ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

The RIME Prototype

Philippe Mulhem and Catherine Berrut

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

The RIME Prototype

Philippe Mulhem

CLIPS - Institut de Mathématiques et d'Informatique de Grenoble

Grenoble, France

Philippe.Mulhem@imag.fr

Catherine Berrut

CLIPS - Institut de Mathématiques et d'Informatique de Grenoble

Grenoble, France

Catherine.Berrut@imag.fr

Abstract

In this paper we give a technical description of the PRIME system prototype. PRIME allows us to provide an operational side to the theoretical work we have done in the “Modelling and Multimedia Information Retrieval” (MRIM) team. PRIME is designed to provide a generic way to express the storing, manipulating and retrieval of multimedia data. These tasks are separated into two parts, namely the strict database tasks and the information retrieval tasks. This paper focus on this generic part, and we address more specifically the problem of managing and retrieving images. We describe the implementation of a medical application managing Magnetic Resonance Images based on the generic core.

1 Introduction

This article deals with the RIME prototype, called PRIME, that corresponds to an implementation of the theoretical work that has been done in the Laboratoire Génie Informatique of Grenoble, and now in the Communication Langagière et Interaction Personne-Système Laboratory of Grenoble, in the context of the RIME information retrieval system [1][2] [3][4][5]. In this paper we describe the problem we intend to address with PRIME, dealing with multimedia information. The target application of PRIME is a medical one.

As shown in Fig. 1, PRIME allows medical data to be handled in two ways : a “data base” way and, an “information retrieval” way. The advantages of this approach are twofold: an efficient access to multimedia data is allowed by direct use of a database system functionalities, and a content-based access is provided by the “IR side” of PRIME. This simple division is obviously only functional and the architecture of PRIME described in part 4.1 underlines the strong relationships between these two parts of the prototype.

In the context of PRIME, we mainly focus on the problems arising during the manipulation, storage and retrieval of still images. Though, a lot of theoretical and practical results already exist on textual data processing in IR, and the texts play an important role in the description of images [6]. PRIME integrates a retrieval process on texts based on a semantic tree representation of them [7], but this functionality is not yet provided for images, and this aspect is beyond our scope of interest here.

As we show in Fig. 2, PRIME is not application specific in its functionalities and in its architecture. On the contrary, all our work in PRIME includes a systematic analysis of the problems to distinguish the part of the general processing and the part of the needs of a specific medical application that plays the role of a “final product”. In fact, by validating our theoretical results, we intend to show our *savoir-faire* in the medical field, and in other fields as well. We provide the ability to **storing, manipulating and retrieving data**. As we will see below, these three simple words hide a lot of things; for instance, the O2 database system [8] is used in PRIME to store the images.

This paper has the following outline : we show in part 2 the general problems found in the literature concerning the management of images; in part 3, we describe more precisely the context of PRIME and the general elements on

Figure 1: PRIME objectives

Figure 2: Synopsis architecture of PRIME

which we focus in PRIME; then, in part 4, the architecture and the actual state of PRIME are discussed; in this part, we clearly separate what is general to image management, and what is purely dedicated to the medical context. The part 5 describes the future extensions of PRIME in a way to ease images indexing and to provide more information retrieval capabilities. We conclude in part 6.

2 General problems with image management

Applications that handle images face multilevel problems:

- First of all, the classical physical problem with images is the memory size needed by big collections of images. For instance, the Observatory of Strasbourg [9] uses a database of a hundred spatial images, and each image is about 1 Gigabyte sized. Medical images are also very demanding on secondary storage : an X- ray image of a thorax is 5 Megabytes sized, and a tomodensitometry is a 512 * 512 matrix of 12 bit pixels image, i.e. it needs about 400 Kilobytes to be stored.
- The second problem is the way to model and to represent the image content, and also the ways to retrieve these image data. We address this problem in this paper.

