



Atelier Logiciel d'acquisition et d'intégration des connaissances en Traitement et Interprétation d'images : Une approche méthodologique

Marinette Revenu, Régis Clouard, Abderrahim Elmoataz, Christine Porquet

► To cite this version:

Marinette Revenu, Régis Clouard, Abderrahim Elmoataz, Christine Porquet. Atelier Logiciel d'acquisition et d'intégration des connaissances en Traitement et Interprétation d'images : Une approche méthodologique. 10e Journées Francophones Acquisition, 1995, Grenoble, France. pp.315-328. hal-00960301

HAL Id: hal-00960301

<https://hal.science/hal-00960301>

Submitted on 17 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Atelier logiciel d'acquisition et d'intégration de connaissances en Traitement et Interprétation d'Images : une approche méthodologique

Marinette REVENU, Régis CLOUARD, Abderrahim ELMOATAZ, Christine PORQUET

Groupe de Recherche en Informatique, Image et Instrumentation de Caen

GREYC -- URA CNRS 1526

ISMRA - 6, Bd Mal Juin 14050 CAEN Cedex

Email: Marinette.Revenu@greyc.ismra.fr

RÉSUMÉ : Le Traitement d'Images est un domaine qui nécessite la conception de systèmes à base de connaissances pouvant servir de plate-forme expérimentale. Nous proposons un Atelier Logiciel permettant d'acquérir et d'intégrer des connaissances procédurales, syntaxiques et sémantiques au fur et à mesure de leur mise en évidence par la réalisation d'applications. Cette intégration est organisée dans une architecture réflexive fondée sur les concepts "Tâche, Méthode, Modèle et Outil". Cette architecture a été testée dans d'autres domaines que l'Image, ce qui atteste de son potentiel dû à la généralité de la méthodologie.

MOTS-CLÉS : Acquisition des connaissances, ingénierie de la connaissance, planification hiérarchique.

1. Introduction

L'objectif de l'Équipe Image du GREYC est de traiter informatiquement des images complexes, principalement d'origine biomédicale, dans le but d'aider à la compréhension des phénomènes physiques visualisés au travers de ces images. Il s'agit, par exemple, de fournir des mesures destinées à évaluer l'évolution d'une tumeur à partir d'une image microscopique de cellules, ou d'établir une carte des sillons corticaux sur une image IRM (Imagerie par Résonance Magnétique) pour guider l'interprétation d'une image TEP (Tomographie par Émission de Positons) reflétant l'activité du cerveau.

La compréhension d'images par ordinateur peut être abordée selon deux approches, suivant les cas à traiter :

- soit en concevant des systèmes entièrement automatiques lorsque le domaine de l'image à analyser est suffisamment connu,
- soit en créant des environnements interactifs de travail favorisant le développement incrémental d'applications. Ces environnements ou Ateliers seront le lieu d'acquisition de nouvelles connaissances et de leur opérationnalisation.

L'explicitation des connaissances pour traiter des images est particulièrement difficile. Néanmoins, les capacités du système visuel humain incitent à tenter de reproduire ou d'approcher ses performances. Les travaux portant sur l'élaboration de modèles psychologiques et computationnels du traitement de l'information des images rétiniennes proposent une démarche dont il est intéressant de s'inspirer. David Marr [Marr-82] estime qu'il faut distinguer trois niveaux d'abstraction pour comprendre, en tant que traitement de l'information, une tâche aussi complexe que la perception visuelle. Pour chaque processus, il faut d'abord spécifier les tâches à effectuer, c'est-à-dire ce qu'il s'agit de calculer et pourquoi. Ensuite, il faut trouver les méthodes ou "algorithmes" qui permettent de réaliser ces calculs et les "représentations" qui constituent les entrées et les sorties de ces algorithmes. Enfin, il est nécessaire de décrire "l'implémentation" des algorithmes. L'élaboration et l'opérationnalisation des différents concepts introduits pour décrire les processus de traitement d'images constituent peu à peu les bribes d'une théorie, dite "computationnelle", expliquant les relations existant entre ces concepts. Cette théorie procure le moyen de préciser quels sont les choix possibles parmi les opérateurs et quelles sont leurs limites. Elle implique la nécessité de faire cohabiter des connaissances déclaratives et procédurales.

Pour créer un Atelier de conception d'applications, en correspondance avec les trois niveaux de compréhension des processus de la perception visuelle, nous proposons d'introduire les concepts de "tâches, méthodes et outils", complétés par des "modèles" représentatifs des données. Dans un premier temps, nous indiquerons en quoi ces concepts répondent à la problématique particulière du traitement d'images et permettent une intégration progressive des connaissances, des plus simples aux plus complexes, au fur et à mesure du développement d'applications. Nous présenterons ensuite l'architecture logicielle correspondante et les modèles d'acquisition. Enfin, nous montrerons comment la réflexivité du langage de représentation des connaissances influe sur l'implémentation, et nous exposerons une expérimentation.

2. La conception d'un Système à Base de Connaissances en Traitement et Interprétation d'Images

La conception de systèmes généraux de Traitement d'Images est au cœur de nombreux travaux de recherche. Diverses approches à base de connaissances ont été envisagées pour effectuer les étapes de traitement précédant la phase d'interprétation : des environnements de programmation d'algorithmes pour aider à la réalisation d'applications à partir de briques représentant les opérateurs et les entités manipulées [Serfaty-91], des systèmes de segmentation d'images dirigés par les buts ou par les données en utilisant des stratégies sous forme de règles [Matsuyama-89], des systèmes de supervision de bibliothèques d'opérateurs fondés sur la génération de plans de traitement [Clément-93]. C'est précisément sur ces étapes dites "Bas niveau", en opposition avec les étapes "Haut niveau" correspondant à l'Interprétation d'Images, que, dans un premier temps, nous voulons valider notre Atelier.

