Scenario recognition for temporal reasoning in medical domains

Michel Dojat, Nicolas Ramaux, Dominique Fontaine

Artificial Intelligence in Medicine 14 (1998) 139–155

Scenario recognition for temporal reasoning in medical domains

Michel Dojat [a,*], Nicolas Ramaux [b], Dominique Fontaine [b]

[a] *Institut National de la Santé et de la Recherche Médicale, U438-RMN Bioclinique, Centre Hospitalier Universitaire-Pavillon B, BP 217, 38043 Grenoble, Cedex 9, France*
[b] *Université de Technologie de Compiègne, HeuDiaSyC UMR CNRS 6599, BP 20529, 60205 Compiègne, Cedex, France*

Abstract

The recognition of high level clinical scenes is fundamental in patient monitoring. In this paper, we propose a technique for recognizing a session, i.e. the clinical process evolution, by comparison against a predetermined set of scenarios, i.e. the possible behaviors for this process. We use temporal constraint networks to represent both scenario and session. Specific operations on networks are then applied to perform the recognition task. An index of temporal proximity is introduced to quantify the degree of matching between two temporal networks in order to select the best scenario fitting a session. We explore the application of our technique, implemented in the *Déjà Vu* system, to the recognition of typical medical scenarios with both precise and imprecise temporal information. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Temporal reasoning; Temporal constraint network; Patient monitoring; Scenario recognition; Intensive care monitoring

1. Introduction

Interpreting data over time is an essential task to diagnose and control processes. The work presented in this paper deals with the same recurrent question: how to

* Corresponding author. Tel.: +33 4 76765748; fax: +33 4 76 765896; e-mail: dojat@ujf-grenoble.fr

PII S0933-3657(98)00020-7

recognize dynamically from discrete medical data, high level clinical scenes as they are developing? To answer this question, we propose *scenario recognition* as a technique for reasoning about time in dynamic systems when no mathematical model of the process is known which can determine its behavior completely. We apply it to temporal reasoning in medical domains: the time-course of a clinical process is compared to a predetermined set of possible behaviors for this process. These predetermined behaviors are named *scenarios*. The recognition of the scenario S (or a part of S) states that the observed time-course of the process, called a *session* (Σ), corresponds to S. This recognition allows us to anticipate forthcoming events from the partial instantiation of the recognized scenario, and to intervene on the process in order for instance to keep it out of specific expected (undesirable) situations.

1.1. Two example problems

We introduce two typical medical situations which are used as running examples for the rest of the paper to illustrate the two main aspects of our work: (1) the scenario recognition *on the fly* while the time-stamped session is developing; and (2) the scenario recognition *a posteriori* with temporal imprecision attached to the occurrence date of some events.

1.1.1. Scenario recognition on the fly

A simple scenario $S0_1$ excerpt from the management of mechanical ventilation may have the following sequence (Fig. 1):

Part 1: a progressive increase in the respiratory rate appears after a long period of stable ventilation. The last suctioning of the endotracheal tube is reported 2 h prior to the respiratory rate variations. Afterwards no specific events are reported.

Part 2: A few minutes later, an alarm (high respiratory rate, i.e. Tachypnea) is raised on the ventilator. The patient is then disconnected from the ventilator for suctioning and reconnected to the ventilator. His/her ventilation is erratic for a few minutes.

Part 1 suggests that bad ventilation is due to the partial obstruction of the endotracheal tube whereas Part 2 suggests that it is due to the stress generated by the disconnection/reconnection process.

A computerized system capable of gradually recognizing the scenario $S0_1$, may automatically increase the mechanical assistance before suctioning, adapt the assistance to prevent short stress after reconnection and finally replace an adequate

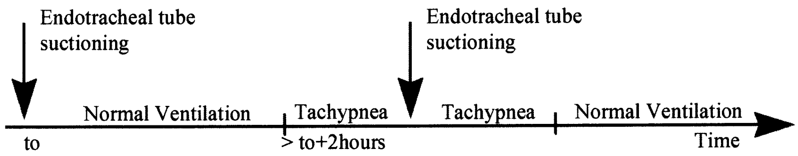


Fig. 1. A medical scenario ($S0_1$), excerpt from mechanical ventilation management, for recognition on the fly.

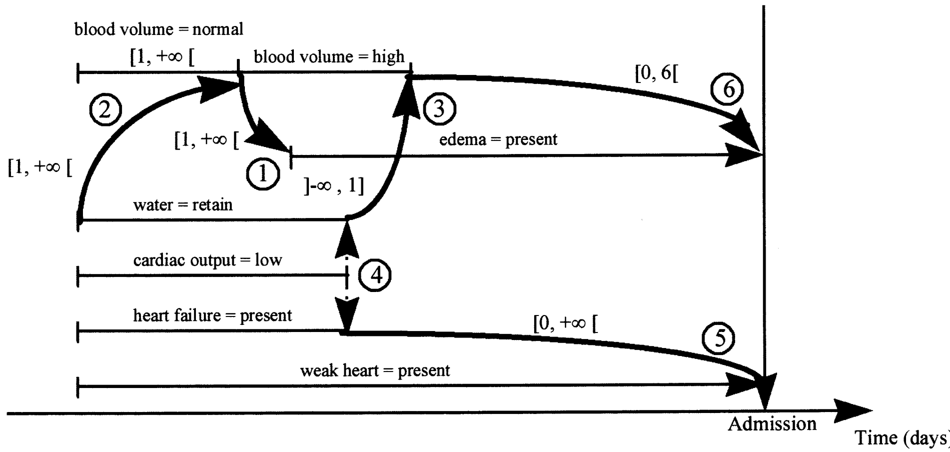


Fig. 2. A medical scenario ($S0_2$), excerpt from cardiovascular disorders management, for recognition *a posteriori*. Adapted from [12].

value of assistance when the episode $S0_1$ comes to a end. In this example, all information incorporated into the session is time-stamped and the recognition must be performed on the fly while the session is developing.

