



HAL
open science

A Software Workbench for Knowledge Acquisition and Integration in Image processing

Régis Clouard, Marinette Revenu, Abderrahim Elmoataz, Christine Porquet

► **To cite this version:**

Régis Clouard, Marinette Revenu, Abderrahim Elmoataz, Christine Porquet. A Software Workbench for Knowledge Acquisition and Integration in Image processing. Int. Workshop on the Design of Cooperative Systems (COOP), 1995, Juan-Les-Pins, France. pp.298-313. hal-00963717

HAL Id: hal-00963717

<https://hal.science/hal-00963717v1>

Submitted on 15 Apr 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.

A SOFTWARE WORKBENCH FOR KNOWLEDGE ACQUISITION AND INTEGRATION IN IMAGE PROCESSING

Régis Clouard[◇], Marinette Revenu[◇], Abderrahim Elmoataz^{◇◇} & Christine Porquet[◇]

[◇]Laboratoire d'Algorithmique et d'Intelligence Artificielle de Caen
ISMRA, 6, bd du maréchal Juin
14050 Caen Cedex, France

^{◇◇}Laboratoire Universitaire des Sciences Appliquées de Cherbourg
Rue Max Pol Fouchet
50130 Octeville, France

Phone : 31.45.27.21

Fax: 31.45.27.60

E-mail: Régis.Clouard@l2i.ismra.fr

Abstract:

Abstract: Image Processing is a typical domain where one has to implement knowledge-based systems integrating cooperative activities, on the one hand between human experts, when knowledge acquisition is concerned, and on the other hand between human experts and system, when problem-solving is concerned. Designing Image Processing applications involves three types of expertise: expertise about the domain of application, where the application is specified, Image Processing expertise, where a solution in terms of Image Processing operations is proposed, and expertise about computer design concerned with finding a satisfactory implementation of the system. In this article, cooperative processes are presented along two aspects: knowledge acquisition and problem-solving. In the first case, the user directs the resolution stage, whereas the system provides him with its abilities in ergonomics, computation or storage. In the second case, it is the system that solves problems, and the experts can intervene either when they are questioned by the system to get additional information about the problem, or spontaneously to modify choices made by the system or correct some of its deficiencies.

Keywords:

Model for man-machine cooperation, procedural and semantic knowledge integration, hierarchical planning of graphs of operators, image processing applications.

Résumé :

Le Traitement d'Images est un exemple typique de domaine qui nécessite la réalisation de systèmes à base de connaissances intégrant des activités de coopération, d'une part entre experts humains, lorsqu'il s'agit de l'acquisition des connaissances, et d'autre part, entre experts humains et système de résolution lorsqu'il s'agit de résolution de problèmes. La conception d'applications de Traitement d'Images fait intervenir trois types d'expertise : celle du domaine d'application détient la définition de l'application, celle du Traitement d'Images propose une solution en terme de traitements, et celle de la conception informatique autorise une implémentation informatique satisfaisante. Ce papier présente les processus de coopération selon deux niveaux : acquisition des connaissances et résolution de problèmes. Au premier, c'est l'utilisateur qui mène le processus de résolution, le système met à la disposition de l'utilisateur ses capacités d'ergonomie, de calcul et de stockage. Au second, c'est le système qui résout les problèmes, les experts interviennent soit lorsque le système les interroge pour des compléments d'informations sur le problème, soit spontanément pour modifier les choix faits par le système, ou pour pallier à ses lacunes.

Mots clés :

Modèle de coopération homme/machine, intégration procédurale et sémantique des connaissances, planification hiérarchique de graphe d'operators, applications de Traitement d'Images.

Introduction

In this paper, various forms of cooperation are presented: cooperation involving experts from different domains, necessary for designing and maintaining knowledge-based systems, and Human-Machine cooperation used during the resolution of applications within these systems. The Image Processing system that we describe here is part of a larger project of a "Software workbench for the integration of knowledge in image processing and image understanding" [Revenu 94], aiming at automating the building of graphs of operators in order to design applications in digital image processing and image understanding. This system belongs to the more general class of "intelligent" systems for piloting operators [Clément 93],[Matsuyama 89]. In other words, solving an application is achieved by the dynamic building of a "program", using selection and linking of operators, and adjustment of their parameters. Operators are stored into run-time libraries and can be accessed, directly or through a command language. Users of such a system no longer have to program in the classical sense of the word, and they do not have to know the details of operators either. On the contrary, they must be able to accurately describe their objectives and the context of their application in the proper formalism [Dalle et al. 93]. As a matter of fact, intentions and context are taken into account at every level of the resolution process to direct and control choices.

In a more detailed way, the basic resolution process of our system is composed of five steps involving various kinds of knowledge:

(1) *Planning*: First a plan of actions adapted to the problem is generated. This plan is the medium for the selection of operators and their linking. It is built by decomposing a complex problem into a sequence of primitive actions that are actually feasible.

(2) *Instantiation*: This plan is then instantiated with operators chosen in the run-time library. Each primitive action of the plan is decomposed into a sequence of several operators, selected according to the context and the constraints associated to this action. Moreover, since operators have adjustable parameters, optimal values for each of them must be determined.