Pour acquérir de réelles connaissances en "Image", il est en effet vital d'analyser des images issues de la réalité; ce n'est qu'à cette condition qu'une réelle participation des experts du domaine de provenance des images pourra être envisagée. Il n'est pas non plus suffisant de développer de nouveaux opérateurs de traitement sans expliciter vraiment les hypothèses simplificatrices, dues notamment à une implémentation dans un espace discret, et les critères d'évaluation. Ces préoccupations se traduisent par deux actions concrètes à mener : "Placer les utilisateurs le plus vite possible en situation de travail pour que les besoins réels soient exprimés" et "Généraliser la connaissance représentée pour pouvoir la réutiliser".

2.1 La problématique du Traitement d'Images

Les problèmes rencontrés en Traitement d'Images se formulent généralement de la manière suivante:

Estimer un objet f , connaissant une observation g , et sachant que:

$$g = A \cdot f + n \quad (1)$$

Dans cette équation, g désigne un ou plusieurs champs d'intensité (dans le cas général, f et g peuvent être des vecteurs ou des fonctions analytiques à plusieurs dimensions), n un terme de bruit d'observation et A un opérateur, linéaire ou non selon la transformation effectuée, que l'on souhaite inverser. Par exemple, si l'image doit être simplement restaurée, c'est-à-dire si l'on souhaite retrouver sa qualité initiale, A représente la fonction de transfert liée au capteur d'acquisition de l'image.

Mathématiquement, tous ces problèmes sont des problèmes inverses mal posés au sens d'Hadamard. Cela signifie que l'équation (1) n'est pas inversible. Les solutions sont multiples et ne dépendent pas continûment des données observées. Un peu de bruit peut entraîner une grande variation des solutions. Le principe général utilisé pour s'affranchir de cette difficulté intrinsèque consiste à introduire des informations connues a priori sur l'espace des solutions, c'est-à-dire sur la scène observée, et à choisir une ou plusieurs stratégies de résolution.

Vis à vis de la résolution de problèmes, le Traitement d'Images est un problème mal structuré, au sens de [Simon-83]. Ce n'est pas un problème combinatoire car il est irréaliste de chercher à produire et à évaluer toutes les images solutions possibles par une procédure algorithmique. Ce n'est pas non plus un vrai problème d'optimisation, puisqu'il n'existe pas de fonctions de coût à optimiser, qui soient génériques et indiscutables. La recherche dans un espace d'états n'est également pas envisageable car elle supposerait disponibles des descriptions de l'état initial et de l'état final, ainsi que des fonctions permettant de mesurer la différence entre deux états intermédiaires. Reste la possibilité de rechercher la solution dans un espace de plans [Currie-91], solution adaptée en résolution de problèmes lorsque la description exhaustive des configurations pouvant être atteintes est infinie.

C'est par ce moyen que nous voulons intégrer l'acquis existant en Traitement d'Images, qu'il s'agisse des bibliothèques d'opérateurs ou des nombreuses techniques et stratégies de traitement. Tout l'enjeu est de savoir quand les utiliser et quels sont leurs effets réels sur les

images. Ces connaissances sont difficiles à expliciter et à formaliser. Le savoir accumulé est non structuré et extrêmement diffus. Il n'existe pas de théorie sur laquelle s'appuyer pour modéliser le domaine. Enfin, la démarche d'un expert est relativement empirique. Celui-ci n'est souvent pas capable d'expliquer toutes les étapes de son raisonnement, et ce, pour deux raisons essentielles. En premier lieu, les problèmes se posent généralement selon des termes du domaine d'application qu'il faut donc d'abord traduire en termes du Traitement d'Images, et ils ne sont pas spécifiés complètement au départ. En second lieu, la connaissance sur les opérateurs est imprécise et peu fiable. Il n'existe pas aussi de fonctions quantitatives d'évaluation des résultats. Tout ceci justifie le recours à beaucoup d'étapes d'essais-erreurs et d'optimisation. Les objectifs initiaux doivent souvent être revus et complétés au cours de la résolution, ce qui nécessite l'intervention de plusieurs types d'experts: l'expert du domaine de provenance de l'image et l'expert en Traitement d'Images, auxquels nous adjoindrons l'expert informaticien ayant des compétences aussi bien en Intelligence Artificielle qu'en Algorithmique.

2.2 Le principe d'acquisition et d'intégration des connaissances

Plutôt que de viser la réalisation d'un système général de résolution, nous proposons un principe d'acquisition des connaissances, intégrant les opérateurs déjà existants et s'appuyant sur une composante coopération des divers acteurs du système. L'objectif est d'élaborer peu à peu une sorte de théorie computationnelle du Traitement d'Images, dans laquelle non seulement les techniques, leurs représentations et leurs effets seront explicités, mais aussi les spécifications des critères permettant de sélectionner ces opérateurs, de contrôler leur application et de les articuler, en laissant un rôle important à l'utilisateur pour pallier les manques, par exemple en critères d'évaluation. Selon [Visetti-91], cette approche conduit ainsi à remplacer l'idée d'un système autonome par un système d'aide à la validation des raisonnements, ouvrant, en outre, sur une vision plus globale du projet.