1.1.2. Scenario recognition a posteriori

A second scenario $S0_2$ is built from examples of the management of patients with complex cardiovascular disorders as presented in [12] (Fig. 2). The problem here is to recognize the apparition of an edema due to the existence of a heart failure for hours. To cause edema, the high blood volume must persist for at least one hour, the water retention must have started at least one hour prior to the beginning of the high blood volume and must not have ended more than one hour prior to the end of high blood volume, the low cardiac output and the heart failure must be simultaneous with the water retention, and the heart must be weak at least during the presence of heart failure. Finally, the maximum relaxation time for edema is 6 h. In Fig. 2, all of these temporal constraints are represented respectively with the thick arrows labeled 1, 2, 3, 4, 5 and 6[1].

A computerized system capable of recognizing the scenario $S0_2$, may conclude that edema is due to the presence of high blood volume. Such a system must deal with imprecise temporal information.

The goal of this paper is to propose a technique, implemented in a prototype system called *Déjà Vu*, for the automatic recognition from a session of scenarios such as $S0_1$ or $S0_2$. Firstly, in Section 2, we review the different approaches

[1] The example is extremely simplified compared to a real medical situation and should not be taken to represent medical reality. We use a time scale of hours but it may take from hours to days to mobilize edema. The relationships represented are however characteristic of time relations in medicine and are sufficient to illustrate the use of our technique for the management of imprecise temporal relations.

proposed in the literature to interpret time-varying data. In Section 3, we present the basic definitions for scenario and session and introduce the notion of temporal constraint networks for representing both scenario and session. In Section 4, we detail the mechanisms for comparing scenarios and sessions.

Section 5 is devoted to the high level language used by the expert to construct the scenarios base. The *Déjà Vu* system is described in Section 6. Section 7 demonstrates the application of the recognition of medical scenarios from time-stamped sessions or from sessions including imprecise temporal information. In Section 8, we discuss several aspects of our work for intelligent patient monitoring and conclude in Section 9, pointing out work under development.

2. Previous work on the interpretation of time-varying medical data

In medical domains, several approaches have been proposed to allow computerized interpretation of physiological parameters over time. The different steps for transforming raw data (quantitative or numeric information) into high level temporal abstractions (qualitative or symbolic information) are overviewed in Fig. 3. In fact, in the schematic, the systems cited need all the steps shown. However, the original methods they are based on use one specific step. Note that similar steps to those for data interpretation shown in Fig. 3 are present in systems for image interpretation: mathematical morphology techniques, the use of deformable and statistical models for segmentation, and spatial abstraction mechanisms, respectively replace time-series analysis, trends detection and classification, and temporal abstraction mechanisms.

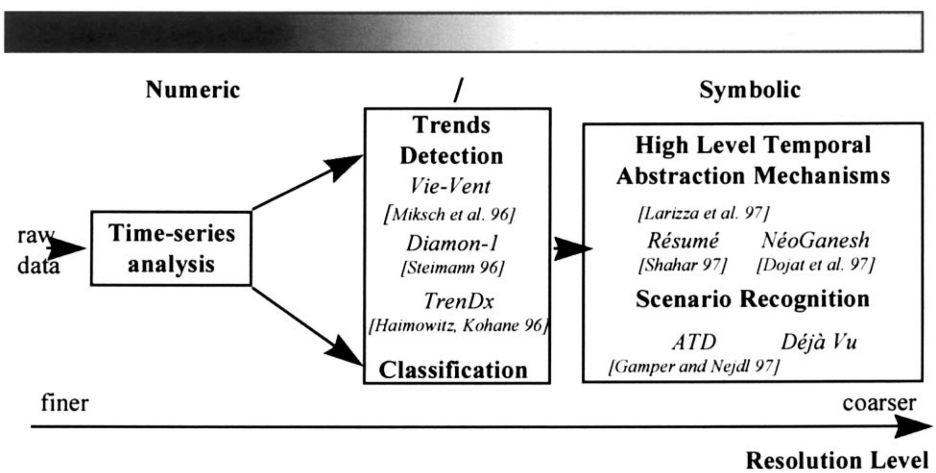


Fig. 3. Methods for interpretation of temporal data. Each rectangle represents a class of methods for temporal data analysis. With this progressive transformation (signal-to-symbol conversion), the resolution level (time granularity) increases and methods vary from purely numeric techniques to symbolic techniques.

Methods of formal time-series analysis classically provide useful tools for interpreting continuous raw data. They are based on statistical techniques such as smoothing and nonparametric approaches, and rely on the knowledge (or the presumption) of an analytic model that fits the raw data and whose parameters must be estimated. The choice of the analytic model requires a basic understanding of the processes that generate the time-series data. Clearly, this is not always the case, particularly in medical domains. To overcome the difficulty in using time-series analysis in a real-world application, the monitoring of artificial ventilation of premature newborn infants, Miksch et al. [15] introduce in the Vie-Vent system sophisticated methods for qualitative trend descriptions of dynamic respiratory data. Similarly, because curve-fitting models of pediatric growth were non adaptable to clinical needs, Haimowitz and Kohane have developed the $TrenD_x$ system [7] which detects data consistent with a collection of domain-specific predetermined patterns called trend templates. In the same vein, in DiaMon-1 Steimann introduces for the interpretation of time-varying data two methods based on fuzzy set theory, one for the detection of trends and one for the tracking of disease histories [20]. With the Résumé system [19], Shahar proposes a knowledge-based framework for automatically deriving, from raw data, high level temporal abstractions. NéoGanesh, a closed-loop system for artificial ventilation control, uses two specific knowledge-based abstraction mechanisms, *aggregation* and *forgetting*, to dynamically interpret ventilation parameters [3]. Similarly in [10], Larizza et al. demonstrate that temporal abstraction mechanisms are not only useful for summarizing a patient's behavior over a time period, but may also be exploited for defining an abstract description of the patient's states for the interpretation of the patient's evolution.

