

Temporal Knowledge Representation and Organization for Case-Based Reasoning.

I. Bichindaritz

LIAP-5, UFR de Math-Informatique
Université René Descartes - Paris 5
45 rue des Saints-Pères
75006 Paris, France

E. Conlon

LIAP-5, UFR de Math-Informatique
Université René Descartes - Paris 5
45 rue des Saints-Pères
75006 Paris, France

Abstract

This article presents the temporal knowledge representation and its organization in a case-based reasoning system called MNAOMIA. The general case-based reasoning methodology is grounded on a model of reasoning such that memory, learning and reasoning are inseparable. This particular focus forces pertinent knowledge representation and organization in the system memory. The main aspects of the temporal dimension of knowledge representation and organization of MNAOMIA are detailed in this paper, such as: the temporal representation language for the cases, the automatic learning of trends from the cases during the reasoning process, and the organization of the memory in generalization-based hierarchies of trends under which the cases are indexed.

1 Introduction

Case-based reasoning [10] is an artificial intelligence methodology for the processing of empirical knowledge. It reasons from cases, which are sets of empirical data, such as patients cases in a medical domain. Previously processed cases are stored in a case-base, or memory, and used by such a system to process new cases. The processing of a new case uses one or several memorized cases similar to the new case. The specificity of the case-based approach of reasoning lies in its focus on the inseparability of reasoning from memory and from learning. In particular, the organization and the composition of the memory, giving it the ability to be accessed and updated constantly during the reasoning process, are a key issue for a case-based reasoning system, even when the temporal dimension is not addressed. The description of the knowledge representation language and of its organization permits to describe the memory adequately, and is thus the main subject of this paper, in relation with the temporal dimension.

The case-based reasoning (CBR) system presented here, called MNAOMIA, is a complete case-based reasoning system applied to the domain of eating disorders in psychiatry. Its aim is to provide assistance to experts in the different cognitive tasks they perform, namely diagnosis, treatment planning, patients

follow-up and clinical research [3]. A first version of the system, which did not deal with temporal data, was able to provide assistance to the work of the clinical team. The second version of the system, the memory of which is presented here, has been enriched with the processing of temporal data.

The second section presents the temporal knowledge representation of the cases and the trends. The third section deals with the composition of the case memory and with its organization around trends, and the fourth section explains how trends are learnt. The case-based reasoning process of the system is then briefly presented in the fifth section. It is followed by a comparison with related work, then by the conclusion.

2 Knowledge Representation

2.1 Case Representation

Initial Case Representation. A case represents a patient evolution over time. It is a conjunction of representation elements, each one associated to a definite date. Some of these elements may be fixed in time, such as the surname, others can be recorded periodically, such as the weight for every day, else others can be recorded at different dates, such as the result of a diagnosis from an evaluation. As stated before, an evolution is made up of the conjunction of all the above elements, also called events (see Figure 1). An event is any fact about the patient recorded at a given time.

$Event = \langle Time, Attribute, Value \rangle$

For example :

Surname Event = $\langle 15/06/95, Surname, Catherine \rangle$
Weight Event = $\langle 21/06/95, Weight, 38.9kg \rangle$
Diagnosis Event = $\langle 17/06/95, Diagnosis, Anorexia \rangle$

Abstract Case Representation. An abstract case represents a patient evolution over time, such as given in the initial case representation, but in a summarized way. A sequence of similar events can be summarized as an interval where the value of the attribute is rising or falling or remaining constant. An interval is described thus:

$$El_i = \langle t_{i,s}, t_{i,f}, lb_i, ub_i, Att_i, F_i(t_{i,s}, t_{i,f}) \rangle .$$

where $t_{i,s}$ is the biggest of the starting times, $t_{i,f}$ is the smallest of the final times, lb_i is the lower bound, i.e. the smallest of the starting times, ub_i is the upper bound, i.e. the biggest of the final times, calculated from the set of all the intervals generalized for the same attribute Att_i .

3 Memory Representation

3.1 Memory Composition

The memory M is a network of cases c_i and trends T_j linked by relationships R_k , i.e. *more-specific-than* between trends, and *instance-of* between trends and cases. Cases and trends are nodes in this network.

$$M = (T, K, R)$$

where $T = \{..T_j...\}$,
 $K = \{..c_i...\}$,
 $R = \{..R_k(n_1, n_2)...\}$, with $n_1, n_2 \in (T \cup K)$.