(3) *Execution*: Executing the plan corresponds to the running of each code in the order specified by the plan. Because it is generally hard to a priori find out the optimal values of parameters, the optimization stage must take place during the execution of operators, thanks to a trial-and-error process that aims at maximizing some rating function.

(4) *Evaluation*: The evaluation of intermediate results vouches for the relevance of final ones. This is achieved by quantitative or qualitative measurements of the differences existing between characteristics of the present results and what is expected.

(5) *Correction*: Evaluating results enables also to locate non-adapted parts of the plan and to diagnose the origin of errors, in order to propose corrections that take these errors into account.

The final result of the planning process is a complete program adapted to the application under study. This program is general enough to deal with all the cases of a class of problems described by the same objective within the same context, for which it was specially built.

In the field of Image Processing, such works are a relatively recent concern, on the one hand because they have to deal with complex tasks that require sophisticated software architectures, and on the other hand because there now exists a sufficiently wide enough corpus of applications.

The domain of Image Processing covers the whole set of operations aiming at reducing the quantity and increasing the quality of the information contained in images, in order to produce either more synthetic images, or specific evaluations in terms of numeric measures. No decision is taken at this level; only image-to-image transforms are used, without trying to understand the contents of image, thus considering the image as an objective source of information. Among Image Processing objectives, one can list tasks such as improving the objective or subjective quality of images, restoring deteriorated images, region-based or boundary-based segmentation, in hoping that regions will correspond to the objects to be extracted and lines to their boundaries, and also tasks of image compression for transmission or storage purposes. Figure 1 shows an example of a segmentation task.

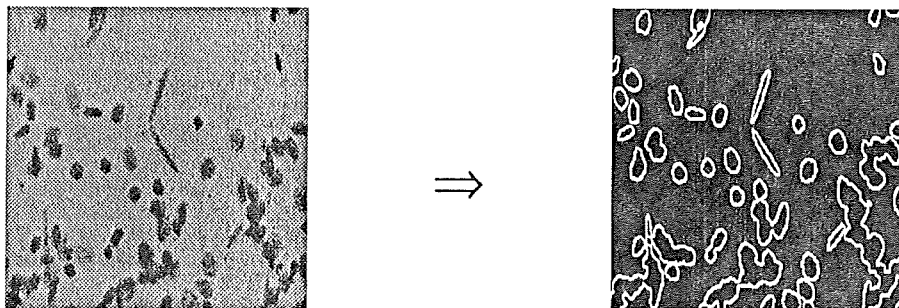


Figure 1: A segmentation of biomedical images of cytology, the task is "Isolate all cell nuclei of medium size and convex form without separating clusters of cells."

Most Image Processing problems belong to the mathematical class of ill-posed problems [Poggio et al. 85]. It means that if these problems can be put into a form such $Y = A \cdot X + n$, where Y is the image, X the scene under observation, A any kind of transformation function and n any kind of noise function, function A is not always invertible, and even if it is, it is not unique and does not continuously depend on the observed data; so a little noise on data can result in a great variability (instability) of the solution. In other terms, even if the state of the input image (Y) and the expected output image (X) are known, there exists no algorithm (A^{-1}), able to pass from the former to the latter in a single-valued and systematic way. It is thus necessary to introduce knowledge to direct choices and to control their application.

In the domain of Image Processing, there exists a large amount of algorithms that can be coded, together with numerous techniques and strategies that can be represented as pieces of knowledge. Of course, the major challenge is to know when to use them and what their actual effects on images are. In fact, this knowledge is difficult to extract and to formalize for several reasons. Firstly, the accumulated knowledge still remains relatively difficult to access because it is not structured and above all extremely diffuse. Moreover, there exist neither theories nor universal methods, on which one could rely in order to find out models of the domain. Finally, the thought processes of experts are relatively empirical. Experts are generally unable to explicit every step of their reasoning, for at least two reasons. The first one is due to the fact that problems are hardly impossible to specify completely from scratch, whatever the user may be. Secondly, knowledge about operators is imprecise and unreliable and there are often no quantitative functions for assessing results. Both reasons explain why one has to resort to many trial-and-error steps and also why initial objectives must often be revised and completed in the course of resolution.

The domain of Image Processing is a typical example of a domain where one has to develop systems establishing a cooperation between several high-level expertises. First and foremost, the definition of an Image Processing problem refers to an objective related to a specific application domain, whereas the problem is solved thanks to competences related to another quite different domain, i.e. competences in Image Processing. Owing to the complexity and fragility of the data handled by the system, it is essential, on the one hand, that the system can call on the expert of the domain of application, to interactively refine the terms of his/her request, and on the other hand that the Image Processing expert can enter the resolution loop, to take part into the development of a solution defined as a plan of actions. Moreover, the knowledge acquisition issue can only be envisioned through the resolution of concrete applications resulting of a cooperation between a domain expert, an Image Processing expert and a designer-programmer.