Notre souci est de traiter des applications réelles, tout en nous préoccupant de la généralité des principes. Nous préconisons notamment:

- *La réalisation d'applications concrètes avec des méthodes déjà identifiées* pour préciser la nature des vrais problèmes et pour établir un dialogue efficace entre les experts du Traitement d'Images et les experts du domaine d'application grâce à l'existence de résultats concrets.

- *La spécification d'une architecture logicielle favorisant la coopération.* L'objectif est d'obtenir un système permettant de gérer des connaissances, c'est-à-dire de les représenter et de les rendre opérationnelles. Ce doit être un cadre structurant pour le développement d'applications et pour l'acquisition de connaissances portant sur le savoir, le savoir-faire et les outils informatiques déjà disponibles.

L'acquisition des connaissances ne peut se faire que par de nombreux allers-retours entre ces deux niveaux. Les connaissances que l'on peut extraire des applications réalisées sont injectées dans le système à base de connaissances où elles peuvent être testées et validées sur de nombreux exemples.

Par ailleurs, l'éventail très large des connaissances à représenter conduit, d'une part, à privilégier la cohabitation des paradigmes de programmation déclaratifs et procéduraux. Et d'autre part, dans un souci de généralité et de pragmatisme, nous souhaitons disposer dans notre atelier de plusieurs modes de contrôle des stratégies:

- (a) enchaînement figé d'un agencement de briques de base dans le cas d'applications bien spécifiées,
- (b) contrôle d'exécution partiellement ou totalement à la charge de l'utilisateur,
- (c) choix, par l'utilisateur ou le système, de la meilleure stratégie parmi plusieurs déjà explicitées, c'est-à-dire choix d'un squelette de plan,
- (d) construction dynamique d'un plan de résolution.

Il est essentiel qu'il n'y ait pas de rupture dans la chaîne des connaissances opératoires et stratégiques. Tout élément formalisé dans le cadre d'un développement de type (a) doit pouvoir être réutilisé dans un développement de type (b), (c) ou (d). Pour instaurer cette sorte de continuum dans le mode d'intégration, nous avons identifié trois niveaux d'abstraction pour l'intégration des connaissances : intégration procédurale, intégration syntaxique et intégration sémantique.

1. *L'intégration procédurale des connaissances.* Il s'agit d'offrir un mode de programmation orienté "tâche", reposant sur la définition d'une plate-forme de conception d'applications suffisamment consistante pour traiter n'importe quelle application où le concept de tâche est valide et significatif. La plate-forme dispose d'opérateurs génériques prédéfinis et de mécanismes facilitant la construction d'enchaînements syntaxiquement corrects. La conception d'applications par un expert du Traitement d'Images revient alors à sélectionner, à paramétrer et à ordonner ces opérateurs. Le contrôle de l'application reste ici totalement figé, défini par le concepteur. Le rôle d'un tel environnement est d'être le support de la communication et de l'échange entre les concepteurs d'applications, c'est-à-dire entre l'expert en Traitement d'Images et l'expert du domaine d'application.

2. *L'intégration syntaxique des connaissances.* La spécification de certains algorithmes par des méthodes procédurales est limitée. Il est utile que le concepteur dispose d'un langage pour exprimer les éléments non décomposables d'un algorithme sous une forme déclarative (sources de connaissances ou règles) à l'instar des systèmes experts. Des connaissances de contrôle, également fournies et structurées hiérarchiquement en tâches et stratégies, prennent en charge la sélection de ces granules de connaissances selon le contexte. Ce niveau d'intégration est qualifié de syntaxique car il n'est pas ici question de raisonner sur les stratégies fournies par le concepteur. Il s'agit d'une transformation de la description des algorithmes, utile par l'innovation et la généralisation des idées qu'elle peut engendrer.

3. *L'intégration sémantique des connaissances.* L'objectif est ici d'identifier et de formaliser l'expertise de Traitement d'Images sous une forme explicite, qui soit à la fois compréhensible pour un expert humain et utilisable par un système à base de connaissances en vue de faire de la résolution de problèmes. Il faut pour cela proposer un modèle de la connaissance et un

vocabulaire d'expression qui aient une réelle pertinence cognitive. Pour que la connaissance codée dans le système puisse être comprise et remise en cause, il faut en effet qu'il y ait une correspondance directe avec celle qui est exprimable par l'expert [Aussenac-92]. Concrètement, la validation de cette expertise est effectuée via un générateur de plans qui doit fournir une solution opérationnelle en terme de décomposition de tâches dans un formalisme identique à celui utilisé par le concepteur au niveau d'intégration procédurale. Le plan sert de support pour la communication entre les experts, pour échanger, critiquer ou réviser des savoir-faire.

La possibilité de faire cohabiter plusieurs paradigmes de programmation, compatibles et permettant d'aller jusqu'à la phase d'expérimentation, nous paraît essentielle pour acquérir tout type de connaissances. En particulier, l'aide au raisonnement nécessite qu'il y ait, au sein de notre Atelier, un module résolvant des problèmes en générant automatiquement des plans, et un cadre permettant d'instaurer une coopération entre experts et concepteurs.