A new patient case presented to the system must be matched to memory and inserted in memory where there is a correspondence between one of memory's trends or cases and the new case. So a matching predicate must be defined between a case and a trend. Close versions of this predicate will also be used to match cases between themselves, as well as to match trends between themselves.

3.2 Trend Formation

A trend T is a conjunction of representation elements El_i .

$$T = \bigwedge_i El_i$$

Each element is the generalization of one of the element types of a case representation. When these element types are not repeated thru time, they are generalized by themselves. So a trend representation has the three basic types of representation elements as an abstract case representation : time-independent attribute-value pairs, time-points attached to attribute-value pairs, and time-intervals attached to attribute-value pairs. But a trend generally has two other types of elements, generalized from the previous ones, for a given attribute Att_i :

1. **A value attached to a time-point** is generalized as the mean of all the attribute Att_i values Val_j , which fall within the original time-point t_i range, called *timerange*(Att_i) :

$$El_i = \langle t_i, Att_i, \sum_{j=1}^n Val_j / n \rangle .$$

More precisely, each time-point t has a (min,max) range for matching. This depends on the time type Typ (such as *Year*, *Month*, *Week*, *Day*, etc.) and on the attribute, and is returned by the function *timerange*(Att, Typ). Then the time-point range is (min= $t - \text{timerange}(Att, Typ)$, max= $t + \text{timerange}(Att, Typ)$). By simplification, this function is also noted *timerange*(Att).

2. **A value attached to a time-interval** has two components to be generalized, the interval and the value of the function on the attribute.

Generalization of time-intervals. Let int_i and int_j be two intervals :

$$int_i = \langle t_{i,s}, t_{i,f} \rangle, int_j = \langle t_{j,s}, t_{j,f} \rangle .$$

The intersection between two intervals, $int_i \cap int_j$, does not capture the overall generalization of the previous intervals. A reference to previous intervals is needed, i.e. the knowledge that int_i and int_j produced the current generalized interval. An effective means of keeping track of the intervals that make up a generalization is to store the highest and lowest time-points previously met, called the lower bound and the higher bound. So the generalization of the intervals int_i and int_j is the interval int_k (see Figure 2) given by :

$$TrendInterval = int_k = \langle t_{k,s}, t_{k,f}, lb_k, ub_k \rangle$$

where $t_{k,s} = \sup(t_{i,s}, t_{j,s})$, $t_{k,f} = \inf(t_{i,f}, t_{j,f})$, $lb_k = \inf(t_{i,s}, t_{j,s})$ and $ub_k = \sup(t_{i,f}, t_{j,f})$.

This gives the interval a range within which it can

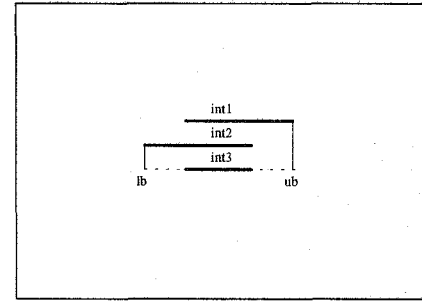


Figure 2: Generalization of two intervals.

expand. If the new patient interval's start point is within the lowerbound, the intersection is extended to this startpoint, similarly with the upperbound. These bounds allow the system to take previous instances into account when generalizing.

What is required for a match of two intervals is given by the operator *during* [1] :

$$int_i \text{ during } int_j \iff ((t_{i,s} \geq t_{j,s}) \wedge (t_{j,f} \geq t_{i,f})).$$

Upperbound and lowerbound are not used during the actual match. They are used when generalizing a trend interval that matches.