Rather than aiming at developing a general-purpose system, we are interested in setting up a workbench directed towards the cooperation between various agents involved in the system so as to provide assistance to the working out of some kind of computational theory of Image Processing, analogous to Marr's vision theory [Marr & Poggio 79]. Not only strategies, techniques, representations and outputs will have to be explicitly defined, but we shall also try to specify criteria that can be used to select and control an application in order to reach a goal under some specific context. In the first part of this article, we are going to present the motivations and objectives on which our work is based. In the second part, models of cooperation for knowledge acquisition and problem solving will then be detailed and discussed.

1. Conceptual model of an Image Processing system

During the development of an Image Processing system, the cooperation between several experts consists in using and respecting some kind of general thought processes. The resolution of an Image

Processing problem involves three categories of experts: experts from the domain of application (biologists, geographers), experts in Image Processing and designer-programmers. Domain experts are required to express clearly and exhaustively the purpose of their application, and also to provide several criteria that will be used during the a posteriori evaluation of results. Image Processing experts have to propose some kind of answer in terms of plans of actions adapted to the peculiarities of the request and the nature of images. Designer-programmers finally have to provide a framework for an efficient implementation.

1.1. Motivations

Our approach is motivated by the development of real-size applications, while being concerned by generic principles. We are addressing issues such as:

- The development of concrete applications with classical methods, in order to precise the nature of real problems and to establish an efficient dialogue between Image Processing experts and domain experts, that can be based on concrete results.
- The specification of a software architecture favoring cooperation. The objective is to get a system that can manage knowledge, i.e. to represent it and make it operational. It must be a framework for the development of applications and for the acquisition of knowledge on techniques, know-hows and knowledge on computer tools.

Knowledge acquisition must be achieved through many comings and goings between these two stages. The knowledge that can be extracted from running applications is injected into the knowledge-based system, where it can be tested and validated on numerous examples. Because the knowledge acquisition issue is a difficult one, we also propose a progressive knowledge integration according to two successive levels (Figure 2):

1. *Procedural integration*. It consists in programming at the “knowledge level” [Newell 82] and is based on the definition of an environment for designing applications, large enough to deal with any application where the concept of task is valid. This environment contains generic predefined operators and mechanisms to facilitate the building of syntactically correct linking of operators. The development of applications by an Image Processing expert then consists in selecting operators, adjusting their parameters and ordering them. Such an environment becomes the medium for communication and dialogue between the application designers, i.e. between the Image Processing expert and the domain expert [Serfaty 91].

2. *Semantic integration*. We aim at identifying and formalizing the expertise in Image Processing in explicit terms that can be understood by the experts and, at the same time, be used by the knowledge-based system. To that purpose, a model of knowledge and an expressive vocabulary must be proposed, that have an actual cognitive relevance. The knowledge corresponding to Newell’s “knowledge level” is seen as a medium for the communication between experts, so that they can exchange views, criticize and revise know-hows. Then, its operational version corresponding to the “symbolic level” must be used by the knowledge-based system. Thus, it seems necessary that the knowledge coded into the system, should reflect, at least partially, the knowledge

given by the experts [Aussenac et al 92]. Three types of partners are cooperating during the knowledge acquisition stage: the system offers a unifying framework for representing and using knowledge, the designer implements basic building blocks and the Image Processing expert, by conducting experiments with these building blocks, can produce an operational representation of his/her knowledge, which can then be further formalized and generalized if necessary.

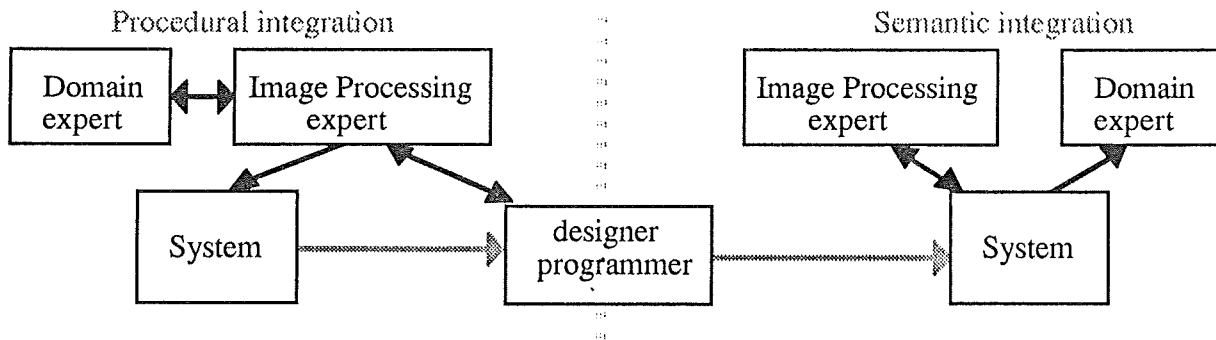


Figure 2: Procedural integration versus Semantic integration

1.2. Objectives of the system

The system design must answer two objectives, according to the expectations of the various types of users:

1. *Two functioning modes must be available.* In the former, the resolution is on the user's initiative. The user can dispose of the whole system environment in an ergonomic way, in order to solve his/her problem. At any time, the user can ask the system to automate the resolution of some specific task. In the latter mode, the user asks the system to take the problem-solving entirely in charge. The system is then working automatically, but if the user wishes to intervene, he/she can do it in order to dynamically modify the choices of the system or change strategies and their parameters.