2.3 La coopération homme/machine

Les systèmes coopératifs visent à rendre explicite le rôle de l'utilisateur dans le processus de résolution d'un problème. Ils ne cherchent pas à automatiser entièrement les tâches d'un expert, mais à faire coopérer utilisateur et système en exploitant la complémentarité de leurs compétences. Cette vision de la coopération peut s'étendre au "système global experts-machine". Les moyens de la coopération ne sont alors pas nécessairement représentés totalement dans le système. Cette approche "coopérative" pour concevoir le système ne conduit pas à répartir de façon optimale des tâches entre des agents pour résoudre un problème. Elle fournit un cadre et des mécanismes pour gérer efficacement une représentation de connaissances opérationnelle en faisant coopérer des utilisateurs, des experts, des informaticiens et une architecture de contrôle. L'intérêt du système réside plus dans sa capacité d'accompagner l'utilisateur dans l'exploration méthodique de son espace problème, que dans celle de trouver la meilleure solution [Zacklad-93].

Trois types d'utilisateur sont reconnus actuellement dans nos activités; ils correspondent aux trois types d'experts nécessaires à l'analyse d'une image. Ils se distinguent par leurs compétences en matière de Traitement d'Images et par leurs connaissances sur l'Atelier lui-même.

1- *L'utilisateur expert du domaine de provenance de l'image.* Il utilise le système en mode automatique. Il présente son problème via les interfaces d'entrée et attend ses résultats via les interfaces de sortie. Néanmoins, il peut être interrogé interactivement par le système pour compléter les termes de sa requête.

2- *L'utilisateur expert en Traitement d'Images.* Il dispose du système comme d'un environnement lui permettant de construire lui-même ses propres plans de traitement, d'enchaîner des codes de la bibliothèque ou de demander au système de résoudre sélectivement des parties de plan.

3- *L'utilisateur ayant la double compétence d'expert en Traitement d'Images et d'expert en conception de SBC.* Il a la possibilité d'incrémenter la base de connaissances, ou de la

modifier. Il peut aussi intervenir spontanément sur le déroulement d'une session de résolution en modifiant les choix faits par le module de raisonnement.

A ces trois catégories d'utilisateurs, peuvent être associés deux modes principaux de fonctionnement. Dans le premier mode, c'est l'utilisateur qui a l'initiative de la résolution. Il peut disposer de tout l'environnement du système pour résoudre son problème. A tout moment, il peut solliciter le système pour automatiser la résolution de tâches particulières. Dans le second mode, l'utilisateur demande au système de prendre en charge la résolution d'un problème. Le système fonctionne alors en mode automatique, mais si l'utilisateur le souhaite et le peut, il doit pouvoir intervenir lors de la résolution pour modifier dynamiquement les choix faits par le système, changer les stratégies et leurs paramètres.

3. L'Atelier d'Acquisition et d'Intégration de Connaissances

L'architecture de l'Atelier d'acquisition et d'intégration des connaissances s'appuie sur la définition de briques de base avec lesquelles seront également conçues des applications. Elles permettent donc de décrire aussi bien les mécanismes d'exécution des tâches du domaine des images que ceux d'exécution des tâches de contrôle de l'Atelier lui-même.

3.1 L'architecture "Tâche, Méthode, Outil et Modèle"

Le modèle conceptuel de notre Atelier est fondé sur les concepts "Tâche, Méthode, Outil et Modèle" (cf. Figure 1). Par l'analyse des tâches cognitives mises en jeu, par l'explicitation des modèles de données, et par l'identification des méthodes de résolution de problèmes [Revenu-94], nous proposons de spécifier ce que doit faire le système et comment. Le langage d'expression des connaissances associé à ces concepts doit être relativement "naturel" aux yeux de l'expert et "performant" aux yeux du cognicien [Krivine-91].

A la notion de tâche, qui émerge des divers domaines concernés: théorie de la vision et ergonomie, nous avons ajouté les concepts de méthode, outil et modèle. L'ensemble de ces concepts ont été repris de divers travaux portant sur l'acquisition des connaissances. On peut citer la méthodologie KADS [Breuker-87] qui fournit un modèle statique de la connaissance, et les environnements de programmation COMMET [Steels-92], LISA [Delouis-94], GOLEM [Deslandes-94] qui produisent des modèles opérationnels que l'on peut valider au fur et à mesure des développements. COMMET introduit les concepts de tâche, méthode et modèle, LISA spécifie les méthodes alternatives ainsi que les aspects réflexifs d'une architecture, et GOLEM complète ces aspects par l'introduction d'outils.

Les *tâches* décrivent ce que doit effectuer le système dans l'objectif de résoudre un problème. Ce sont des entités structurées qui rassemblent toutes les informations impliquées dans le contexte d'exécution et dans le traitement opéré sur les données pour obtenir les résultats. Les données et les résultats sont décrits à l'aide de modèles.

Les *méthodes* spécifient comment une tâche peut être accomplie. Plusieurs types de méthodes peuvent être décrites :

- les **méthodes solutions** quand le problème est primitif,
- les **méthodes de décomposition de tâches** quand la résolution est déterministe,
- les méthodes d'inférence, c'est-à-dire les **méthodes de résolution de problèmes**, dans le cas général.

Les méthodes solutions font référence aux *outils*. Ceux-ci sont la représentation des procédures de calcul couramment utilisées par l'expert, et sur lesquelles il existe un certain consensus. Ce sont des connaissances opératoires décrivant la syntaxe d'appel, les domaines de valeurs des paramètres et la nature des entrées et des sorties.