Generalization of the value attached to a time-interval. The generalization is the mean value of all velocities matched against this time-interval. To generate this it was necessary to store the sum of all velocities and the count of values. So :

$$TrendValue = \langle \sum_{i=1}^n Val_i, n \rangle$$

So the overall generalization for an interval bound to a numeric value is :

$$El_i = \langle t_{i,s}, t_{i,f}, lb_i, ub_i, Att_i, \sum_{j=1}^n F_{j,i}(t_{j,i,s}, t_{j,i,f}), n \rangle$$

The overall generalization for a qualitative value attached to a time-interval is :

$$El_i = \langle t_{i,s}, t_{i,f}, lb_i, ub_i, Att_i, G_i(t_{i,s}, t_{i,f}) \rangle$$

$G_i(t_{i,s}, t_{i,f})$ in this case is a qualitative value which cannot be generalized any further. We can assimilate the two representations if we define (this simplification is only for the clarity of this text, it is not used in the actual system where keeping n explicitly is required) :

$$G_i(t_{i,s}, t_{i,f}) = \sum_{j=1}^n \frac{F_{j,i}(t_{j,i,s}, t_{j,i,f})}{n}$$

3.3 The Matching Predicate to Relate Cases to Trends

In the paradigm of reasoning by analogy, which is used in MNAOMIA [4], an analogy is a relation of resemblance between two entities D and D' , one from domain D , the other from domain D' . The resemblance between D and D' supposes the existence of an element of description in common d . Most of the time this resemblance is not found in the initial descriptions of D and D' , but can be obtained by the transformation of their descriptions, i.e. abstraction. Thus the common property, d , can be decomposed into two initial properties $d(D)$ and $d(D')$.

Such is the situation in this problem domain, we are matching between the description of a patient history and the trends so the paradigm applies here. We need a matching predicate between the description of a case C and the description of a trend T .

Let

$$C = \bigwedge_{i_C} El_{i_C}$$

and

$$T = \bigwedge_{i_T} El_{i_T}$$

be the descriptions of the trend T and the case C .

An element in the trend description El_{i_T} and an element in the case description El_{i_C} match if all their components match. Three main types of element (and their generalizations) have been defined for a case, and so a matching predicate must be defined for each of these :

$$match(C, T) \iff \forall_{i_T}, \exists_{i_C} [match(El_{i_C}, El_{i_T})]$$

Also, a non-matching predicate has to be defined for each of these :

$$nomatch(C, T) \iff \exists_{i_T}, \forall_{i_C} [nomatch(El_{i_C}, El_{i_T})]$$

where C is an instance of T (linked to it in memory by an *instance - of* relation). Depending of the type of the representation element, the *match* predicate takes the following forms :

1. *match - indep*(El_{i_C}, El_{i_T}) matches two time-independent elements :

$$\begin{aligned} match - indep(El_{i_C}, El_{i_T}) &\iff \\ \forall_{i_T}, \exists_{i_C} [(Att_{i_C} = Att_{i_T}) \wedge (Val_{i_C} = Val_{i_T})] & \\ nomatch - indep(El_{i_C}, El_{i_T}) &\iff \\ \exists_{i_T}, \exists_{i_C} [(Att_{i_C} = Att_{i_T}) \wedge (Val_{i_C} \neq Val_{i_T})] & \end{aligned}$$

2. *match - point*(El_{i_C}, El_{i_T}) matches two time-point elements :

$$\begin{aligned} match - point(El_{i_C}, El_{i_T}) &\iff \\ \forall_{i_T}, \exists_{i_C} [(t_{i_C} \geq t_{i_T} - timerange(Att_{i_T})) & \\ \wedge (t_{i_C} \leq t_{i_T} + timerange(Att_{i_T}))] & \\ \wedge (Att_{i_C} = Att_{i_T}) \wedge (Val_{i_C} \in & \\ [(Val_{i_T} - range(Att_{i_T}), (Val_{i_T} + range(Att_{i_T}))]) & \\ nomatch - point(El_{i_C}, El_{i_T}) &\iff \\ \exists_{i_T}, \exists_{i_C} [(t_{i_C} \geq t_{i_T} - timerange(Att_{i_T})) & \\ \wedge (t_{i_C} \leq t_{i_T} + timerange(Att_{i_T}))] & \\ \wedge (Att_{i_C} = Att_{i_T}) \wedge (Val_{i_C} \notin & \\ [(Val_{i_T} - range(Att_{i_T}), (Val_{i_T} + range(Att_{i_T}))]) & \end{aligned}$$

In addition to the *timerange* function previously defined, each attribute *Att* has a (min,max) range for matching. This depends on the attribute, and is returned by the function *range*(*Att*). Then the attribute range is (min=*val* - *range*(*Att*), max=*val* + *range*(*Att*)).