2. *The user must be directly involved in the resolution of his/her problem.* System and user have to interact with each other so as to build the solution by a collaborative process [Baker 92]. In the collaboration paradigm, each party, system or user, jointly participates to the solving of the same problem. They solve it together and each of them intervenes during the working out of the solution, to accomplish the task he/she/it knows how to perform. The user comes into the resolution loop to the best of his/her abilities in the domain.

The system recognizes three types of users, that correspond to the three types of experts that have to take part to the resolution of an Image Processing problem. They are distinguished from each other by their abilities in Image Processing and their knowledge about the system itself. Each of them corresponds to one functioning mode of the system.

1. *The user skilled in the domain of origin of images.* The user uses the system in automatic mode. The user presents a problem through input interfaces and gets results from the output ones. Nonetheless, the user can be interactively questioned by the system to complete the terms of his/her request.

2. *The user skilled in Image Processing.* The user can dispose of the system as an environment to build his/her own plans of actions, to link programs from the run-time library or to ask the system to selectively solve some parts of the whole plan.

3. *The user skilled both in Image Processing and knowledge-based systems.* The user is then in a position to increase the knowledge base, or to modify it. The user is also able to spontaneously intervene in the course of a resolution session, by changing choices made by the reasoning module.

1.3. Resolution model

The conceptual model of the system need not be an exact simulation of the thought processes of experts. On the contrary, it must be operational and, at the same time, understandable by the expert who can revise, criticize and increase the knowledge-base. We are here only interested in Image Processing knowledge. Knowledge about the application domain cannot be explicitly represented into the system, because it is not re-usable. But, as it is essential to the resolution of applications, it must be acquired and represented in the course of resolution, by means of a formal language, only based on a symbolic description of concepts and properties belonging to the domains of Image Processing and mathematics.

We are now going to describe the mechanisms that enable the use of know-hows in Image Processing. In particular, we shall study the reasoning process defined as a decomposition into abstraction levels.

1.3.1. Knowledge organization

A request can be defined as an image, a set of goals to be reached and of constraints on these goals, and a context of application. These three kinds of data give information about distinctive features of the domain. The request can be as complex as desired. The context is given by describing the contents and characteristics of the class of images under study.

Our resolution model is close to the hierarchical decomposition principle called "Task/Method/Tool" [Deslandes & Revenu 94], [Moire & Revenu 94]. Moreover, it is based on an object representation of all these entities. The use of an object-based approach is not a mere programmer's trick, but it provides some semantics and a fixed representation, thus removing any interpretation ambiguity. A Task defines a specific problem to be solved and is associated to the description of a goal to be reached and its context. A Tool represents operational knowledge about a computational action. It describes its syntax, the accepted values for its parameters, and the nature of its inputs and outputs. We consider a Tool as a terminal task, and as it corresponds to some library program, it produces results. A Method characterizes a specific know-how to perform a task, by proposing a decomposition of this Task as a sequence of sub-Tasks and Tools.

The Task/Method/Tool paradigm can be applied to the control of domain knowledge, as well as to the control of control knowledge. Several potential functioning modes can thus be available: automatic, cooperative, or step-by-step.

1.3.2. Hierarchical nature of the domain and plan representation

To implement this paradigm, we have chosen to study the domain of Image Processing under five abstraction levels [Clouard et al. 93]. We distinguish, by decreasing order of abstraction, the request, task, design, procedure and operator levels. The first four levels can be identified as a further decomposition of Task concepts, whereas operators are Tools. The object-based representation ensures that each level of abstraction will have its own representation and the proper associated semantics.

A *request* decision represents the problem as it is given by the user. It contains the initial image, the user's intentions and the context of the application (e.g. Isolating the objects from the background without separating clusters of objects).

A *task* decision defines an elementary sub-problem related to some objective in Image Processing (e.g. Background elimination).

A *design* decision describes a general-purpose Image Processing technique, without taking into account technical implementation constraints (e.g. Object extraction by pixel classification).

A *procedure* decision characterizes an algorithm but a technical know-how about possible Image Processing operations that can be implemented as algorithms (e.g. Binarization based on minimizing inter-class variance).

Finally, an *operator* decision characterizes an executable algorithm of the library, by giving the type of its input data, the number, the type and the accepted values for its parameters, etc. (e.g. Binarization <low_threshold> <high_threshold> <input_image> <output_image>).

Each abstraction level can be seen as describing a version of the application, according to an abstraction level in Image Processing. The first two levels describe the problem to be solved as a decomposition into elementary sub-problems. The next two levels correspond to Image Processing techniques that can be applied to perform these tasks. And the last level contains programs and values for their parameters.