One way to access such data is to use contextual information like the author name of the shot (or the creator of a virtual image), the type and features of the tools used to generate the image, the date of creation, etc. This kind of information is referenced as “external features” in the following. In addition to these features, an image can be multiviewed. Roughly speaking, we consider at least 3 different points of view that are specific to the content of any image:

- there is first the physical point of view. Based on this point of view, we characterize images using their color (like in QBIC [10] for instance). This point of view gives the same importance to each pixel of an image. Vector formatted

images cannot fit into this point of view, because the vectors already represent groups of pixels. This physical point of view generates important noise when it is used alone for retrieval. For instance, if we indicate that we want images containing 75% of red colour (corresponding to a red apple in a user's mind), the retrieval system answers red cars, as well as red fruits, buildings, ..., without any discrimination.

- another point of view, called the "logical" one, is more related to groups of pixels that form consistent entities. This point of view handles contours as well as colours of entities, and textures. In the CAFIIRS system [11], dedicated to face photographs, eyes or mouth are entities of the images. In the same way, QBIC handles the retrieval of images based on sketches (handmade reference graphics). The use of this point of view improves the precision of query result compared to the sole use of the physical point of view. Thus, a shape is not discriminant in all contexts: a shape of a fish can be similar to a shape of bird, as is it shown with QBIC in [10]. Vector formatted images can use this point of view, but a system can also use more complicated shapes based on simple vectors, as it is shown in MULTOS [16].

- the last point of view deals with symbols related to the images, like in the Chabot system [12], or to entities of the images like in the RIME approach or the work of Meghini [17] where the symbols can be keywords or more complex representations. The interest of this approach is to allow queries using concepts, without having to work only in the image space. With this point, the difficulty is to represent really the images content, and not only texts that describe images. In the context of collections of very similar physical images, like Magnetic Resonance (MR) images, the symbolic view can play a role in the retrieval.

The three points of view above represent a big part of image content. The colours of images can be used in aerial photographs to detect specific types of ground. The logical point of view is used to find images "with a sunset" by retrieving images that contain a half red-colored circle. The symbol related point of view handle retrieval where "we see an iceberg". It has been shown [12] that the use of the physical level and the symbolic one improves the precision of a system compared to the use of the physical point of view alone. Moreover, the use of the logical point of view allows inferences on the relationships between entities of an image.

So, a generic image retrieval system has to deal with at least these three points of view of the images. For now, we do not consider others symbolic related points of view like the one dealing with religious or spiritual connotations (see [21]).

3 The context of PRIME

The context of PRIME is the storage and the retrieval of patient medical data. More precisely, we focus on the management of Magnetic Resonance Images of the brain. These original images are already in a numeric form, unlike X-ray images. The final users of PRIME are physicians. We then studied the specific needs of physicians who use these images, in a way to define the features of an appropriate image model. The needs that we choose to handle with the PRIME system are :

- during the analysis of an image that has given features, a specialist needs to find similar images, to help the study of the original image. For instance, a specialist may then validate the diagnosis he is making, by using some cases that are judged similar by the system. The figure 3 shows the context of this work.

Figure 3: PRIME context

- during research or teaching activities, a specialist needs sometimes to use images having some features for illustration. For instance, the physician can search for “MR images of the brain on the axial plane showing a tumor on the anterior part of the ethmoid cells”. In this case, the system can rank the retrieved images by the level of “anteriority” of tumors on the cells, or by the decreasing distance of a tumor to the ethmoid cells.

In the following we link the physicians needs to the classification of part 2. The symbolic point of view of medical images is needed. When a physician searches for a “tumor” or the “ethmoid cells”, he uses this symbolic level of the images. Relationships can also be used between the image entities. Composition relationships can be used at this level. For instance, the “anterior part of the ethmoid cells” is a structural component of the “ethmoid cells”. The logical level of the MR images may also be used. In such images, parts of the images (like “ethmoid cells”) have to be defined, as well as spatial relationships (like “IN”, “OVERLAP”, “TOUCH”, “EAST”, ...) between these entities.

4 PRIME description

As we have seen in the introduction, PRIME provides generic features for managing images, and the medical field is only one of the possible uses of this system. We describe PRIME in detail now.

4.1 PRIME architecture

PRIME aims at being a multimedia information retrieval prototype, but we focus here on image data. An IR system managing images must handle the representation, the storage, the indexing and the retrieval of images. As written in the introduction, we split these functionalities into two parts in Fig. 4, namely i) the database functionalities and ii) the information retrieval functionalities.