Les *modèles* décrivent les entités sur lesquelles sont réalisés les traitements. Les primitives utilisées sont les primitives épistémologiques suivantes : concepts, propriétés, relations et structures.

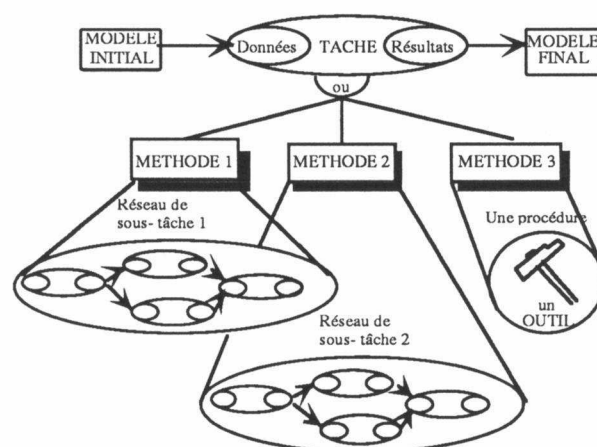


Figure 1 : Les primitives tâche, méthode, outil et modèle

Ce modèle permet la mise en œuvre et la fédération des trois niveaux d'intégration: procédural, syntaxique et sémantique.

3.2 L'intégration procédurale des connaissances

L'intégration procédurale des connaissances fait référence à la conception de graphes de décomposition de tâches, figés pour un type d'utilisateur donné. Outre la représentation des tâches et des méthodes en accord avec la stratégie du concepteur, elle nécessite la mise à disposition d'une bibliothèque d'opérateurs de traitement d'images représentés dans l'architecture par les outils.

Notre mode de conception des opérateurs reflète la nécessité de disposer d'un modèle abstrait des opérateurs pour acquérir une vision plus globale des mécanismes sous-jacents aux traitements (cf. Figure 2).

Un opérateur de Traitement d'Images [Dalle-93] correspond à une action appliquée sur les données d'entrée pour construire les résultats. Cette action peut se concevoir comme l'application d'une opération élémentaire sur des sous-structures qui balayent les entrées. Nous avons dégagé trois types de structures de consultation des données, ou rôles:

- celles qui prennent en compte des critères géométriques, les *loci* comme les fenêtres, les voisinages, les points, etc.,
- celles qui agissent comme éléments structurants pour des opérations spécifiques, comme les masques de la morphologie mathématique,
- celles qui permettent une représentation intermédiaire des données d'entrée et de sortie, comme les histogrammes, ce qui donne la meilleure perception de l'image pour un traitement donné.

La conception d'un opérateur se décompose alors en trois étapes:

- 1- Sélection des modes de consultation des données d'entrée et de sortie
- 2- Balayage en parallèle des données d'entrée et de sortie selon le mode de consultation choisi. Ce balayage peut être systématique ou conditionné.
- 3- A chaque étape, application d'une opération ponctuelle sur l'élément d'entrée consulté.

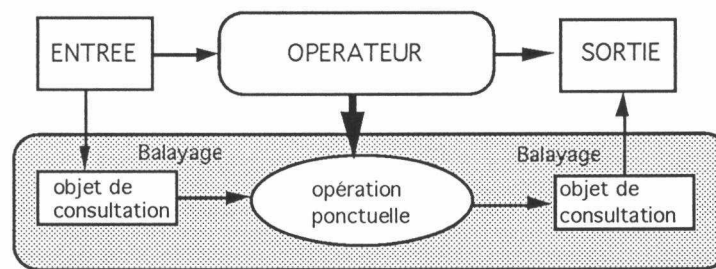


Figure 2: Schéma d'un opérateur abstrait

Ce mode de conception général permet de mettre en évidence des analogies entre des traitements portant sur des représentations différentes d'une image et utilisant les mêmes structures de consultation des données. Les propriétés des opérateurs peuvent alors être généralisées d'une représentation à l'autre. Citons par exemple l'opérateur "érosion" qui peut être appliqué sur une image sur sa forme tableau de pixels ou sur sa représentation en graphe de voisinage de régions.

3.3 L'intégration syntaxique des connaissances

Pour pallier la limitation d'expression des algorithmes par des méthodes procédurales classiques, il est intéressant d'étudier comment les procédures de calcul travaillent et de faire une description explicite des mécanismes internes des algorithmes. C'est ce que nous appelons "intégration syntaxique des connaissances" puisque ce sont les mêmes connaissances que celles enfouies dans du code informatique, mais exprimées différemment. Leur formulation déclarative en permet une utilisation plus variée et plus souple. Tous les enchaînements de ces granules de connaissances n'ont pas à être prévus initialement. Les méthodes spécifiant comment peut être accomplie une tâche sont alors constituées d'un ensemble de sources de

connaissance décrites par des expressions au format "Conditions - Actions", et un mécanisme de contrôle hiérarchique applique ces sources jusqu'à ce qu'une condition générale soit vérifiée.

Cette approche est particulièrement intéressante pour tenter de concevoir des algorithmes de segmentation en décrivant les lois de la perception visuelle sous forme de règles de production. Pour ce faire, un langage d'expression des sources de connaissances a été spécifié. Il permet de faire référence aux objets de l'image (contours et régions). Il comporte la définition de valuations et de comparateurs flous, ce qui rend l'expertise de segmentation potentiellement indépendante de la nature de l'image traitée.