An application is represented as a plan of actions, taking the form of an AND/THEN tree of goals composed of five abstraction levels (Figure 3). The root of this tree is the request, then we successively find the tasks, designs and procedures. The leaves of the tree are operators. The links between nodes are precedence links within a same level, and decomposition links between levels of decreasing abstraction order. A decomposition of nodes consists in combining sequential links (THEN) or parallel links (AND) between the nodes of the next lower level.

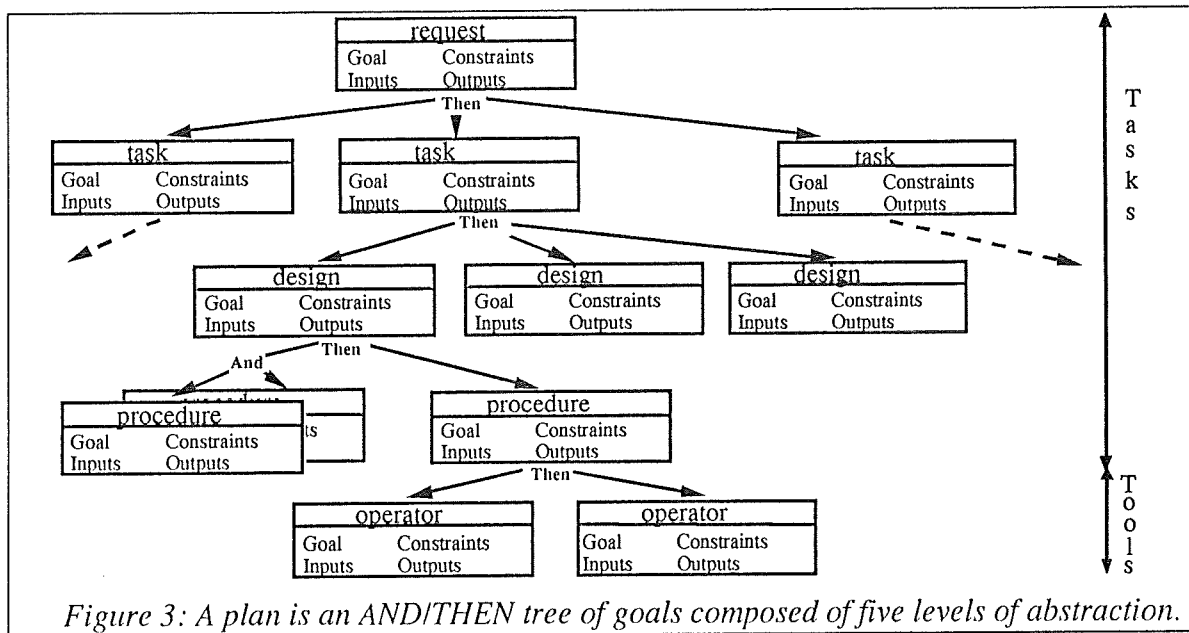


Figure 3: A plan is an AND/THEN tree of goals composed of five levels of abstraction.

Representing an application as a plan does not reflect the exact thought processes of experts. However, in Image Processing, as in many other domains, there exists an expertise about operators, that, although it is incompletely formalized, can help the process of selecting operators and choosing their parameters. A posteriori, human activities can always be represented in the form of plans [Suchman 87]. This model is thus both computational and understandable by experts.

2.3.3. Knowledge in Image Processing

With such a knowledge organization, processing an image consists in building a specific plan of actions taking advantage of the hierarchical levels of abstraction, then to instantiate this plan to get a graph of operators, to execute each operator and to assess their results by taking also advantage of the hierarchy. Results are progressively brought up the tree, and evaluated by different levels of criteria. Methods specify how to define a plan, to instantiate, execute and evaluate it. In Image Processing they refer to three categories of knowledge:

(1) *Knowledge about the domain of application* is a means to associate a sense and a subject to the image under study. They also bring the necessary pieces of information to understand the request. This knowledge can only be acquired from the user at the beginning of each new application, but also interactively owing to ambiguous and incomplete points that must be clarified by trial-and-error processes.

(2) *Knowledge about Image Processing* are in charge of finding out the right methods to perform Image Processing tasks, as well as evaluating their results. We distinguish knowledge about Image Processing methods on the one hand, and knowledge about the way to use them on the other hand.

(3) *Knowledge about operators* is used to direct the selection, parameter adjustment and linking of operators to achieve Image Processing methods. We distinguish syntactical knowledge, used to build syntactically correct sequences of operators, and semantic knowledge used for selecting and controlling these operators.

In the case of procedural integration, a great part of the Methods is the user himself/herself. In the case of semantic integration, making the planning knowledge operational is achieved thanks to a Blackboard architecture based upon the BB1 model [Hayes-Roth 85], and takes the form of Knowledge Sources.

2. Various forms of cooperation

In our system, two forms of cooperation are available, the first one is essential to the knowledge acquisition process, and the second one to the resolution of applications. Consequently, our system presents two faces, whether the user chooses the “manual” or “automatic” option. The manual option is simply a plan editor, that enables the Image Processing expert to manually build applications as they are prescribed by the domain expert. The automatic option corresponds to a planning process involving cooperative work between the designer-programmer, the Image Processing expert and the reasoning module.