All of these approaches are built with specific goals: to efficiently match data against patterns for DiaMon-1 and $TrenD_x$, to interpret data and to define therapeutic actions in real-time for Vie-Vent and NéoGanesh, and to generate temporal abstractions for Résumé and [10]. Clinical scenario recognition, as proposed in this paper, is complementary to the approaches described above: the session we want to recognize may be constructed from them.

3. Scenario and session: basic concepts

Our approach is based on two entities: the Session and the Scenario.

3.1. *Definitions*

In our context, Session and Scenario have the following definitions:

Session (Σ): This temporal structure represents the real evolution of the observed process in a working situation. The session is perceived via sensors: raw data are filtered, processed and finally classified into discrete values that determine the states of the process. The states succession defines the dynamics of the process behavior. Temporal abstraction mechanisms may be used to construct a concise view of the time-course of the process (Section 2).

Scenario (S): A scenario models an expected evolution of the process. Scenarios are defined by a domain expert using a high level language (Section 5.2). We do not assume that all the possible situations for a process are modeled and stored in the scenario base. According to the type of applications envisaged, two different scenario interpretations may be considered:

● a scenario models *a class of behaviors*. This definition may be related to the notion of *Script* [18] that describes a stereotyped sequence of events in a particular context; or to the notion of *Chronicle*, a primitive in the temporal logic of McDermott [14] that represents a complete possible history of the process. Scenario recognition is then equivalent to a classification problem: how to match a class (a scenario) with a given pattern (a session)?

● a scenario models *a real*, *previously encountered situation*. A scenario is then considered as an unique case stored in memory (a case base). As proposed in case-based reasoning (CBR), recognition of a scenario consists of retrieving the previous case from memory and adapting it to fit the session. In CBR, the temporal dimension of scenarios and sessions is generally missing.

3.2. Scenarios and sessions as temporal networks

We consider scenarios as temporal entities modeling a class of behaviors. Temporal representation is then central in our approach. We have identified three main point-based approaches to represent scenarios and sessions associated with corresponding temporal reasoning methods. Lévy [11] proposes to represent scenarios and sessions with linear sequences of time-stamped events. Scenario recognition consists of finding a pattern (the complete scenario) in a chain of characters (an ordered collection of events representing the session). With this model, it is not possible to introduce imprecision attached to the temporal occurrence of events. For instance, we can not express the fact that two events, $\mathtt{e}_1$ and $\mathtt{e}_2$, should appear in any order but before a third one, $\mathtt{e}_3$. Scenarios may be represented using temporal constraint networks whose vertices represent time-stamped events and whose edges represent temporal constraints. The more relaxed the temporal constraints, the more general the scenario. These constraints may be numeric or symbolic leading to constraint propagation algorithms in $O(n^3)$ or $O(n^2)$, respectively, where n is the number of vertices in the network, when conjunctive temporal constraints are considered The management of disjunctive constraints is NP-hard. Based on temporal constraint propagation, Dousson et al. [8] model a scenario as a network which contains a set of events $\mathtt{e}_\mathtt{i}$ and windows of relevance $\mathtt{w}(\mathtt{e}_\mathtt{i})$, indicating all the possible occurrence dates of possibly forthcoming events $\mathtt{e}_\mathtt{i}$. The session is represented as a linear sequence of events.

Figs. 4 and 5 respectively show the temporal network that represents the scenario $S0_1$ (Fig. 1) and the temporal network that represents the scenario $S0_2$ (Fig. 2). $\mathtt{e}_\mathtt{i}$ stands for occurring events and the edges are labeled with numerical temporal constraints. In Fig. 4, ($\mathtt{e}_1$, $\mathtt{e}_2$) bounds the disconnection of the patient

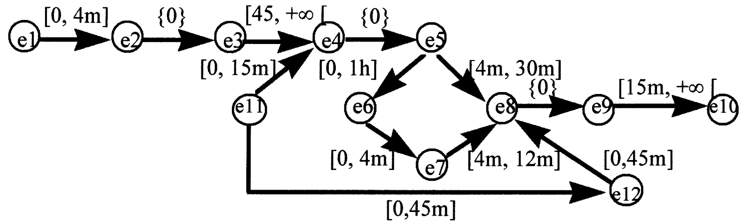


Fig. 4. The temporal network for SO_1, where m and h stand for minutes and hours, respectively.

during suctioning. (e_3, e_4) bounds the normal ventilation period. e_5 initiates and e_8 terminates the tachypnea period (high respiratory rate). (e_6, e_7) bounds the patient's disconnection. e_9 initiates and e_{10} terminates the normal ventilation period. (e_{11}, e_{12}) bounds the progressive increasing of respiratory rate.

In Fig. 5 (e_1, e_2) and (e_3, e_4) bound the normal and the high blood volume periods, respectively. (e_7, e_6) bounds the weak heart period and (e_9 ,e_8) the heart failure. (e_{11}, e_{10}) bounds the low cardiac output and (e_{13}, e_{12}) bounds the water retention. (e_{14}, e_{15}) bounds the edema presence and e_5 stands for the admission time.