3. *match - interval*(El_{i_C}, El_{i_T}) matches two time-interval elements :

$$\begin{aligned} match - interval(El_{i_C}, El_{i_T}) &\iff \\ \forall_{i_T}, \exists_{i_C} [(t_{i_C,s}, t_{i_C,f}) during (t_{i_T,s}, t_{i_T,f}) & \\ \wedge (Att_{i_C} = Att_{i_T}) & \\ \wedge F_{i_C}(t_{i_C,s}, t_{i_C,f}) \in & \\ [(F_{i_T}(t_{i_T,s}, t_{i_T,f}) - range(Att_{i_T}), & \\ (F_{i_T}(t_{i_T,s}, t_{i_T,f}) + range(Att_{i_T}))]) & \\ nomatch - interval(El_{i_C}, El_{i_T}) &\iff \\ \exists_{i_T}, \exists_{i_C} [(t_{i_C,s}, t_{i_C,f}) during (t_{i_T,s}, t_{i_T,f}) & \\ \wedge (Att_{i_C} = Att_{i_T}) & \\ \wedge F_{i_C}(t_{i_C,s}, t_{i_C,f}) \notin & \\ \left[\frac{(F_{i_C}(t_{i_C,s}, t_{i_C,f}) - range(Att_{i_T}), & \\ (F_{i_C}(t_{i_C,s}, t_{i_C,f}) + range(Att_{i_T}))}{n} \right] & \end{aligned}$$

3.4 Hierarchical Organisation of Trends

Trends are linked to other trends by the relations *more - specific - than* (and its reverse relation *more - general - than*).

This relation permits the building of a hierarchy

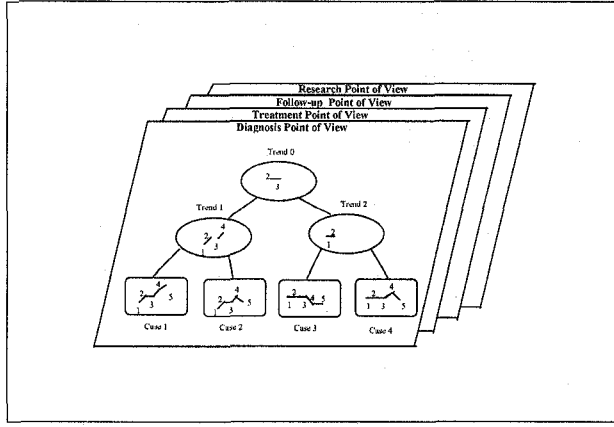


Figure 3: The hierarchical organization of the memory around several points of view.

of trends. This hierarchical organization of the trends is shown on Figure 3.

Let $T_G =$

$$\begin{aligned} &\wedge_{iG} < Att_{iG}, Val_{iG} > \\ &\wedge_{jG} < t_{jG}, Att_{jG}, Val_{jG} > \\ &\wedge_{kG} < t_{kG}, Att_{kG}, \sum_{pG=1}^{nG} \frac{Val_{pG}}{nG} > \\ &\wedge_{lG} < t_{lG,s}, t_{lG,f}, Att_{lG}, Fl_{lG}(t_{lG,s}, t_{lG,f}) > \\ &\wedge_{mG} < t_{mG,s}, t_{mG,f}, lb_{mG}, ub_{mG}, Att_{mG}, \\ &Gm_{mG}(t_{mG,s}, t_{mG,f}) > \end{aligned}$$

The match between a trend T_S and the most-specific trend T_G that generalizes it is realized if :

$$\begin{aligned} match - trend(T_S, T_G) &\iff \\ \forall_{iG}, \exists_{iS} [match(El_{iS}, El_{iG})] \end{aligned}$$

