-PRC ISIS / PRC CHM), V.Clément a présenté 2 exposés ; M.Thonnat a été invitée au forum, organisé par cette action, sur le thème "Quelle dynamique pour la conception de systèmes en signal et image".

## 6.3 Conférences invitées, tutoriels, cours, etc.

Monique Thonnat a présenté un séminaire en mai à l'université Jaume I (Castillon, Espagne) sur l'interprétation de séquences d'images.

Sabine Moisan et Régis Vincent ont été invités mi-septembre à l'université de Maryland dans le cadre de notre coopération sur le pilotage de programmes avec Chandra Shekhar et l'équipe de R. Chellappa. Sabine Moisan a présenté un séminaire à l'université de Maryland sur l'application de techniques d'intelligence artificielle aux problèmes de traitement d'images et de vision.

Régis Vincent a présenté un séminaire fin septembre à l'université de Massachusetts dans l'équipe de Victor Lesser sur le pilotage de programmes.

## 6.4 Animations scientifiques

V.Clément et J.C. Ossola ont présenté nos travaux au grand public lors des journées *Science en Fête* organisées par le ministère de la Recherche.

## 6.5 Diffusion de logiciels

La plate-forme LAMA (version 1.0 $\beta$ ) a été installée au mois de septembre à l'université de Maryland (USA) dans le département de CFAR (Center for Automation Research). Ce département disposait d'OCAPI et souhaitait disposer des nouvelles fonctionnalités offertes par LAMA. Cette mise à disposition nous a permis de tester LAMA sur des applications extérieures au projet, et notamment sur une base de connaissances SAR qui a été développée au CFAR.

La première version officielle de LAMA devrait être disponible courant janvier 1997.



## 6.6 Diffusion sur Internet

Nous avons continué à développer notre serveur de projet (accessible par <http://www.inria.fr/Equipes/ORION-fra.html> [version française] ou par <http://www.inria.fr/Equipes/ORION-eng.html> [version anglaise]), qui présente en particulier des démonstrations et nos publications.

## 7 Publications

### Thèses

- [133] J. OSSOLA, *Coopération de systèmes à base de connaissances pour l'analyse et la reconnaissance d'objets naturels complexes : application au classement de galaxies ou de zooplanctons*, thèse de doctorat, université de Nice Sophia Antipolis, mai 1996.
- [134] J. VAN DEN ELST, *Modélisation de connaissances pour le pilotage de programmes de traitement d'images*, thèse de doctorat, université de Nice Sophia Antipolis, octobre 1996.

### Articles et chapitres de livre

- [135] N. CHLEQ, «Constrained resolution and abductive temporal reasoning», *Computational Intelligence* 12, 3, May 1996.
- [136] S. MATHIEU-MARNI, S. MOISAN, R. VINCENT, «A knowledge-based system for the computation of land use mixing and the classification of a multi-spectral satellite scene», *International Journal of Remote Sensing* 17, 8, May 1996, p. 1483–1492.
- [137] S. YU, P. SAINT-MARC, M. THONNAT, M. BERTHOD, «Feasibility study of automatic identification of planktonic foraminifera by computer vision», *Journal of Foraminiferal Research* 26, 2, April 1996, p. 113–123.

### Communications à des congrès, colloques, etc.

- [138] F. BRÉMOND, M. THONNAT, «A context representation for surveillance systems», in : *ECCV Workshop on Conceptual Descriptions from Images*, April 1996.
- [139] F. BRÉMOND, M. THONNAT, «Interprétation de séquences d'images et incertitude», in : *Rencontres Francophones sur la Logique Floue et ses Applications*, décembre 1996.
- [140] N. CHLEQ, M. THONNAT, «Realtime image sequence interpretation for video-surveillance applications», in : *International Conference on Image Processing*, September 1996.
- [141] J. OSSOLA, F. BRÉMOND, M. THONNAT, «A communication level in a distributed architecture for object recognition», in : *8th International Conference on Systems Research Informatics and Cybernetics*, August 1996.

### Rapports de recherche et publications internes

- [142] E. BROOKS, *Knowledge Inference Engine*, Mémoire, University of Edinburgh, September 1996, rapport de DEA.
- [143] R. VINCENT, S. MOISAN, M. THONNAT, «Une bibliothèque pour des moteurs de pilotage de programmes», *rapport de Recherche n°3011*, I.N.R.I.A., 1996.

## 8 Abstract

Orion is a multi-disciplinary team at the frontier of computer vision, knowledge-based systems, and software engineering. The aim of Orion is to design and develop technics and softwares aimed at **automatic image interpretation** and at **program supervision and reuse**. This objective is pursued by developing knowledge expression languages, as well as learning mechanisms and reasoning mechanisms for this knowledge, dedicated to specific classes of problems.

Orion focuses on the study of the knowledge involved in both classes of problems addressed: knowledge on objects and scenarii to recognize for automatic image interpretation and knowledge on the programs and on their use for automatic program supervision. We principally study hybrid knowledge representation techniques (frame and rule based), temporal reasoning techniques and planning techniques. Symbolic learning is also a concern for knowledge base refinement.

The main application domain of Orion is computer vision. In a long term we intend to apply our methods and techniques to other domains (signal processing or scientific computation).