Following [5], we consider both scenario and session as *temporal constraint networks* represented by $Gs = (V_s, E_s)$ and $G_\Sigma = (V_\Sigma, E_\Sigma)$ where V and E are the sets of vertices and edges of each network, respectively. This choice allows us to manage imprecision attached to the ordering and occurrence of the events that compose the session. The mechanism of recognition consists of matching two temporal networks: one network for the session and one network for the scenario. Then, we dispose of a battery of standard algorithms developed in network theory, for performing operations on session and scenario such as the matching task. For instance we use the well-known path consistency algorithm proposed in [13] to verify the coherence of the networks and to solve the recognition task.

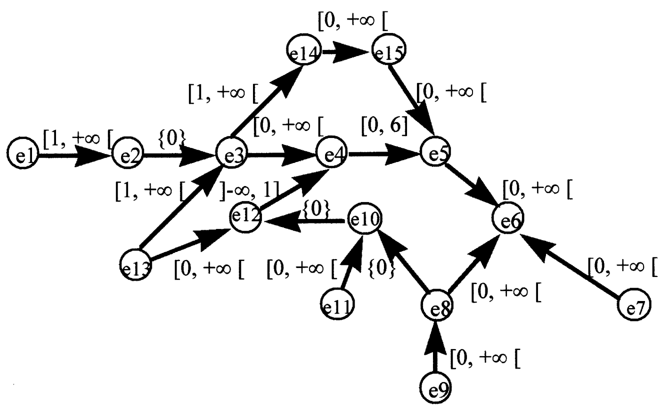


Fig. 5. The temporal network for SO_2

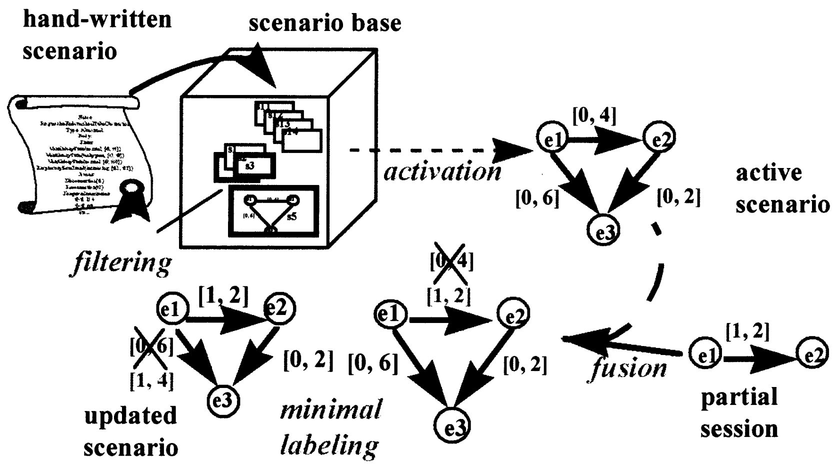


Fig. 6. The different steps of the scenario recognition process.

4. Recognition task

4.1. The comparison mechanism

The comparison mechanism is illustrated in Fig. 6. The hand-written scenario is transformed into a minimal point algebra network and stored into our scenario base (Section 5.1). Only a part of the scenario base is relevant to the session description depending on the context (filtering). When its preconditions are valid, the corresponding filtered scenario is activated. This *active scenario* is then retrieved to be matched against the *partial session*. The operations of fusion (between session and scenario) and minimal labeling[2] are performed to construct the updated scenario. In the example shown in Fig. 6, the first operation of fusion between the partial session and the scenario modifies C_{12} initially set to [0,4] to [1,2]. Then the minimal labeling operation changes the constraint C_{13} initially set to [0,6] to [1,4]. The updated scenario is the central element of the mechanism of recognition. In recognition on the fly, the constraint network of the session is evolving, as the result of the progressive construction of the session: each time an event predicted by a scenario occurs in the session, its occurrence date is taken into account and the temporal constraints are updated.

4.2. Two levels of comparison

To discriminate the scenarios matching a session, we introduce two levels of comparison: a symbolic level and a numeric level.

[2] This operation consists of finding the minimal constraint network minG which has the same solutions as the network G.

4.2.1. Symbolic level

This level is based on three relations: *satisfaction*, *compatibility* and *incompatibility* ([5] for their formal definition). To determine the relation corresponding to a pair < session/scenario >, we use standard operations on temporal networks. Compatibility (resp. Incompatibility) is established if the minimal network $\mathtt{F}$, obtained from the *fusion*[3] of $\mathtt{G}_{\mathtt{S}}$ and $\mathtt{G}_{\Sigma}$ leads to a *coherent*[4] (resp. *incoherent*) network. Moreover, if $\mathtt{F}$ is included into $\mathtt{G}_{\Sigma}$, satisfaction is demonstrated [5]. As a result, at the end of the process described in Fig. 6, we declare the following between a session and a scenario:

- incompatibiliy if the updated scenario has no solution
- compatibility if the updated scenario has at least one solution
- satisfaction if the network of the updated scenario is included into the network of the session.

If some events present in the scenario are not referenced in the session, weintroduce the notion of *partial compatibility* or *partial satisfaction*.

4.2.2. Numeric level

The criteria of compatibility or incompatibility are not always sufficient. If a session satisfies a scenario, the recognition comes to an end and we consider that the system behaves as described by the scenario. If the session is only compatible with a scenario, it is not certain whether this scenario references the behavior of the system. Moreover, if two scenarios are compatible with the session, we must decide which scenario gives the best description of the session. It may be also interesting to determine the least incompatible scenario with the session. In order to solve this problem we introduce a temporal proximity index to measure the distance between the session network and a scenario network, and discriminate or classify the scenarios with respect to their ability to reference the same session. Intuitively, a *temporal proximity index* between a scenario and a session must estimate a temporal overlapping between their networks. When a scenario and a session are incompatible, the temporal networks being in some way disjunct, this index must estimate a temporal difference between them. The temporal proximity index is differentiated into two indexes: a *temporal compatibility index* (`Cind`) or a *temporal incompatibility index* (`Iind`). We detail the calculation of these indexes in [5].