Figure 4: The two main cores of PRIME

We define database functionalities as features that can be handled by database systems at the present time. In this category, we put the representation and the storage of the raw images, as well as the basic operations available on these data. For images, these basic operations would be: contrast and brightness modification, zooming and scaling. Other generic operations are the export to a file and the import of images from a file. On these data, a DBMS query language is used to access images by external attributes, like the date of the medical examination. As we will see in part 4.4, this query language can also be used as a first step in providing simple content access image retrieval. The strict IR functionalities are related to the model of IR used, namely indexing, querying, query and document models, as well as the matching function between the image documents and the queries; some of these elements require also processing usually dedicated to a DBMS. The reasons why we use an object oriented database system are because the data manipulated by PRIME are structured and are inter-related, and because we want to provide generic functionalities that can be refined (using inheritance) in specific contexts. Relational database systems do not fit the

need of manipulation of complex structures and the indexing terms related to images can be rather complex as we described above.

The software decomposition of PRIME is :

- all the DB functionalities in PRIME including querying are processed by the O2 Object Oriented DBMS. The basic operations on images are provided by a toolbox written in O2C (the programming language of O2). This toolbox links specific software to the internal representation of media objects.

- IR functionalities (i.e. the querying, the indexing and matching function) are provided outside the O2 system when the database system is not well adapted, by software coded in C and C++. Our approach is however to consider that the elements of the indexing language that index the images are data that must be stored in the DBMS.

Then the global architecture of PRIME is shown in Fig. 5.

Figure 5: General Architecture of PRIME

4.2 Representation and manipulation of images

4.2.1 The generic modelling of images

We now describe the modelling of raw images. The following elements allow a generic description of the image data, without assuming any specific context.

The O2 system provides by default a class *Image* that defines :

- the structure representing an image. This structure stores the colormap and the bitmap image.
- a method called *loadfile* that reads a file and stores the image in the database
- a method called *display* that displays the image of the receiver on the screen

PRIME extends these facilities with a class called *Extended_Image* that relates the raw data to the semantic content of each image, and provides methods to process the basic operations on them. The *savefile* method generates a GIF format file containing the image of the receiver. Methods named *Contrast*, *Zoom*, *Scaling*, *Brightness* and *Rotation* display a window to acquire the parameters of the operation; then they call the method *Call_Toolbox* on a given object of the class *Toolbox*. This last method executes its task, and then generates a new image (a new object of the class *Extended_Image*) having the desired properties. Finally, the basic operation method displays the resultant image.

The *Call_Toolbox* method generates the appropriate data to process the wanted operation using a modified version of the *xloadimage* [13] software.

The representation of the semantic content of an image follows the approach of Mechkour [5]. This description is based on “Image Objects” (noted IO). In our case, an IO has several views : a structural one, a symbolic one and a spatial one. Fig 6 shows the description an image containing a house and a tree according to the model of Mechkour. The logical description of the image has 3 views : the SPAtial view, the SYMbolic view and the STRuctural view. The spatial view contains the graphical element describing the IO on the image (circle, ...), the symbolic view stores a description of the IO using symbol external to the image (House, ...), and the structural view is related to the composition of IOs (a tree *iscomposed* of a trunk and a foliage).

Figure 6: The generic model of images

In PRIME, we force the structural representation of an image to be a tree. The symbolic view of an image object is a string. These two elements are not considered as restrictions in our case, because it is always possible to have a root of the image objects structural tree corresponding to the whole image, and a textual description as a symbolic view is considered as a generic way to describe Image Objects.