2.1. Cooperation for knowledge acquisition

In this section, we are studying the case of “manual” resolution of applications. The Image Processing expert dynamically builds a plan of actions adapted to the solving of a specific problem submitted by the domain expert. The Image Processing expert appeals to the designer-programmer when new tools must be added to the system. During this stage, two kinds of cooperation must be taken into account: cooperation between the domain expert and the Image Processing expert, and cooperation between the Image Processing expert and the designer-programmer. This is an important stage, when new problems, never addressed before, are studied with a view to integrating new techniques that have just been made explicit, into the system. New concrete problems are thus solved, and the domain expert’s interest is aroused because he/she can count on rapid results. Moreover, as plans can directly be transformed into C++ programs, experts are provided with a software package that can deal with all the images of the class under study.

2.1.1. Building applications

In this first case, the system is used as a simple editor of graphs of goals. The Image Processing expert can thus create his/her own base of plans of actions, by linking together several Tasks and Tools. Plans are coded in a completely static manner: a task is associated to a set of design decisions, a design decision to a set of procedures and a procedure to a set of operators. This not a problem-solving issue but only high-level programming. Plans of actions can be totally or partially re-played on new applications, and they can also be revised and modified. In this static representation of plans according to the task/method/tool paradigm, methods simply correspond to the links between tasks and tools.

Within the working area, the Task and Tool browsers enable the Image Processing expert to select Tasks and Tools, link them together, and adjust their parameters. This working area takes up

the structure of our hierarchical decomposition: plans follow the five-level hierarchy and are represented as AND/THEN trees of goals. The working area can also be seen as the place where the Image Processing expert and the domain expert can dialogue with each other. Thanks to predefined plans of actions, it is possible to produce numerous intermediate images and to ask domain experts their advice about output images. In fact, images have the major advantage of being directly interpreted by both parties, in their respective languages, and they can then be used as a support to start a dialogue without the barrier of language constraints. So they allow the Image Processing expert together with the domain expert to make the specifications of their request progressively evolve, by many trial-and-error steps, and in the end to be automatically provided with the corresponding plans of actions. On the opposite, cooperation between users and the designer-programmer is only required when some library programs are missing.

During this procedural integration process, Image Processing techniques are specified as knowledge that is directly operational for the planning module, because the plans created "manually" share the same structure as the plans automatically built by the planning module. Conversely, control knowledge for the choice of these Image Processing techniques and the evaluation of intermediate results cannot be exhibited and made explicit during procedural knowledge integration.

Knowledge acquisition thus requires an intermediate stage when one has to specify the set of criteria that lead to the decomposition rules of each node of the plan, i.e. type of decomposition (AND/THEN) and principally, application and non-application conditions for this decomposition. The designer-programmer can then start writing Methods as Knowledge Sources, the condition part of which corresponds to the application condition for the decomposition of the node, and the action part will result in its specific decomposition. This knowledge acquisition process can only be achieved by a cooperation between the Image Processing expert and the designer-programmer. Then, once a first integration stage is over, the system is used as an experimentation tool when several plans related to the same request can be tried and tested, in order to evaluate their performance, to find reasons for their eventual failure or to propose improvements.

2.1.2. Means of cooperating

In the "manual" mode, the system's aspect is essentially a graph-editor that can be used to interactively build plans with the pointing-device, but there are also some secondary interfaces to visualize information about the system: list of tasks, list of tools, lists of the task/tool links. The plan is visualized as a tree, which can be manipulated with the pointing-device, as much as one likes, in order to print the contents of a node, execute some part of the plan, show input and output images, modify parameters, change links, etc. This graphical representation gives a global and synthetic overview of the plan of actions that is being built.

A plan is based on the Task model, which implies that it is generated by linking tasks and decomposing them into design decisions, then into procedures and operators. Owing to the fixed

Result assessment also takes advantage our hierarchy: results are progressively brought up towards hierarchical levels. Each operator, procedure, design or task decisions in the plan can be considered as responsible of assessing its own output. Operator decisions have to score the quality of their output images; design decisions the result of a sequence operator decisions and task decisions the result of a sequence of design decisions. A request is said to be solved when each task is scored with suitable results. Correcting errors is done by changing the sequence of design decisions or the sequence of operator decisions or the specifications of operator decisions that did not succeed.

In the automatic case, Methods are Knowledge Sources stored into our knowledge base. A Knowledge Source is associated to a unique goal and characterizes a unique way of decomposing a goal into sub-goals, by the means of a condition part specifying when this decomposition is possible. Each potential decomposition of a same goal thus results in one specific Knowledge Source.

The system applies a classical cooperation scheme: the user's intervention is on the system's initiative when some information about the request are missing (domain expert) or when the system declares being unable to continue (Image Processing expert), who wants to opportunistically modify choices of the planning module or simply to get extra information. When an intervention is on the system's initiative, it is the system's opinion that prevails and in all other cases, the user's opinion is prevailing.