5. High level representation of scenarios

5.1. Temporal ontology

Our ontology divides the world into two types of entities: (1) *atemporal entities* (`ATObject`) used to model the observed system; and (2) *temporal entities* (`Tobject`) used to model its evolution. `ATObject` is similar to the notion of

[3] The fusion operation of `G1` and `G2` consists of computing the intersection of two overnetworks of `G1` and `G2`.

[4] A minimal network is coherent if and only if it has at least one solution.

Feature in [17]. It is a predefined label for abstraction. `Descriptor` and `Trend`, subclasses of `ATObject`, are qualitative information obtained from quantitative values of a set of parameters representative of the system. `ATObject` are obtained by knowledge-based abstraction mechanisms: `Descriptor` is the result of the point-temporal abstraction mechanism [19]; `Trend` qualifies the rate of variation of parameters. For scenario recognition, we consider `Trend` as a specific descriptor of the system. For instance, the patient's ventilation is characterized by physiological parameters including `RespiratoryRate (numeric value)`, from which we infer the value of the descriptor `RespiratoryState (normal, tachypnea, …)` and the value of the trend `RespiratoryRateTrend (decreasing, increasing, …)`.

`TObject` is similar to the notion of *Fluent* [17] or to the notion of *Time-object* introduced in [8] with a set of temporal, causal and structural relations. `TObject` is composed of two basic classes: instantaneous entities (`PointBasedTObject`) and entities valid on a period of time (`IntervalBasedTObject`). We define a model of change in which events happen at time-points and initiate and/or terminate properties. Our definition of `Event` is similar to the Event Calculus definition [9]: an event occurrence initiates a change in our description of the world (`Descriptor=Value, {initiates or terminates}, occurrenceDate`). The occurrence date of an event is represented by an interval that models the imprecision attached to its value. A distribution law can be associated to this interval and used during the numeric matching. `Measurement (Descriptor=Value, occurrenceDate)` states that the descriptor has the value (`Value`) when the measurement is performed (`occurrenceDate`). `Measurement` is used when the sample frequency is insufficient, compared to the dynamics of a parameter, for assuming the default persistency of the parameter value between two measurements. However, unlike `Event`, a `Measurement` occurrence does not necessarily introduce some change in the modeled world, i.e. a new value for a descriptor. This concept is useful, for example for integrating episodic blood gas measures. `Property`, subclass of `IntervalBasedTObject`, represents a descriptor whose value (`Value`) holds during the validity interval `Ival` associated with it (`Descriptor=Value, Ival`). For instance, an assertion such as (`RespiratoryRateTrend= increasing, [t1, t2]`) means that the respiratory rate increases during the period between instants `t1` and `t2`.

The design of scenarios by the expert and the automatic construction of the session by the *Déjà Vu* system is facilitated by the introduction of the notion of *Transition*. `Transition`, subclass of `PointBasedTObject`, represents the simultaneous occurrence of two events `e1, e2: e1` which terminates a property `p1` for the descriptor `d` and `e2` which initiates a new property `p2` for `d`. The use of `Transition` for the construction of the session is illustrated in Section 6.

5.2. Scenario description

Scenarios are represented by a temporal constraint network. Temporal constraints C_{ij} between two events e_i and e_j are expressed with equalities or inequalities between their respective occurrence dates t_i and t_j: $V^n_{i=1}$, $C_{ij} = t_j - t_i \in v^n_{i-1}[b_i, b_{i+1}]\ b_i, b_{i+1} \in \Re \cup \{+\infty, -\infty\}$.[5]

Each scenario contains a conclusion part triggered when the scenario is recognized or rejected. The triggering functions to introduce a new set of events for futher reasoning. A scenario is made up of four parts: *name*, *type* (abnormal, ...), *body* (Properties, Events and Temporal constraints), and *conclusion*.

The conclusion part is triggered if temporal constraints between events and properties, which appear in the scenario body, are verified in the session.

In the scenario in Fig. 7, which represents the complete scenario $S0_1$, the conclusion states that the tube is obstructed and the patient's secretions are abundant. `Properties` are used to represent patient's respiratory states and respiratory rate trends. `Events` are used to indicate the patient's disconnection from the ventilator.

Before its incorporation into the scenario base, each scenario, expressed in the specified ontology, is transformed into a point algebra network. Each `Tobject` that appears in the scenario description is transformed into a `TNetworkNode`: `Event` and `Measurement` are the two basic temporal objects that constitute the nodes of the temporal constraint networks representing sessions and scenarios. For each property, two events are incorporated automatically corresponding to its initiation and its termination. To keep the relation embedded into a property, the transformation operation introduces a symbolic link called a *constraint of persistency* that forces two events to initiate and terminate the same property. The temporal constraint propagation algorithms do not take into account this type of constraints. Their satisfaction is tested during the first step of the session and scenario matching (Section 6).