The spatial objects that we provide now are points and rectangles. The spatial relationships, taken from the work of Mechkour, are vectorial ones (*EAST*, *WEST*, *NORTH*, *SOUTH*), topological ones (*IN*, *OUT*, *TOUCH*, *CROSS*) and metric ones (*CLOSE*, *FAR*). For now, these relationships are not weighted. Because of the limitations of the vectorial spatial relationships based only on the gravity center of spatial objects, we use also the *STRICT_EAST*, *STRICT_WEST*, *STRICT_NORTH* and *STRICT_SOUTH* spatial relationships. If we name $Eo1$ and $Eo2$ the set of points (i.e. a pairs (x, y) of coordinates) inside the objects $o1$ and $o2$, “ $o1$ *STRICT_EAST* $o2$ ” is defined as : $\forall p_1 \in Eo1, \forall p_2 \in Eo2 \cos(\widehat{p_1 p_2}) \geq 0$, where the point p is (abscissa of $p_1 + 1$, ordinate of p_1), and $\widehat{p_1 p_2}$ is the angle between the vectors $\vec{p_1 p}$ and $\vec{p_1 p_2}$ in the positive trigonometric direction.

The definition of the other strict vectorial relationships are like the one above. These relationships are more related to the “usual” notion of vector relations that the one proposed in [5].

The image objects, spatial relationships, symbolic views and spatial views are described by classes. This approach allows enhancement of the descriptions of images using inheritance. The structural view of an image object is an attribute of this image object, in a way to ease the structural traversal of images index.

In addition to these elements, the *Extended_Image* class has also a method called *Indexing*, that provides the display of a generic interface to index the receiver.

4.2.2 The specific representation of MR images

For the representation of MR images, we create a class named *MR_Image*, sub-class of *Extended_Image*. Each object of this class has an additional attribute that contains a reference to the series it belongs to. An MR examination of the brain is usually composed of several (between 1 and 5) series of slices according to a given axis. So, a serie of MR images is a logical group of images, and relating the images to their series allows a simple navigation between images that are similar from an external point of view (i.e. external criteria).

Using the inheritance relationship of the *MR_Image* class, the overriding of one method allows the creation of MR images as the result of each basic operation. The method *Indexing* calls the indexing interface we described in part 4.3. This indexing of MR images is a specialization of the generic indexing.

4.3 Indexing of images

In this part we describe the indexing of images. As we have seen above, a generic interface is provided as well as one interface dedicated to MR images. A complete description of the interface to the MR images has been presented in [19], and here we describe this specific interface. We show also, via the description of the specific interface, the generic interface. We consider two principles that we integrate in the proposed interface :

1- The image model that is built using the interface is explicitly shown to the user on screen, i.e. the user knows and understands what is generated during the indexing process.

2- Most of the indexing task has multiple subtasks that can be done in any order. This is defined in [14] as nondeterminancy: “Nondeterminancy gives the user flexibility in choosing which task to perform when”. In our interface, the indeterminacy has to be preserved in such a way as to ease the indexing task.

Figure 7: The images and textual windows

For this description, we use Figures 7 and 8 and we focus on the visible items of the interface. The interface is composed of four windows : the textual report window (bottom of Fig. 7), the initial image window (upper left window in Fig. 7), the image working window (upper right window in Fig. 7), and the indexing window (Fig. 8). We now describe these windows in detail.

The initial data windows are the textual report window and the initial image window. This medical report is provided to ease the indexing and helps in the description of the symbolic parts of image objects. The window containing the initial image, i.e. the whole image that we want to index, is kept intact as a reference. It is necessary to keep the initial image during the indexing phase, to retain the context of the part on which we are zooming. In the generic interface, the initial image window is present, because we always need to see the context of the window part we work on. The textual report window of the specific interface is in fact a textual window that the generic interface provides as help during the indexing task.

The working zone at the physical level of the image is the "Image Working Window". This image contains the representation of the spatial objects of the IOs. We highlight two points for this window :

- in its upper left part, an arrow with the character 'N' indicates North. This indication is used to determine vector relationships between IOs. Initially, north is up (as in Fig. 7), but after a clockwise rotation of 90°, the up arrow indicates the right part of the window.
- the "IMAGE" button, in the upper right part of the window permits the selection of the IO corresponding to the root of the IO tree. This button reflects the persistence of the corresponding IO, because it can not be destroyed.

The indexing window has 7 parts, numbered from 1 to 7 in Fig 8. It is the core of the indexing process and permits the instantiation of the elements of the image model. Moreover, we add some properties to the interface in a way to improve its ergonomics.