$$\begin{aligned} match - trend(T_S, T_G) &\iff T_S \text{ more-specific-than } T_G \\ &\iff \\ \forall_{iG}, \exists_{iS} [(Att_{iS} = Att_{iG}) \wedge (Val_{iS} = Val_{iG})] \\ &\wedge \\ \forall_{jG}, \exists_{jS}, [(t_{jS} \geq (t_{jG} - timerange(Att_{jG}))) \\ &\wedge (t_{jS} \leq (t_{jG} + timerange(Att_{jG}))) \\ &\wedge (Att_{jS} = Att_{jG}) \wedge (Val_{jS} \in \\ &[Val_{jG} - range(Att_{jG}), (Val_{jG} + range(Att_{jG}))]) \\ &\wedge \\ \forall_{kG}, \exists_{kS}, [(t_{kS} \geq (t_{kG} - timerange(Att_{kG}))) \\ &\wedge (t_{kS} \leq (t_{kG} + timerange(Att_{kG}))) \\ &\wedge (Att_{kS} = Att_{kG}) \wedge (Val_{kS} \in \\ &[(\sum_{pG=1}^{nG} \frac{Val_{pG}}{nG} - range(Att_{kG})), \\ &(\sum_{pG=1}^{nG} \frac{Val_{pG}}{nG} + range(Att_{kS}))]) \\ &\wedge \\ \forall_{lG}, \exists_{lS} [(t_{lG,s}, t_{lG,f}) \text{ during } (t_{lS,s}, t_{lS,f})] \\ &\wedge (Att_{lG} = Att_{lS}) \\ &\wedge Fl_{lS}(t_{lS,s}, t_{lS,f}) \in \\ &[(Fl_{lG}(t_{lG,s}, t_{lG,f}) - range(Att_{lG}), \\ &(Fl_{lG}(t_{lG,s}, t_{lG,f}) + range(Att_{lG}))]) \\ &\wedge \end{aligned}$$

$$\begin{aligned} &\forall_{mG}, \exists_{mS} [(t_{mG,s}, t_{mG,f}) \text{ during } (t_{mS,s}, t_{mS,f})] \\ &\wedge (Att_{mG} = Att_{mS}) \\ &\wedge Gm_{mS}(t_{mS,s}, t_{mS,f}) \in \\ &[(Gm_{mG}(t_{mG,s}, t_{mG,f}) - range(Att_{mG}), \\ &(Gm_{mG}(t_{mG,s}, t_{mG,f}) + range(Att_{mG}))]. \end{aligned}$$

More precisely, the memory is organized around points of view. Each point of view is specific of a precise cognitive task performed (i.e. diagnosis, treatment planning, etc.). These points of view are situated in a different part of the memory than the entities (trends, cases) previously defined : the theoretical memory (see Figure 4), but a different hierarchy of trends is associated to each of them.

4 Trends Learning

MNAOMIA performs an incremental concept learning during its case-based reasoning [5]. The search thru the memory, from a new case, starts from the root of the hierarchy, which is the most general trend of the memory. Most of the time, this root is empty, and the set of the most general trends in the memory are searched. For each of these, the match between the new case, and the trend, is evaluated, so for each description element of the trend. A successful match is returned if at least one interval of the case matches the concept description. The new case will be stored at any node it matches. Therefore the match predicate generalizes any trend elements that match at the time of the match. The pruning of trends and attributes will eventually remove any redundant concepts.

The pruning is done by Pos and Neg attributes which are associated with each description element in a trend. If a description element's negative score falls below a certain level it will be removed. If a trend's number of description elements falls below a certain level it is removed. These attributes are updated while the match is evaluated. When a match is successful for the whole trend, the algorithm passes the list of description elements of the case not contained in the trend to the sub-trends of this concept. The sub-trends are the trends linked to the current trend by *is-more-specific-than* relations.

Search(Trend, Case)

result := {}

for all representation elements in Case

if match(Trend.Element, Case.Element)

then POS := POS + 1

else

NEG := NEG + 1

if NEG - POS < Threshold

then

remove the element

from the Trend description

if Trend is empty

then

suppress the Trend from the Memory

link its sub-trends to its super-trends

in the hierarchy (fusion)

if there is at least one contradictory

description element between Trend and

Case

then return Nil

```

else for all Trend sub-trends
    newresult :=
        Search(sub-trend,
            unmatched elements in Case)
    if newresult <> Nil
    then result := result + newresult
    if result <> {}
    then return result
    else return {Trend}

```

This algorithm is a modified version of the GBM algorithm [14]. The main difference is in the *match* function. Matching a Case against a Trend is not only comparing values by an equality function, but a more complex matching process as the one defined previously. As a matter of fact, the Search algorithm could be adapted to other incremental concept learning algorithms than GBM, such as COBWEB [6,7] for example.