Name :
ProgressiveEndotrachealTubeObstructionFollowedByUnstabilitiesAfterSuctioning
Type: Abnormal
Body:

Properties

(RespiratoryState=normal, [t3, t4])
(RespiratoryState=tachypnea, [t5, t8])
(RespiratoryState=normal, [t9, t10])
(RespiratoryRateTrend=increasing, [t11, t12])

Events

(DisconnectionPatient=true, initiates, t1)
(DisconnectionPatient=true, terminates, t2)
(DisconnectionPatient=true, initiates, t6)
(DisconnectionPatient=true, terminates, t7)

Temporal constraints

t2-t1 ∈ [0, 4], t3-t2 ∈ [0, 0], t4-t3 ∈ [120, +∞[, t5-t4 ∈ [0, 0], t11-t4 ∈ [0, 15], t6-t5 ∈ [0, 60], t8-t5 ∈ [4, 30]

t7-t6 ∈ [0, 4], t8-t7 ∈ [4, 12], t12-t8 ∈ [0, 45], t9-t8 ∈ [0, 0], t10-t9 ∈ [15, +∞[, t12-t11 ∈ [0, 45]

Conclusion:
(EndotrachealTubeState = Obstructed, [t5, t6])
Secretion.Value = high

Fig. 7. The complete scenario $S0_1$.

[5] For disjunctive constraints, the symbolic comparison is NP hard. In our experience with medical scenarios representation, disjunctive constraints were never required.

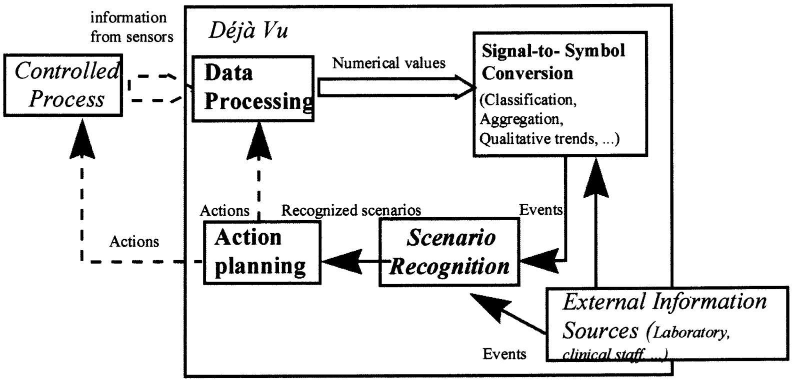


Fig. 8. The global architecture of the *Déjà Vu* system. Each rectangle represents a specific function of the *Déjà Vu* system. The dashed arrows show the information flux in case of a closed-loop system.

6. The *Déjà Vu* system

Fig. 8 shows the architecture of the *Déjà Vu* system for patient monitoring. In real-time applications, *Déjà Vu* takes raw data from the controlled process by means of several sensors. Numeric data are then processed (Data Processing module) and transformed to symbolic values describing the process (Signal-to-Symbol Conversion module). These symbols are used to assess high level concepts describing the time-course of the process (Section 2). The scenario recognition module is fed with information obtained from external sources (laboratory, nurses, ...) and with events generated by the Signal-to-Symbol Conversion module (Section 5.1). The output of the scenario recognition module is a set of partially or totally recognized scenarios, ordered according to the calculated temporal proximity index. The partial recognition of a scenario generates a set of expected events which are taken into account by the action planning module to determine the current action to perform. The recognition of a scenario is stopped when the action planning module decides that this scenario is no longer relevant (change of context or incompatibility) or when the scenario is totally recognized.

The session is incrementally built from the incoming events. Each time a proposition change is detected by the temporal abstraction module, one transition is generated. For instance, when at time t_5, `RespiratoryState` changes from `Normal` to `Tachypnea`, the transition (`RespiraytoryState, (Normal, Tachypnea),` t_5) is introduced into the session. The date t_5 may be imprecise and included into a temporal interval ($t5 \in [bs, be]$). The transition is transformed into two events, `e1=(RespiraytoryState=Normal, terminates,` t_5`)`, and `e2=(RespiraytoryState=Tachypnea, initiates,` t_5`)` added to the corresponding temporal constraint network.

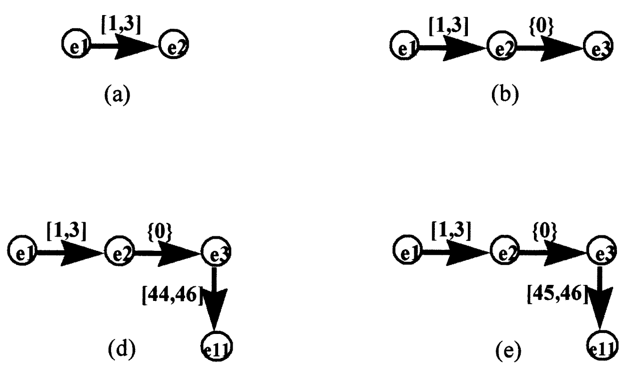
The matching task between a session and a scenario is decomposed into two phases: (1) the determination of a subset of events from the session that verify the symbolic links and must be linked to the nodes of a scenario; and (2) the computation of the operations of fusion and minimal labeling. The current version of the *Déjà Vu* system is implemented using the Smalltalk Language (VisualWorks 2.5, ParcPlace US). Currently, we are exploring the use of a constraint solver, such as BackTalk [16] written in Smalltalk, to realize the minimalisation. This requires the transformation of a temporal constraint network into an equivalent network with finite domain non-temporal constraints.

7. Medical scenario recognition: a case study

We illustrate the use of the *Déjà Vu* scenario recognition module with two examples: (1) the gradual on-line recognition of part of scenario $\mathtt{SO}_1$ (Figs. 1 and 4); and (2) the recognition *a posteriori* of scenario $\mathtt{SO}_2$ (Figs. 2 and 5).