We begin by the description of the parts directly dedicated to the indexing. They are numbered from 3 to 6 in Fig. 8.

Part 3, called "GRAPHIC", contains elements dedicated to the indexing. The icons +, and □ allow the creation of the graphical elements, respectively crosses and rectangles, corresponding to the spatial objects of the image. The ↗ icon allows the selection of graphical elements, and the "eraser" icon allows the deletion of a selected spatial object and invokes the deletion of the corresponding IO. In this part, we can operate on the physical part of the image, by zooming, etc. For future work, a "Form_Rec" button will call a contour detection processes.

Part 4, called "SEMANTIC", is dedicated to filling the symbolic part of the selected IO. This filling is done using text.

The button "Erase" is used for clearing text, and the "Construct" button calls the pre-existing natural language processor of RIME, to generate the semantic representation of the text. The construction of the semantic representation of the text is a necessary phase to really instantiate the symbolic part of the current IO.

The "Edit" menu provides the usual functions to manipulate text : Copy/ Cut/ Paste. By default, we choose to define the symbolic view of the root IO as the entire medical report associated with the image. This choice offers the same functionality than the previous version of RIME.

The part 5, titled "STRUCTURAL", shows the tree of the image objects.

Each IO, except the tree root, has a name which is the string "Object_" followed by a unique number generated by the system. The tree root has the name "IMAGE".

The "Erase" Button allows the deletion of a link between two nodes (corresponding to two IOs). We can build new links by indicating the "Source" and the "Target" of the links.

The "SPATIAL" zone deals with spatial relationships between image objects. These relationships are displayed under a list of *<name of the source IO> <name of the relationship> <name of the target IO>*.

We can delete spatial relationships between IO by using the "Erase" Button. In this zone, we create relationships by selecting the "Source" IO, the "Target" IO and the "Relation" between the IOs.

By default, all the IOs (except the tree root) are related to the "IMAGE" IO by an "IN" topological relationship, because the spatial object of each IO of the image is in the image.

With the "Visualize" menu, the user can display the existing relationships in the image. We can stress the displayed relationships, by their types and their related image objects.

In accordance with the model described above, the "STRUCTURAL" part is dedicated to the IO tree, the "GRAPHIC" part deals with the graphical elements of the spatial objects of the IO, the "SPATIAL" zone concerns the spatial relationships between IOs, and the "SEMANTIC" part defines the symbolic view of IOs.

In this interface, we minimize the number of buttons and menu names. Destructions are carried out by an "Erase" button (or the "eraser"). The building of parts of IO, are directed by the "Construct" Button, and by "Source" and "Target" when necessary. At each level, the more we go from left to right, the more we find specific elements of a part.

We next describe 2 zones that give help to the physician the image is indexed. In part 7, the "DIALOG MANAGER" accounts for each system task, and it indicates the possible errors made by the user during the indexing process.

The "STATES" part is more model-related. It indicates, for each part of the model (structural, symbolic and spatial),

Figure 8: The indexing window

if the system considers that the indexed image is in a satisfactory state or not. From a graphical point of view, this part includes three “status men”, one per axis of the model. A status man is : i) “happy” if the image verifies the model and a set of a priori conditions, ii) “neutral” if the image verifies the model, and iii) “unhappy” in other cases.

For the semantic part, the status man is neutral if each IO has a symbolic part. The status man is happy when every text of the IO has been processed to extract its semantic content. For the structural status man, it is either happy (the IO are organized in a tree), or unhappy (the structural view is not a tree, but this state is tolerated during the indexing process). The spatial status man is neutral only if defaults spatial relationships (i.e., “Object.i IN IMAGE” relationships) exist. It is happy if it exists at least one spatial relationship created by the user who performs the indexes.

We finally have, in part 1, the general elements of the indexing window. “Save” and “Quit” for respectively saving the result of the indexing process, and exiting. For now, we only accept the exit if the image is in a consistent state with respect to the model (that is, no status man is unhappy). We can “Undo” the previous action at every part of the indexing window. The “Automatic” button will, in future releases, allow more assisted indexing processes, but this automatic button is only usable on specific cases.