New trends are learnt when a new, revised case is added to the memory, at a given *Trend*. The algorithm is as follows :

```

Insert(Trend, Case)
UnMatched :=
    Elements of the Case not explained by
    the Trend's Elements
for all cases Ct indexed under the Trend
    nb := number of successful matches of
        match(Unmatched, Ct.Element)
    if nb > threshold
    then
        create a new sub-trend Tn of Trend with
            these successful matches
        index Case under Tn
        index Ct under Tn
        remove Ct from the cases indexed under
            Trend
if Case has not yet been indexed under a
    new sub-trend of Trend
then index Case under Trend

```

Here again, this algorithm is that of GBM [14], but the *match* function, and the generalization of a Trend, are much more complex than in GBM, due to the temporal elements in the cases descriptions.

5 Overview of the Reasoning Process

This system is a case-based reasoning system. Case-based reasoning is both an artificial intelligence methodology, and a cognitive model of reasoning such that reasoning is inseparable from memory and from learning. This cognitive model is a very high aim, which current case-based reasoning systems are just beginning to pursue. The tight interaction between reasoning, memory and learning is one characteristic of the general description of the system, as presented on figure 4, which has been inspired by Aamodt [2] and Bichindaritz [4]. The main steps in the reasoning process are to abstract, to retrieve, to reuse, to revise and to retain. They also correspond to the ST-MODEL [18] of medical reasoning. According to this model medical reasoning consists of abstracting the

important features from a patient case, selecting hypotheses from this abstraction, deriving expectations from these hypotheses, comparing the expected consequences of the first ranked hypothesis to the patient's actual condition, and confirming in memory the right hypotheses. So this case-based reasoning general framework is well adapted to medical reasoning.

5.1 Abstract

The initial patient case is a conjunction of attribute-value pairs attached to time. This representation is abstracted into an abstract case representation, in which the relevant features are calculated, in this problem time intervals bound to overall attribute values. A new case has an initial representation such as :

$$InitialCase = \bigwedge_i \langle Time_i, Att_i, Val_i \rangle$$

For example :

```

< ..., Surname, Catherine >
< 21/06/95, Diagnosis, Anorexia >
< 21/06/95, Weight, 38.9kg >
< 22/06/95, Weight, 38.5kg >
Case1 = < 24/06/95, Weight, 39.0kg >
< ... , ... , ... >
< 21/06/95, Calories, 1532kcal >
< 28/06/95, Calories, 1621kcal >
< ... , ... , ... >

```

Time_i can take several forms, and the user of the system is able to chose the granularity of the description, such as week, month, or like here day. From

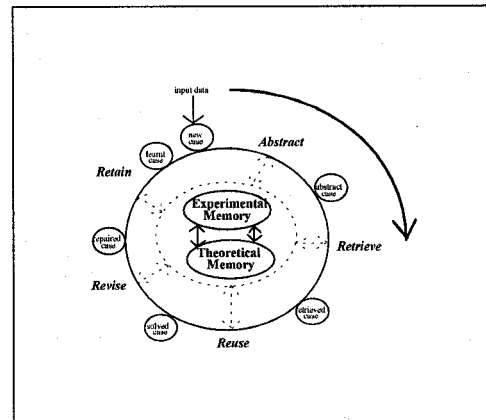


Figure 4: Architecture of the system.

these data, an abstract case representation is calculated. The advantage of an abstract representation is that it summarizes the main facts in the patient evolution, thus getting away too specific dates and values. The abstraction process calculates intervals, and attaches values to these intervals. The user also has the

option of defining the intervals in terms of days or in terms of weeks. If the evolution is to be defined in terms of weeks, the weekly velocity of the parameter, for example weight gain, is calculated and entered as the parameter value for that week.