7.1. Recognition on the fly

Time origin is the occurrence date of the first disconnection, $\mathtt{e}_1$. An agenda stores the expected events. The representation of updated scenario networks is restricted to the session nodes. At time $\mathtt{t=0}$, the events in the agenda are $\mathtt{e}_2$ and $\mathtt{e}_3$, both expected between `0` and `4 min`. This agenda is updated only when an expected event occurs or when the system clock reaches a time equal to the smallest upper bound of the set of expected time intervals. Hence, at time $\mathtt{t=0}$, the *Déjà Vu* scenario recognition module concludes that if no events occur this agenda should not be updated before `4 min`. At `t=2 min`, the *Déjà Vu* scenario recognition module is informed that $\mathtt{e}_2$ occurred between `1` and `3 min` after $\mathtt{e}_1$. The session and the updated scenario networks are shown in Fig. 9 (a). The updated agenda consists of the single expected event $\mathtt{e}_3$ with a time interval between `0` and `1 min`. At `t=3 min`, the *Déjà Vu* scenario recognition module is informed that $\mathtt{e}_3$ occurred simultaneously with $\mathtt{e}_2$. The session network is shown in Fig. 9 (b). After detecting $\mathtt{e}_3$, events `e4, e5 and e11` appear in the agenda (Fig. 9 (c)). The expected time



Event	Intervalle
e_4	[45, +∞[
e_5	[45, +∞[
e_{11}	[45, +∞[

(c)

Event	Intervalle
e_4	[0, 11m]
e_5	[0,11m]
e_{12}	[0,41m]

(f)

Fig. 9. The recognition process applied to the ventilation management example.

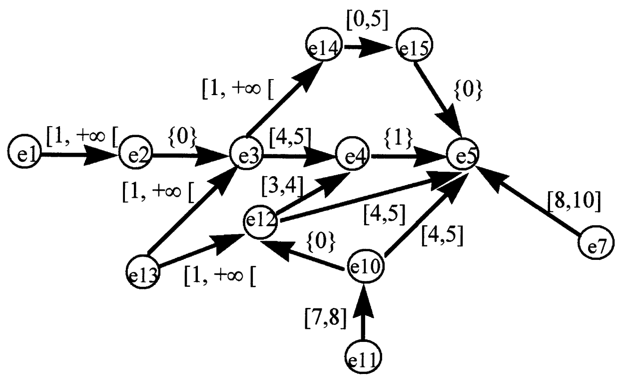


Fig. 10. The fusion network of the scenario and session corresponding to the cardiovascular disorder example.

interval for e_{11} has been calculated previously during the minimal labeling and is equal to: $[45\ m, +\infty]$.50 min after the occurrence of e_3, the *Déjà Vu* scenario recognition module is informed that e_{11} occurred between 44 and 46 min after e_3. The updated networks for the session and the scenario are shown in Fig. 9 (d, e). The constraint linking e_3 and e_{11} indicates that the coherent updated scenario is not included into the session. Then, we have a partial compatibility between session and scenario and calculate the corresponding compatibility index (equal in this case to 1/2). After the occurrence of e_{11}, the new agenda is shown in Fig. 9 (f). Each time a new event appears in the session, the agenda is updated with the same technique detailed above.

For each partially recognized scenario, the outputs of the *Déjà Vu* scenario recognition module is the agenda which contains the expected events and the corresponding compatibility index. In the example, we have recognized the normal ventilation (e_3) following the reconnection of the patient (e_2) and the beginning of the increase in the respiratory rate (e_{11}). The *Déjà Vu* scenario recognition module predicts that the increase in the respiratory rate will last no longer than 41 min (e12). The ventilation will remain normal for 11 min (e_4) and will be followed by a tachypnea (e_5).

7.2. Recognition a posteriori

The *Déjà Vu* scenario recognition module starts with the following information about the time course of the patient's disease. The patient was admitted to hospital with high blood volume for 5–6 h ,finishing 1 h prior. His/her cardiac output was low during a period of 7–8 h and returned to normal 4–5 h prior to admission. His/her heart rate is weak for 8–10 h. The fusion operation between this session and the scenario (Fig. 5) leads to the temporal network showed on Fig. 10. Several constraints are modified compared to the initial scenario network. There is no empty constraint in the network after the minimalisation operation. Thus, we conclude that the session satisfies the scenario. Due to some events present in the scenario not being referenced in the session, we conclude a partial satisfaction. From this network, we can infer that the water retention started (e_{13}) at least 5 h prior to admission (e_5) and stopped (e_{12}) between 4 and 5 h prior to admission. Edema, bounded by (e_{14}, e_{15}), is suspected to be present for less than 5 h.

8. Discussion

The notion of clinical scenarios (or clinical scenes) to reflect physiological processes was previously introduced in [1]. Our main innovation is to consider both scenario and session as temporal constraint networks.

The approach proposed by Gamper and Nejdl [6] with the ATD system applied to the diagnosis of hepatitis B is close to our work. In ATD specific domain knowledge and causal relationships among states and evoked manifestations present in the candidate scenario, are used to force stronger temporal conditions for specific temporal relations into the session network. If the network obtained is temporally consistent then the candidate scenario models the current session. The use of temporal networks for representing session and scenario is strongly related to our work with a few but fundamental differences. ATD is devoted to the off-line scenario recognition for abduction-based diagnosis. It is mainly concerned with qualitative scenarios and then uses an interval-based algebra. No mechanism is introduced for discriminating between two scenarios fitting the session. The approach can be used in medical applications where qualitative temporal relations among findings and causal relationships among observations and underlying processes can be established.