Voici un exemple de source de connaissance: "Si deux régions ont leurs niveaux de gris proches, si l'une d'elles est de très petite surface et si l'autre est d'une assez grande surface, Alors fusionner ces deux régions". Avec ce mode d'intégration de connaissances, il est commode d'énoncer des conditions d'application d'une action (ici, la fusion de deux régions d'une image) sans se préoccuper simultanément de l'ordre d'application des heuristiques.

3.4 Intégration sémantique des connaissances

Pour aller au-delà dans l'acquisition et l'intégration des connaissances en Traitement d'Images, il est nécessaire d'introduire les connaissances permettant de générer automatiquement des applications par planification, ce qui oblige à expliciter un large spectre de connaissances et fournit un moyen de validation. Nous attribuons à ce stade d'intégration le qualificatif de "sémantique" car à toutes les connaissances sont associées des informations sur le "pourquoi" et le "comment" de leur présence et de leur utilisation.

Trois catégories de connaissances sont nécessaires pour construire les plans:

1- Les connaissances sur le domaine d'application permettent de donner un sens et un sujet à l'image à traiter, et de comprendre la requête.

2- Les connaissances "Image" ont en charge la détermination des techniques pour accomplir les tâches de Traitement d'Images, ainsi que l'évaluation de leurs résultats. Doivent être distinguées les connaissances sur les techniques et celles définissant la façon de les utiliser.

3- Les connaissances sur les opérateurs guident la sélection, la paramétrisation et l'enchaînement des opérateurs pour réaliser les techniques de Traitement d'Images. Se trouvent dans cette catégorie les connaissances syntaxiques, destinées à construire des enchaînements syntaxiques corrects, et les connaissances sémantiques qui permettent la sélection et le contrôle de ces opérateurs.

Le plan est obtenu par la décomposition hiérarchique du problème posé au niveau de la *requête* en problèmes plus simples, exprimant d'abord les *tâches* à effectuer, les *fonctionnalités* ou actions logiques, les *procédures* ou actions physiques abstraites, et enfin les *opérateurs* ou codes exécutables. Le choix d'une hiérarchisation fixe des phases de raisonnement permet de structurer les connaissances mises en œuvre et de rendre possible la construction de plans par assemblage de parties indépendantes. Elle a également un rôle important à jouer pour faciliter la compréhension du plan obtenu.

Une application est représentée par un plan de traitements sous la forme d'un arbre (ET, PUIS, OU) de buts, à cinq niveaux d'abstraction (cf. Figure 3). Certaines alternatives sont créées lors de la génération du plan et peuvent être conservées. Cette représentation permet de lancer l'application en routine en ne conservant que l'enchaînement des opérateurs, ou de comprendre et modifier la stratégie de l'analyse de l'image.

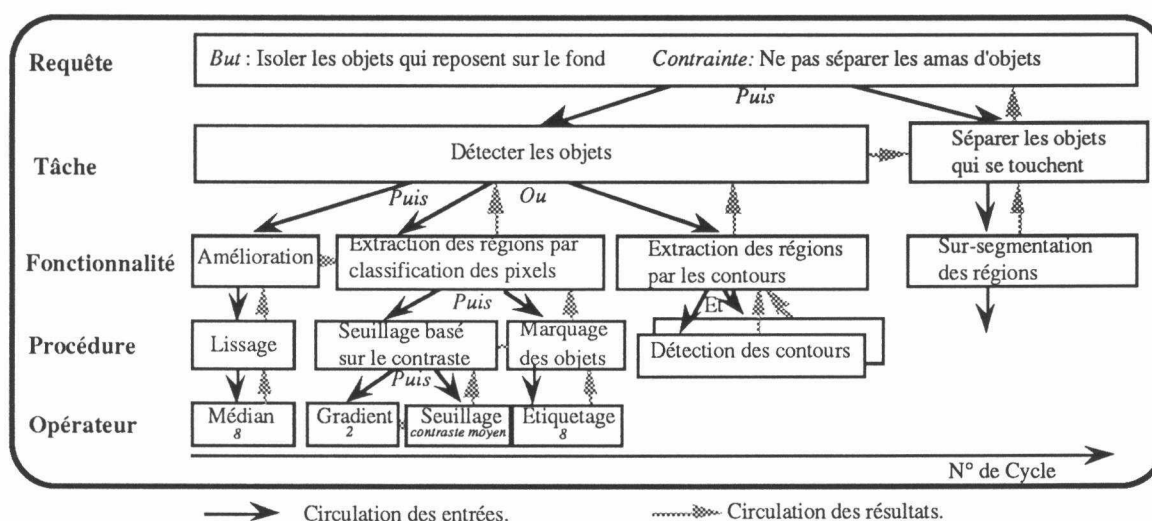


Figure 3: La représentation d'un plan

L'existence d'un module de résolution de problèmes est intéressante dans notre démarche car il permet d'obtenir une première solution lorsque l'on ne dispose pas de connaissances particulières sur l'image à analyser. Il sert alors d'outil d'expérimentation. Enfin, la possibilité de manipuler le plan favorise sa compréhension, son amélioration et sa modification.