The algorithm FindIntervals abstracts intervals from this representation, for all attributes having an evolution.

$$AbstractedCase = \bigwedge_i \langle Time_{i,s}, Time_{i,f}, Att_i, Val_i \rangle$$

For example :

$$AbstractedCase1 = \begin{aligned} &\langle \dots, Surname, Catherine \rangle \\ &\langle 8, \dots, Diagnosis, Anorexia \rangle \\ &\langle 8, 9, Weight, -0.4kg \rangle \\ &\langle 9, 11, Weight, +0.5kg \rangle \\ &\langle \dots, \dots, \dots, \dots \rangle \\ &\langle 8, 15, Calories, +89kcal \rangle \\ &\langle \dots, \dots, \dots, \dots \rangle \end{aligned}$$

The FindIntervals algorithm calculates the start $T_{i,s}$ and the finish $T_{i,f}$ of intervals, for each attribute Att , given an evolution of this attribute ranked in increasing time order :

```
i := 1
Ti,s = the first time point in the evolution
for each time-interval started
  take the current time-point
    in the evolution
  calculate the velocity between the
    time-point and the next one
  calculate the velocity with the next
    time-point added
  if the extended velocity is significantly
    different from the initial velocity
  then end the current time interval with
    Ti,f = the current time-point
    Vi = initial velocity
    i := i + 1
    start a new time-interval with
      Ti,s = the next time-point
  take the next time-point in the evolution
```

5.2 Retrieve

The abstracted case is then matched with the trends in memory, in order to retrieve the most similar memorized cases. First of all, this matching process is directed by the task to perform. This comes from the organization of the memory around points of view from the theoretical memory. Each task performed is associated to a unique point of view. So the retrieval step must be performed for a specific cognitive task, such as diagnosis or treatment. The trends are organized in hierarchies dependent upon the points of view in memory, as is represented in Figure 3. The case is first matched with the most general trends, and following with more and more specific trends.

The Search thru the memory is performed by the

Search(Trend, Case) algorithm described in the previous section. The result returned by this algorithm is a set of Trends.

$$Result = \{Trend_k\}.$$

Under these Trends, cases are indexed by *instance - of* relations. These cases are ranked by a similarity measure, such as

$$sim(Case_1, Case_2) = \frac{\sum_{k=1}^n \alpha \times sim(E_{1,k}, E_{2,k}) + \sum_{k=1}^n n_{k_i, pred} \times n_{i, pert} \times sim(E_{1,k}, E_{2,k})}{\alpha \times n + \sum_{k=1}^n n_{k_i, pred} \times n_{i, pert}}$$

where α is a weight associated to the representation elements which are not important for the current process, and where $n_{i, pert}$ and $n_{k_i, pred}$ are some weighting variables learnt by the system during previous cases processing. These variables are associated to the representation elements of cases and trends.

In the *Retain* part of the reasoning cycle, the learnt case will be stored at every trend that it matched, here the $Trend_k$. This is why a pruning is performed here, with the POS and NEG variables, associated to each trend representation element. Some parts of trends are removed, others are generalized, and finally, new trends are added, while previous trends can be deleted. So the matching, generalizing and pruning are performed simultaneously.

The *Result* set is the set of all most specific matching trends in the memory, for the new abstracted case. The set of all the cases indexed under these trends is the set *Cand* of all the cases similar to this new case. It is ranked by decreasing order of similarity by the *sim* function.

$$Cand = \{Case_k\}.$$

5.3 Reuse

This step is very dependent on the task performed. In problem-solving, it is an adaptation of the most similar case solution to fit the new case. In other tasks, such as interpretation, it can be the construction of an argumentation linking the new case with the retrieved cases.

5.4 Revise

Depending on the success or the failure of the new case processing, the reused case is revised and transformed into an updated case, now ready to be stored in the memory.

5.5 Retain

The algorithm performing the enrichment of the memory with the new revised case has been given in the previous section: *Insert(Trend, Case)*.

6 Related Work

This work is related both to temporal reasoning in general, and to time processing in case-based reasoning.

In case-based reasoning, very few systems have tackled time processing. This is due to the novelty of the domain: more crucial problems had to be solved first. For instance, MNAOMIA has addressed

many important problems before dealing with time, although it was obvious from the start that time processing could not be avoided. The other case-based reasoning systems that handle time use a continuous representation of their cases [15, 16]. They do not learn trends from these temporal data, and focus on matching between cases. The weakness of this approach is first of all its limitation to small amounts of cases and data, and the lack of all the advantages of qualitative reasoning, such as explainability, understandability and knowledge discovery.

On the other hand, thorough work has been performed in temporal reasoning. Since time processing has been driven by the application domain, only subsets of this work has been used in this system, such as the during operator [1,11], or the matching of known trends with new evolutions [8,9,17]. More theoretical work, such as the event calculus [12,13] would have been suited for the domain, but it exceeded the requirements of the application at its current state.