In process monitoring, the recognition must be performed on the fly and generally all the pertinent events are time-stamped. The session can then be easily modeled as an ordered sequence of events. This corresponds to the approach proposed in [4] where only scenarios are modeled as temporal networks. Our choice of representing both session and scenario as temporal networks is motivated by the type of applications we consider. In medical domain some events are not automatically recorded and their occurrence date is only known with imprecision and relatively to other events. For instance, the supervisor may be informed by the clinical staff that the pulse rate dropped between half an hour to one hour before the starting of the severe tachypnea. The importance of managing imprecise temporal relations is reinforced in the a posteriori recognition of medical scenarios. Clearly, the drawback of our approach is its computational burden. This is especially critical for on-line recognition. Nevertheless, we remain optimistic due to three factors:

(1) In practice, constraints of persistency limit the number of active scenarios. The use of a temporal window that indicates the shortest and the longest delay to obtain information about descriptors, allows us to eliminate active scenarios which contain expected events that, as time goes on, will never respect these temporal constraints.

(2) To speed up the recognition process, the scenario base can be organized according to specific relations between scenarios such as causality, inheritance or similarity.

(3) We are mainly interested in intelligent patient monitoring applications such as ventilation management, where real-time constraints are not too hard (approximate response time between 1 s and 1 min).

Many questions remain to be answered, and more work is needed in order to answer them. We only consider the temporal nature of the distance between a session and a scenario and common events are supposed to be identical. However, some events may be slightly different. In this case, we need the introduction of the notion of *atemporal proximity* and links between temporal and atemporal proximity indexes.

The construction of the scenario base is a crucial point. We elaborate, in collaboration with clinicians, typical clinical scenarios from data obtained with a computerized system used in ICU for the automatic control of mechanical ventilation [2,3]. However, the difficulty in building scenarios is very much a limitation of our approach. Learning techniques could be incorporated into our system with advantage to automatically generate new scenarios from sessions.

9. Conclusion

The recognition of typical scenarios is an essential task for patient monitoring systems. In representing scenarios and session as temporal networks, we have:

- a high level of *temporal expressiveness*. Temporal constraint networks offer a natural way to express imprecision linked to the date of occurrence of events. Relative temporal ordering of events can be introduced both in session and scenario.
- *uniform representation* for sessions and scenarios. This allows us to introduce a temporal proximity index to order the recognized scenarios. Moreover, as proposed in CBR, scenarios could be automatically learnt directly from recorded sessions.

In spite of the intrinsic computational complexity of the technique we propose for scenario and session comparison, we believe that our scenario recognition module can be integrated successfully into a ventilation supervisor.

We are currently implementing such a module and evaluating its real-time performance at the Henri Mondor Hospital for the supervision of patients hospitalized in ICU. Systems providing automated support for guideline-based clinical care need to recognize user intentions and plans to achieve them. Our future work includes applying temporal scenario recognition to this task.

Acknowledgements

Nicolas Ramaux has a grant from the French Ministry of Research and Technology. Dr Laurent Brochard was our expert in the domain of mechanical ventilation and contributed much of his valuable time.

References

[1] Cohn AI, Rosenbaum S, Factor M, Miller PL. DYNASCENE: An approach to computer-based intelligent cardiovascular monitoring using sequential clinical 'scenes'. Meth Inform Med 1990;29:122–31.

[2] Dojat M, Harf A, Touchard D, Laforest M, Lemaire F, Brochard L. Evaluation of a knowledge-based system providing ventilatory management and decision for extubation. Am J Respir Crit Care Med 1996;153:997–1004.

[3] Dojat M, Pachet F, Guessoum Z, Touchard D, Harf A, Brochard L. NéoGanesh: A Working System for the Automated Control of Assisted ventilation in ICUs. Artif Intell Med 1997;11:97–117.

[4] Dousson C, Gaborit P, Ghallab M. Situation recognition: representation and algorithms, 13th IJCAI, Chambéry, France, 1993:166–172.

[5] Fontaine D, Ramaux N. An approach by graphs for the recognition of temporal scenarios. IEEE Trans Syst Man Cybern, June 1998.

[6] Gamper J, Nejdl W. Abstract temporal diagnosis in medical domains. Artif Intell Med 1997;10:209–34.

[7] Haimowitz IJ, Kohane IS. Automated trend detection with alternate temporal hypothesis. 13th IJCAI, Chambéry, France, 1993:146–151.

[8] Keravnou ET. Temporal diagnosis reasoning based on time-objects. Artif Intell Med 1996;8:235–65.

[9] Kowalski RA, Sergot MJ. A logic-based calculus of events. New Gener Comput 1986;4:67–95.

[10] Larizza C, Bellazzi R, Riva A. Temporal abstractions for diabetic patients management. In: Keravnou E, Garbay C, Baud R, Wyatt J, editors. Artif Intell Med. Berlin: Springer, 1997:319–330.

[11] Lévy F. Recognizing scenarios: a study. Workshop on diagnosis from first principles (DX-94), New Paltz, 1994.

[12] Long WJ. Reasoning about state from causation and time in a medical domain. AAAI, Washington, 1983:251–254.

[13] Mackworth AK. Consistency in networks of relations. Artif Intell 1977;8:99–118.

[14] McDermott D. A temporal logic for reasoning about processes and plans. Cognit Sci 1982;6:101–55.

[15] Miksch S, Horn W, Popow C, Paky F. Utilizing temporal data abstraction for data validation and therapy planning for artificially ventilated newborn infants. Artif Intell Med 1997;8:543–76.

[16] Roy P, Pachet F. Reifying constraint satisfaction in Smalltalk. J Obj Orient Prog 1996;10:43–51.

[17] Sandewall E. Features and Fluents. The representation of knowledge about dynamic systems. Oxford: Oxford University Press, 1994.

[18] Schank RC, Abelson RP. Scripts, plans, goals and understanding. Hillsdale, NJ: Erlbaum, 1977.

[19] Shahar Y. A framework for knowledge-based temporal abstraction. Artif Intell 1997;90:79–133.

[20] Steimann F. The interpretation of time-varying data with DiaMon-1. Artif Intell Med 1996;8:343–57.