The generic interface provides automatic filling of symbolic view of the root image object by a given text. It does not provide any analyzer for the symbolic part of the IO.

4.4 Retrieval of images

The retrieval of images that is now available in PRIME is only database oriented. Though, we provide facilities to process database queries on the semantic content of images described as IO trees. The goal of this facility is to provide a fast filtering of the image objects that match a query. The image generic querying is provided by a set of methods of the *Image_Object* class. These methods return a boolean or an integer result.

We describe first the generic way to access to images using the database query language. We focus first on the symbolic view of IO, then on their spatial view, and finally on the use of the structure of the IO.

- For the symbolic view, we propose a query facility dealing with the raw text of the view. The method, described as *Contains_string(ch : string) : boolean*, finds if the string *ch* is included in the textual content of the IO. This allows to find IOs containing the string "ethmoid cells", for instance. We can also combine several calls to *Contains_string* with boolean query connectors to find IOs having a symbolic view that contains "tumor" and "ethmoid cells":

```
select ob
from ob in IO_Base
where ob->contains("tumor")
and ob->contains("ethmoid cells")
```

- The spatial view of an IO contains two parts, namely the spatial object representing the IO on the image, and the spatial relationships between the spatial view of one IO and other IOs. The usable criteria on the spatial objects are i) the spatial object type with methods *Ispoint* and *Isrectangle*, and ii) features of these spatial objects, like the *Min_Width* and *Max_Width* according to the X-axis and Y-axis, or the *Surface* method that gives the surface (in *pixel*²) of the spatial object of an IO. If we look for an opacity that is a rectangle and having a surface of 23 *pixel*²:

```
select ob
from ob in IO_Base
where ob->Isrectangle
and ob->Surface = 23
```

The spatial relationships between IOs are a key point when we look for images, and that is why we focus on this problem in the following. As described in [20], it is not possible to handle all relationships of all elements of an image and so we make a classification of the generic spatial relationships between IOs; they can be i) stored relationships, ii) deduced relationships, or iii) computed relationships. The stored relationships are the ones that have been indicated during the indexing process of an image, and those we then in the database. The deduced relationships are the ones that we can, using given inference rules (see [18] for example), compute from the stored spatial relationships. The computed relationships use the characteristics of the spatial objects (gravity center coordinates, length) to compute all possible relationships between image objects. The ways of handling these different spatial relationships depend greatly on the context: when we consider that only the stored relationships are important, you can limit the query processing to these relations, whereas in other cases the use of deduced relations is sufficient, and if no spatial relationships is provided by the indexing, the computed ones are the only ones usable. In PRIME, access to the stored relationships from the spatial view of an IO is provided by the following method of the class *Image_Object* : *Stored_Relation(ch : string) : set(Image_Object)*.

This method returns the IOs that have their spatial view related (by stored spatial relationships) to the spatial view of the receiver. The deduced relationships obey several rules. We describe some of these rules by using a notation *ala* PROLOG Edinburgh syntax : "head :- body", and in the following o1, o2 and o3 are IOs, but it is obvious that the predicates deal with the spatial views of these objects.

Several deduction rules apply :

- some deduced rules are inversion rules, like "EAST(o1,o2) :- WEST(o2,o1)"
- some spatial relationships are transitives, like "EAST(o1,o3) :- EAST(o1,o2), EAST(o2,o3)"
- the IN relationships combined with strict vectorial relationships allow more complicated deductions : "STRICT_EAST(o1,o3) :- IN(o1,o2), STRICT_EAST(o2,o3)", also formulated as "STRICT_EAST(o1,o3) :- STRICT_EAST(o1,o2), RECOVER(o2,o3)".

Some deductions also depend on the type of the object, the following one is only valid for points :

“STRICT_EAST(o1,o2) :- EAST(o1,o2)”.

It is clear that we did not focus on a minimal base of spatial relationships, because we prefer to keep things open to keep things generic.

The deduced relationships are computed by a method *Deduced_Relation*(*ch* : *string*) : *set(Image_Object)*. This method gives the objects that are related, using stored or deduced *ch* spatial relationships to the receiver.