7 Conclusion

The MNAOMIA case-based reasoning system reasons from the temporal data in the patients cases in close interaction with its memory. MNAOMIA organizes its memory around hierarchies of trends learnt by incremental concept learning from the cases. This approach is more related to the temporal reasoning strategy of Allen [1] than to the event calculus of Kowalski and Sergot [13] because it updates its memory at input data processing time with the incremental concept learning previously presented. So interesting work can be accomplished by integrating more of Allen's formalisms in MNAOMIA. Moreover, the memory organization supports the subsequent reasoning of the system. This reasoning, and the complex problems related to the reuse step and to the revise step, are our current research topic. For this work, common formalisms with the event calculus [13] and those of temporal deductive databases seem promising.

References

- [1] Allen J.F., "Maintaining Knowledge about temporal intervals", *Communications of the ACM*, 26(11), 1983, 832-843.
- [2] Aamodt A., "Towards Expert Systems that Learn from Experience", In : *Proceedings of a Workshop on case-based reasoning (DARPA)*, Pensacola Beach, Florida, K.J. Hammond (Edt.), Morgan Kaufmann, San Mateo, CA, 1989, 181-187.
- [3] Bichindaritz I., "A case-based assistant for clinical psychiatry expertise", In : *Proceedings 18th Symposium on Computer Applications in Medical Care*, AMIA, Washington DC, 1994, 673-677.
- [4] Bichindaritz I., *Apprentissage de concepts dans une mémoire dynamique : raisonnement à partir de cas adaptable à la tâche cognitive*, Thesis of University René Descartes, Paris, 1994.
- [5] Bichindaritz I., "A case-based reasoning and conceptual clustering : for a cooperative approach", In : *Proceedings of the 1st UK Workshop on Case-Based Reasoning*, I. Watson and M. Fahrir (Edts.), Springer-Verlag LNCS/LNAI, 1995, (in press).
- [6] Fisher D.H., "Knowledge acquisition via incremental conceptual clustering" In : *Machine Learning 2*, 1987, 138-172.
- [7] Gennari J.H., Langley P. and Fisher D. "Models of Incremental Concept Formation", *Artificial Intelligence*, 40, 1989, 11-61.
- [8] Haimowitz I.J. and Kohane I.S., "An epistemology for clinically significant trends", In : *Proceedings of the Tenth National Conference on Artificial Intelligence*, 1994, 178-181.
- [9] Haimowitz I.J. and Kohane I.S., "Automated trend detection with alternate temporal hypotheses", In2: *International Joint Conference on Artificial Intelligence*, 1993, 146-151.
- [10] Kolodner Janet L., *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [11] Kouramajian V., Fowler J., "Modelling past Current and Future Time in medical Databases", In : *Proceedings of the 18th Symposium on Computer Applications in Medical Care*. 1994, 315-319.
- [12] Kowalski R.A., "Database updates in the event calculus", *Journal of Logic Programming*, 12, 1992, 121-146.
- [13] Kowalski R.A., Sergot M.J., "A logic-based calculus of events", *New Generation Computing*, 4(1), 1986, 67-95.
- [14] Lebowitz M., "Concept learning in a rich input domain", In : R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), *Machine Learning: an A.I. approach*, 2, Morgan Kaufmann, Los Altos, Ca., 1986.
- [15] Ram A. and Santamaría J.C., "Continuous Case-Based Reasoning", In : *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning*, 1993, 86-93.
- [16] Rougerez-Loriette S., "Prédiction de processus à partir de comportements observés : le système REBECAS", LAFORIA Doctoral Report (unpublished), 1994.
- [17] Shahar Y., Das A.K., Musen M.A., Kraemar F.B., "Knowledge-Based Temporal Abstraction for Diabetic Monitoring", In : *Proceedings of the 18th Symposium on Computer Applications in Medical Care*, 1994, 697-701.
- [18] Stefanelli M., Ramoni M. "Epistemological Constraints on Medical Knowledge-Based Systems", In : D.A. Evans and V.L. Patel (Eds.), *Advanced Models of Cognition for Medical Training and Practice*, Springer-Verlag, 1991, 3-20.