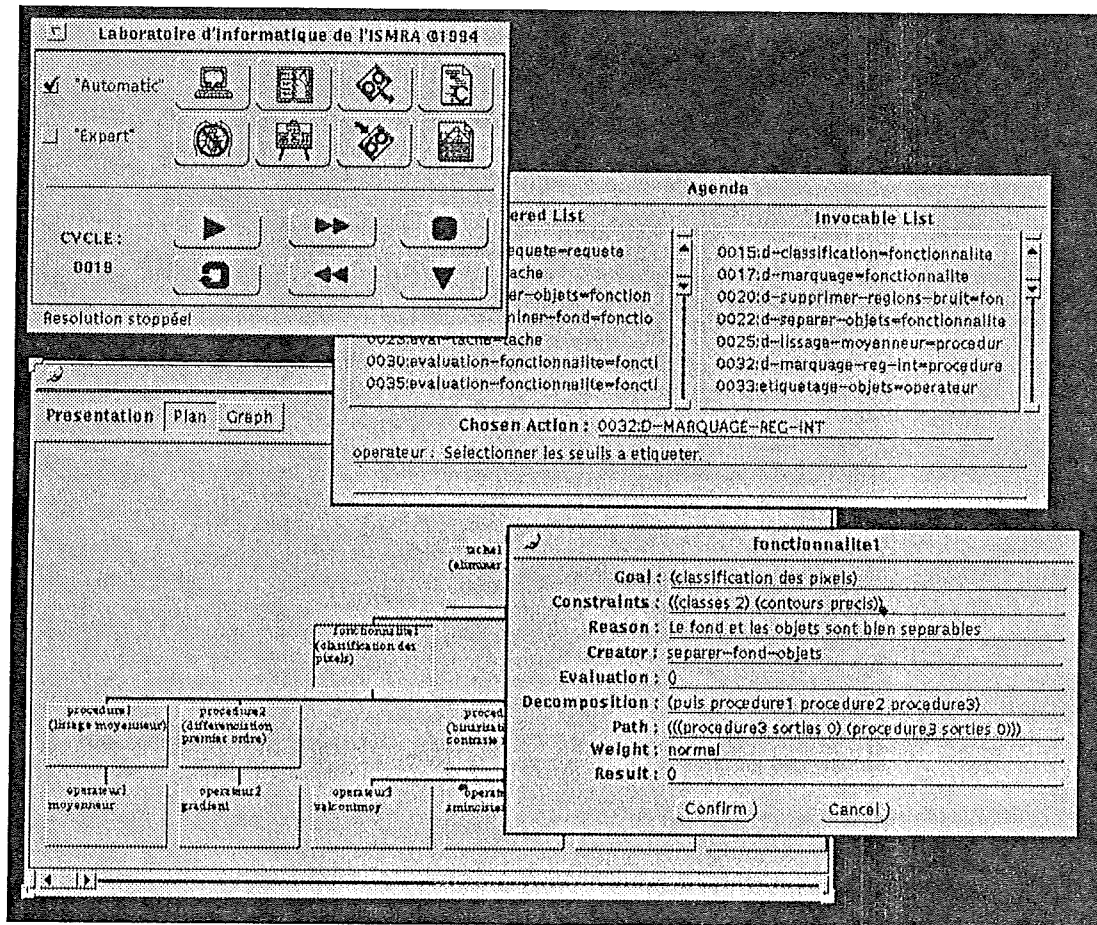
2.2.2. Means of collaborating

In order to make the collaboration between system and users possible, one has to design three interfaces adapted to the three categories of experts. They are implemented around the database which is the place where this cooperation can start developing, for it is the place where the problem is being solved.

1. A dialogue interface has been developed for managing the system-user dialogue and for visualizing results (images) obtained in the course of the resolution. It is used to initialize a resolution process, when the user has to describe his/her request. Then the system questions the user through it, in the course of the resolution, to complete the problem description or to raise some ambiguities. At present, this "dialogue" is performed by a hyper-text module, where a message-box is visualized by the system, in which keywords can be selected thanks to the pointing-device.

2. A graph-editor, already mentioned in section 2.1, enables the user to visualize and manipulate the graph of goals generated by the reasoning module. The importance of this interface must be emphasized, when one is concerned with the quest for efficiency in problem-solving. This kind of interface must be seen as some mediator between the functioning logic of the machine and the user's logic, so as to induce a real collaboration between them [Brutus 92]. Through the graph-editor, users can intervene in the course of a session, by interactively adding, deleting or modifying the contents or the linking of goals. This can be seen as the expert taking the resolution in hand for some time, the system taking back the hand later.

3. An explanation module must be developed so that users can have access to the solving module abilities. At present, we are only concerned with the means to justify and explain the system's behavior to the designer and the end-user. This competence is essential to the quality of a resolution system [Ovalle 91], as far as the production of results or the use of the system are concerned. It reinforces the user's confidence into the proposed solution, whereas a result alone is of little interest. It also makes the maintenance of the knowledge base easier, by pointing to incoherences and inadequacies. Finally, it favors knowledge elicitation, by compelling the expert to argue his/her choices and methods. In this respect the Blackboard architecture is the perfect architecture for explanation generation [Lemaire 92].