3.5 L'articulation entre les niveaux d'intégration de connaissances

La gestion du contrôle des stratégies de l'Atelier doit également être assurée au moyen de tâches, méthodes et outils, pour rendre possible la cohabitation de plusieurs modes de contrôle, et en particulier d'un métacontrôle. A cet effet, un certain nombre de tâches de contrôle ont été représentées pour constituer l'amorçage du système: tâches de gestion des tâches, des méthodes, des outils et des utilisateurs. Enfin, pour que le système soit opérationnel, une boucle minimale de contrôle a été programmée. Elle enchaîne le choix d'une tâche et l'exécution de cette tâche, ce qui implique le choix d'une méthode et sa mise en oeuvre.

Le modèle conceptuel de l'Atelier permet de prendre en compte les deux composantes principales des interactions avec les utilisateurs selon des principes ergonomiques [Pinsky-89]: la description de l'action (la réalisation d'une intervention dans l'environnement physique) et la description de l'activité (le processus de pensée qui aboutit aux actions). L'action apparaît comme une tâche sur le domaine et l'activité est une tâche de réflexion sur les actions, sur les tâches du domaine, et relève du contrôle. L'une et l'autre ont recours à des enchaînements, que ce soit dans le cours d'action ou dans les modèles fonctionnels d'activité, qui sont autant de graphes de tâches, ce qui est en adéquation avec notre méthode de travail.

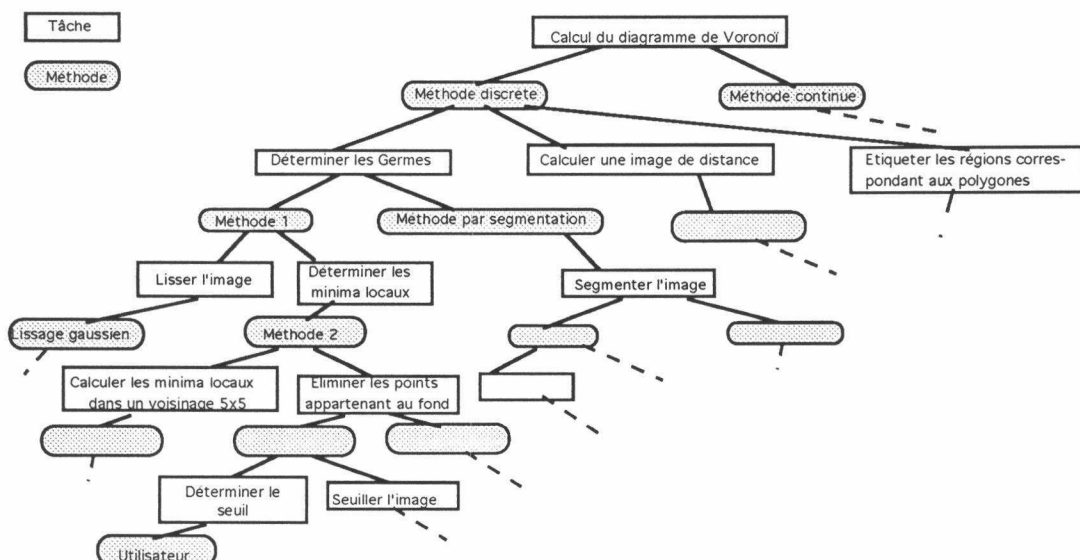
4. Implémentation et résultats

4.1 La représentation de l'architecture par un langage réflexif

Les méthodes utilisées pour la réalisation de l'architecture "Tâche, Méthode, Outil, Modèle", sont fondées sur la représentation des connaissances par un langage à objets réflexif. Il s'agit du langage Airelle développé au GREYC en LeLisp. Les traitements de plus bas niveau sont développés en C et C++, pour des raisons d'efficacité et de portabilité des applications cibles. La réflexivité est aussi à la base du modèle de contrôle choisi pour réaliser le générateur de plans BORG [Clouard-93]. Celui-ci utilise un modèle de Tableau Noir à contrôle par Tableau Noir, de type BB1. Le choix d'un langage de représentation des connaissances réflexif permet de décrire le contrôle et le métacontrôle sous un même formalisme, et de faire évoluer le comportement du système sans remise en cause de l'ensemble.

4.2 Une expérimentation image

Pour illustrer notre propos, nous donnons ici un exemple d'application. Il s'agit d'analyser la structure du tissu de la paroi de l'œsophage, et de détecter des anomalies de répartition spatiale. Le problème se pose d'abord en termes de Traitement d'Images: "Comment mettre en évidence les relations de voisinage entre noyaux de cellules", et ainsi changer de représentation pour raisonner ensuite sur un graphe. Nous avons choisi de partitionner l'espace à l'aide d'un diagramme de Voronoï. La figure 4 montre la décomposition de la tâche "Calcul du diagramme de Voronoï", dont le résultat peut être vu sur l'image associée.



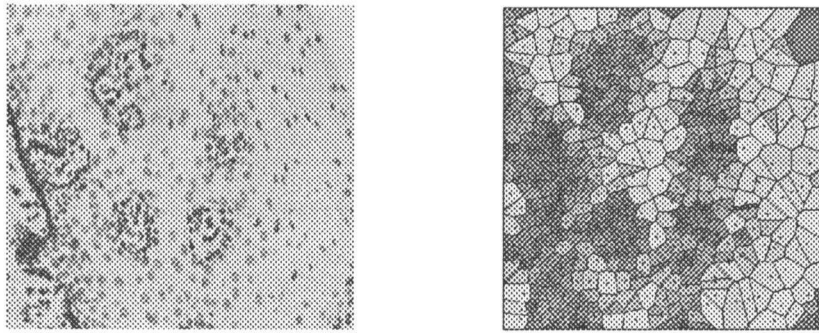


Figure 4: Partie du graphe de tâches et de méthodes pour partitionner une image d'histologie en polygones de Voronoï

5. Conclusion

L'architecture "Tâche, Méthode, Outil, Modèle a été validée par deux expérimentations effectuées dans le cadre de thèses. Le premier travail portait sur la conception d'un SBC pour la sélection de blés hybrides [Deslandes-94], et le second sur le développement d'un système d'aide à la conduite de projets [Moire-94].