The computed spatial relationships are determined by the method *Computed_Relation*(*ch* : *string*) that returns a *set(Image_Object)*. It is a strict application of the definitions of the relationships given in [5] extended by the strict spatial relationships.

For example, if we want to find OIs that contains “ethmoid cells” and that are at the STRICT_EAST of an IO having a symbolic view containing an “tumor” using deduced spatial relationships, the query is :

```
select ob2
from ob2 in (flatten (select ob1->Deduced_Relation("STRICT_EAST")
                    from ob in IO_Base
                    where ob->contains("tumor"))))
where ob2->contains("ethmoid cells")
```

- The structural representation of the IO can also be used to group results. We can related this approach to the retrieval of complex documents. Using the above querying facilities, the system answer contains IO that are composing (using the structural links) others. We propose two types of simplification, one is specifically IO oriented, while the other is a more generically IO oriented. When two IOs of the same image, o1 and o2, answer a query, and when o2 is a descendant of o1 from a structural point of view, the specific-oriented simplification gives only o2, and the generic-oriented one gives o1. In fact, this simplification does not influence in our case the fact that an image is an answer for a given query, but this simplification can give additional information to an interface that shows the interesting parts of an image.

In this part, we do not indicate specific elements of the medical application, because the generic part is not extended. But, in the medical field of MR images of the brain, usable additional knowledge can be taken from an atlas of the brain, allowing the automatic retrieval of spatial relationships by using the axis of the slice and the map of the brain. For instance, the ethmoid cells touch the ethmoid sinus.

5 Future work

We describe here how PRIME is going to be extended.

Firstly, the indexing of images has to be more automatic to be usable on numerous images. Automatism can be integrated during the indexing process in several ways:

- recognition can be processed on the whole image and find shapes. To define image objects, the user could draw contours of objects or choose one of the shapes found in the image. In fact, this approach makes the hypothesis that this contour extraction is possible. In the context of medical images like MR images of the brain, such automatic extraction is not yet available.
- recognition can be directed by the user. When the process of an entire image is impossible, the user can ask the system to find shapes in a given part of the image. For instance, the QBIC system [22] uses snakes that follow the contours of one element of an image.
- when a contour is founded, the system should propose to put the corresponding image object in the most probable place in the image objects hierarchy.

The work described here focus on the database approach of PRIME. Some information retrieval functionalities are going to be integrated for the retrieval of images. We have seen above that we provide three ways to retrieve images using the spatial relationships they contain: we can use stored relationships, deduced relationships and computed relationships. Information retrieval systems intend to give the better answers to a user’s query, and one way to classify images according to their spatial relationships is to give first the images in the following order:

- firstly the images that store the query spatial relationships,

- secondly the images in which the deduced spatial relationships match the query relationships
- thirdly the images in which the computed relationships in the images match the query relationships.

Depending on the result of a stage and on the context application, the next retrieval step should be computed or not.

The structure of the logical view of images can also be matched with the structure described by a query. This process needs to integrate some kind of deduction about the transitivity of the composition of image objects.

The integration of the work already done for the matching of semantic trees will provide much more capabilities on the textual matter of the image objects. For instance, a image containing an "tumor on the left part of the brain" in its textual description matches a query dealing with "tumors on the brain".

We have also to find how the combination of these different representations of knowledge on the images can be mixed, and the conceptual graphs formalism seems to be a good way to achieve this goal, as shown in [5].

Because PRIME is not only dedicated to images, we work on the retrieval of entire medical reports composed of images and texts. This point has to manage the integration of the index of several types of data.

6 Conclusion

We described in this paper the PRIME prototype, that provides a generic way to see the management and retrieval of images. An important point of this work is that our approach is reusable for other media than images (such an approach for texts exists in a earlier version of PRIME). The problem of audio or video data is not for now our scope of interest, but are convinced that our approach is reusable for such data. From a database point of view, we use an object oriented system to ease the extension of this work to specific cases, as we have shown here in the medical context. Because we are not physicians, and because the functionalities of medical applications described here is quite simple for the moment, we do not claim to be comparable to specific systems like Kmed [15], but our generic approach seems to us a promising way to explore.