Hard copy of the interface showing the control panel, the graph-editor, a window describing a goal and the agenda of Knowledge Sources.

Conclusions

In this paper, we wanted to show that the specifications of our system's conceptual model are strongly motivated by the will to solve concrete applications. Our methodology is based on experience gained on the ground, by trying to bring the most satisfactory answer to the issue of designing new applications in a rapid and efficient way thus meeting the users' requirements.

The development of our software workbench that groups together two distinct problem-solving environments, makes experiments easier and arouses some cooperative work between three categories of experts: experts in the domain of origin of images, Image Processing experts and designer-programmers: The originality of the Image Processing domain stems to the fact that two high-level expertises have to cooperate, the domain expertise and the Image Processing expertise itself.

In the present state of the project, the planning module, the graph-editor and dialogue interface are operational [Clouard 94].

Our research takes place within the framework of the "Image Processing and Analysis" Center of Caen. This center covers a wide domain of applications, dealing with biomedical, as well as material or geological images. It constitutes a privileged field of investigation in order to validate and refine the ideas we are advocating and to make various experts cooperate.

References

- Aussenac-Gilles, N., Krivine, J.P. & Sallantin, J. (1992). L'acquisition des connaissances pour les systèmes à base de connaissances. *Revue Française d'Intelligence Artificielle*, Vol 6(1-2): 7-18.
- Baker, M. (1992); Le rôle de la collaboration dans la construction d'explications; *Proceedings of the 2^{ème} Journées Explication du PRC-GDR-IA*, Sophia-Antipolis, France, pp. 25-42.
- Brutus, P. (1992). *GLORIA: Génie Logiciel à base d'Objet et de Réflexivité pour le développement d'applications à Interface Avancée automatique*. Thèse de l'Université de Caen, France.
- Clément, V. and Thonnat. M. (1993). A Knowledge-Based Approach to Integration of Image Processing Procedures, *CVGIP : Image Understanding*, Vol 57(2) : 164-184.
- Clouard, R., Porquet, C., Elmoataz, A. & Revenu, M. (1993); Resolution of image processing problems by dynamic planning within the framework of the Blackboard model; *Proceedings of the SPIE's international symposium: Intelligent Robot and Computer Vision XII*, Vol(2056); Boston, Usa, Septembre, pp. 419-429.
- Clouard, R. (1994). *Raisonnement incrémental et opportuniste appliqué à la construction dynamique de plans de traitement d'images*. Thèse de l'Université de Caen, France.
- Dalle, P., Dejean, P. & Inglebert, J.M. (1993); Méthodologie de conception d'applications de traitement d'images; *Proceedings of the 4^{èmes} journées Orasis du GDR-PRC communication homme/machine*, Mulhouse, France, Octobre, pp. 92-95.
- Deslandes, J. & Revenu, M. (1994) ; Système d'acquisition de connaissances basé sur le principe de coopération. Application à l'aide de la création de variétés de blés hybrides; *Proceedings of the 9^e congrès RFIA, Paris, France*, Janvier, pp. 709-714.
- Hayes-Roth, B. (1985). A blackboard architecture for control, *Artificial Intelligence*, Vol(26): 251-321.
- Lemaire, B. (1992); Construction d'explications: utilisation d'une architecture de Tableau Noir; *Proceedings of the 4^e Journées PRC-GDR IA*, Sophia-Antipolis, France, Juin.
- Marr, D. & Poggio, T. (1979). A computational theory of human stereo vision; *Proceeding of the Royal Society of London, serie B(204)*, England, pp. 301-328.
- Matsuyama T. (1989). Expert Systems for Image Processing: Knowledge-based Composition of Image Analysis Processes. *CVGIP*, Vol (48) : 22-49.
- Moire, T. & Revenu, M. (1994). A software architecture based on a computational model of knowledge: Application to the integration of the method for projet development and management, *Proceedings of the First european conference of Cognition Science in industry*, Luxembourg, Septembre, pp. 37-52.
- Newell, A. (1982). The knowledge level. *Artificial Intelligence*, Vol(18): 82-127.

- Ovalle-Carranza, D.O. (1991). *Contribution à l'étude du raisonnement en univers multi-agent: KIDS, une application pour l'interprétation d'images biomédicales*. Thèse de l'Université de Grenoble, France.
- Poggio, T., Torre, V. & Koch, C. (1985). Computational Vision and regularisation theory. *Nature*. N°317: 317-319.
- Revenu, M. (1994). *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, France.
- Serfaty-Dutronc, V. (1991). *La programmation par "mécanismes": de la compréhension d'algorithmes en Traitement d'Images à leur conception*. Thèse de l'Université de Paris-Sud, France.
- Suchman, L. A. (1987). *Plans and Situated Action: The Problem of Human-Machine Interaction*. Cambridge University Press: Cambridge.