Actuellement, l'Atelier n'est pas encore complètement opérationnel. Le générateur de plan fonctionne indépendamment de l'Atelier. Notre objectif à court terme consiste à concrétiser leur articulation et à mener une évaluation de l'ensemble.

Remerciements

Ce travail est réalisé dans le cadre du Pôle "Traitement et Analyse d'Images" de Caen. Nous remercions les médecins et les biologistes du Centre F. Baclesse pour les images et problèmes qu'ils nous donnent à traiter et pour leur participation dans l'acquisition d'une expertise Image.

Bibliographie

- [Aussenac-92] - Nathalie Aussenac, Jean-Paul Krivine, Jean Sallantin, "L'acquisition des connaissances pour les systèmes à base de connaissances", *Revue Française d'Intelligence Artificielle*, Vol. 6 (1-2), pp 7-18, 1992.
- [Breuker-87] - Joost Breuker, "Model-driven knowledge acquisition: interpretation models", *Esprit Project 1098, Deliverable D1*, 1987.
- [Clément-93] - Véronique Clément, Monique Thonnat, "A Knowledge-Based Approach to Integration of Image Processing Procedures", *CVGIP : Image Understanding*, Academic Press, Vol. 57-2, pp 164-184, 1993.

- [Clouard-93] - R. Clouard, C. Porquet, A. Elmoataz, M. Revenu, "Resolution of image processing problems by dynamic planning within the framework of the Blackboard model", *SPIE : Intelligent Robot and Computer Vision XII*, Boston, Vol. 2056, pp 419-429, 1993.
- [Currie-91] - Ken Currie, Austin Tate, "O-Plan: the open planning architecture", *Artificial Intelligence*, N° 52, pp 49-86, 1991.
- [Dalle-93] - Patrice Dalle, Philippe Dejean, Jean-Michel Inglebert, "Méthodologie de conception d'applications de traitement d'images", *4ème Journées Orasis du GDR-PRC Communication Homme/machine*, pp 92-95, 1993.
- [Delouis-94] - Isabelle Delouis, "Lisa, un langage réflexif pour la modélisation du contrôle dans les systèmes à base de connaissances. Application à la planification de réseaux électriques", *Thèse de l'Université de Paris-Sud*, 1994.
- [Deslandes-94] - Joël Deslandes, Marinette Revenu, "Système d'acquisition de connaissances fondé sur le Principe de Coopération: Application à l'aide à la création de variétés de blés hybrides", *9ème Congrès RFIA*, Paris, pp 709-714, 1994.
- [Krivine-91] - Jean-Paul Krivine, Jean-Marc David, "L'acquisition des connaissances vue comme un processus de modélisation : méthodes et outils", *Intellectica*, 12, pp 101-137, 1991.
- [Marr-82] - David Marr, "Vision: A computational investigation into the human representation and processing of visual information", *San Francisco: W.H. Freeman and Co*, 1992.
- [Matsuyama-89] - Takashi Matsuyama, "Expert for image processing : Knowledge-based composition of image analysis processes", *CVGIP*, Vol. 48, pp 22-49, 1989.
- [Moire-94] - Thierry Moire, Marinette Revenu, "A software architecture based on a computational model of knowledge: Application of the integration of the method for project design and managing methods", *1st European Conference on Cognitive Science in Industry*, Luxembourg, pp 37-52, 1994.
- [Pinsky-89] - L. Pinsky, J. Theureau, "Concevoir pour l'action de l'opérateur", *Ergonomie de conception, Edition Universitaire*, 1989.
- [Revenu-94] - Marinette Revenu, "Conception d'un Atelier Logiciel d'Intégration de Connaissances en Traitement et Interprétation d'Images: un guide méthodologique", *Mémoire d'habilitation à diriger des recherches*, Caen, 1994.
- [Serfaty-91] - Véronique Serfaty-Dutronc, "La programmation par "mécanismes": de la compréhension d'algorithmes en Traitement d'Images à leur conception", *Thèse de l'Université de Paris-Sud*, France, 1991.
- [Simon-83] - Herbert A. Simon, "Search and Reasoning in Problem Solving", *Artificial Intelligence*, N° 21, pp 7-29, 1983.
- [Steels-90] - Luc Steels, "Components of Expertise", *AI Magazine*, pp 28-49, 1990.
- [Visetti-91] - Yves-Marie Visetti, "Des systèmes experts aux systèmes à base de connaissances : à la recherche d'un nouveau schéma régulateur", *Intellectica*, N° 12, pp 221-279, 1991.
- [Zacklad-93] - Manuel Zacklad, "Les SBC vus comme des Systèmes Interactifs d'Aide à la Résolution de Problèmes. Apports méthodologiques de l'Aide à la Décision à l'Acquisition des Connaissances", *JAC*, 1993.