The future work that is planned for PRIME wants to keep the genericity that will allow the reuse of PRIME to avoid from scratch applications design and coding.

References

- [1] C. Berrut, Indexing Medical Reports : The RIME Approach, Information Processing & Management, vol. 26 (1), pp. 93-109, 1990
- [2] C. Berrut and Y. Chiamella, Indexing medical reports in a multimediaenvironment: the RIME experimental approach, ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, Massachussets, USA, June 25- 27, pp. 187-197, 1989.
- [3] J. Y. Nie, An outline of a general model for information retrieval system, ACM SIGIR International conference on research and development of information retrieval, ed. Y. Chiamella, Grenoble, France, pp. 495-506, 1988.
- [4] C. Berrut and M. Mechkour, Representation of images in databases: a preliminary study, IFIP-WG2.6 2nd Conference on Visual Database Systems, Budapest, October, pp. 317-326, 1991.
- [5] M. Mechkour, An Extended Model for Image Representation and Retrieval, DEXA'95. Database and EXpert system Applications, London., September, pp. 395-404, 1995.
- [6] R. K. Srihahi, Computational Models for Integrating Linguistic and Visual Information: A Survey, Artificial Intelligence Review, Luwer Academic Publisher, Vol. 8 (5-6), pp 349-369, 1994-95.
- [7] P. Mulhem and M.F Bruandet, On Objects Matching, Second International Conference on Information and Knowledge Management, Arlington, Virginia, USA, pp. 49-56, 1993.
- [8] O. Deux and al., The O2 system, CACM, Vol. 34 (10), pp. 32-48, 1991.
- [9] <http://cdsweb.u-strasbg.fr/Aladin.html> (Astronomical Observatory of Strasbourg, France)
- [10] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz and R. Barber, Efficient and Effective Querying by Image Content, IBM Almaden Research Center Technical Report RJ 9453 (83074), San Jose, USA, August 1993.

- [11] J. K. Wu, B. M. Mehtre, Y. H. Ang, H. H. Loh, P. C. Lam, and A. D. Narasimhalu, Facial Image Retrieval, Identification and Inference System, ACM International Conference on Multimedia, October, Anaheim, CA, USA, 1994.
- [12] V. E. Ogle, M. Stonebraker, Chabot : A System for Retrieval from Relational Database of Images, Research Report, Sequoia 2000 Project, University of California, Berkeley, 1994.
- [13] Xloadimage : <http://world.std.com/jimf/xloadimage.html>
- [14] L. Bass and J. Coutaz, Developing software for the User Interface, Addison-Wesley, 1991.
- [15] W. W. Chu, A. F. Cardenas and R. K. Raira, KmeD : A Knowledge Medical Distributed Database System, Information System, Elsevier Science, Vol. 20 (2), pp. 75-96, 1995.
- [16] Multimedia Office Filling : The MULTOS Approach, C. Thanos ed., North-Holland, 1990.
- [17] C. Meghini, An Image retrieval Model Based on Classical Logic, ACM-SIGIR, ACM Press, Seattle, Washington, USA, July, pp. 300-308, 1995.
- [18] A. P. Sistla, C. Yu and R. Haddad, Reasoning about Spatial Relationships in Picture Retrieval Systems, 20th VLDB Conference, Santiago, Chile, pp. 570-581, 1994.
- [19] C. Berrut, P. Mulhem and P. Bouchon, Modelling and Indexing Medical Images : the RIME approach, HIM '95, Univ.-Verl. Konstanz, April, pp.105-115, 1995.
- [20] A. Pentland, R. W. Picard, S. Sclaroff, Photobook: Tools for Content-Based Manipulation of Images Databases, SPIE Conf. On Storage and Retrieval of Image and Video Databases II, No. 2185, San Jose, CA, USA, February 1994.
- [21] C.H.C. Leung, J. Hibler and M. Mwara, Content-based Retrieval in Multimedia Databases, Computer Graphics, Vol. 28 (1), February 1994.
- [22] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petrovic, D. Steele, P. Yanker, Query by Image Content: The QBIC System, IEEE Computer, Vol. 28, No. 9, pp. 23-32, September 1995.