

Managing Time Granularity of Narrative Clinical Information: The Temporal Data model TIME-NESIS

Carlo Combi ([°]), Francesco Pinciroli (^{°*}) and Giuseppe Pozzi ([^])

[°] Dipartimento di Bioingegneria del Politecnico di Milano

^{*} Centro di Teoria dei Sistemi del CNR, Milano

[^] Dipartimento di Elettronica e Informazione, Politecnico di Milano

Abstract

In the database field, the need of time management at different levels of granularity has been on for some years. It has been recently emphasized. Such a need is widespread when we have to manage medical clinical information. The temporal dimension is normally given at different granularities. In database systems based on the calendar time, granularity refers to uncertainty in specifying a temporal dimension as well as to use of different time-units. Thus, we need to set up modelling concepts and managing tools by which to establish temporal relationships between temporal clinical data. To manage the temporal dimension of data given at various and mixed levels of granularity, we defined the temporal data model TIME-NESIS (TIME in anamNESIS). The model provides temporal dimension to the data by using intervals that can be specified by different granularities. The model supports a three-valued logic, where True, False and Undefined are the truth values.

1. Introduction

The problem of the representation and management of temporal information drew the attention of many field of computer science: among them we mention here artificial intelligence, database systems, real-time systems [1].

A need has recently been emphasized: the **time management at different levels of granularity** [1, 2, 3]. The granularity of a given temporal information is the level of abstraction at which information is defined [4]. In respect to the definition of granularity we identify three main types of granularity:

- *abstraction granularity*: this type of granularity is not directly related to the time axis. Abstraction granularity refers to the ability of expressing complex and composite temporal concepts, e.g. "A parameter value increased during a period of three months" [5, 6].
- *absolute-time granularity*: it is the ability to express the temporal dimension of information by mixing and using different absolute time references, e.g. "the interval began in a period from 15:23 to 16:34".

Absolute-time granularity refers to uncertainty in specifying a temporal dimension as well as to use of different time-units [2, 3, 7, 8].

- *calendar-date granularity*: it is the capacity of expressing the temporal dimension by using many time units, e.g. years, months, days [9, 10].

Natural language sentences refer to the calendar time both considering instants - "at 4:45 p.m., October 15, 1990, the patient suffered from myocardial infarction" - and considering intervals - "the patient had abdominal pain from 5 p.m. to 7 p.m., March 27, 1989" or "at 4:30 p.m., October 26, 1991, the patient's renal colic ended, it lasted five days" or "the patient suffered from an episode of tachycardia lasting for 150 seconds on October, 26 1991, at 3:22 p.m.". As showed in the above expressions, time dimension is often given at different and mixed abstraction granularities; moreover, time intervals are given in many different ways - starting and ending instants, starting instant and duration, ending instant and duration, a combination of the preceding modalities -.

The need to store in an appropriate way the contents of any of the above sentences is present in the medical clinical field, e.g. during the collection of anamnestic data [2]. Temporal information, related to the events the patient narrates frequently, has a different accuracy depending on whether the information is related to remote events or to closer ones. Furthermore, this heterogeneity in expressing temporal clinical information does not weaken the need of establishing temporal relationships between the stored temporal information. This is the only way to suitably reconstruct and manage the whole clinical history of a patient.

We focused, without losing generality, on the modeling of different absolute-time granularities in clinical narrative data: for systems based on the calendar-(or absolute-) time, there is the need to capture in a more exhaustive way also temporal indeterminacy not exactly related to the calendar date (e.g. "in the afternoon") [3].

As regards these needs, this paper describes a temporal data model named TIME-NESIS (TIME in anamNESIS) to represent and manage the temporal dimension of clinical narrative data given at various and mixed levels of abstraction granularity. TIME-NESIS fits into the general

framework of data modeling for temporal databases. Using a widely accepted terminology, TIME-NESIS deals with the definition at different levels of granularity of the valid time, i.e. the time when the represented information is true in the modeled reality [11].

2. The temporal data model

2.1. The adopted three-valued logic

The wide choices available when specifying the values of an instant, a duration and a temporal interval, lead to suppose that a sure relationship between two instants may be not always defined. Assertions related to relationships between two instants (or durations or intervals) can neither be affirmable nor deniable for sure, but can inspire answers like "may be" or "you do not know". For example, we cannot state if the instant "November 30, 1991 in the afternoon" is after or before the instant "November 30, 1991". Three truth values are then possible: *True*, *False*, *Undefined*.

The proposed temporal data model semantics relies on a three-valued logic. The truth values (also called logical or Boolean) belong to the set $\{T, F, U\}$, where symbols T, F, U stand for "True", "False", "Undefined", respectively [12, 13]. We extended the basic logical connectives *not* (\neg), *and* (\wedge), and the quantifier *for each* (\forall) to manage the three truth values; we introduced another basic connective *istrue*, by the notation $T()$, to state if a formula assumes the logical value *True* or not.

The interpretation of formulas $\neg A$, $A \wedge B$, and $T(A)$ can be summarized by the following truth tables, according to the interpretations of formulas A and B:

\wedge	T	F	U
T	T	F	U
F	F	F	F
U	U	F	U

A	$\neg A$	$T(A)$
T	F	T
F	T	F
U	U	F

The logical connectives *or* (\vee), *implies* (\rightarrow), *co-implies* (\leftrightarrow), *isundefined* (U), *isfalse* (F), and the quantifier *exist* (\exists) are defined by the previously introduced basic logical operators.

2.2. The temporal ontology

The temporal data model uses a time scale based on elementary instants. Let *elementary instants* be the discrete points distinguishable on time scale.

Variables of *elementary instant* type, having the set **Elem** as domain, have properties similar to those of

variables assuming values belonging to the set of integers. In the following the discrete points distinguishable on the time axis are at the level of seconds; the model, however, can be enriched to capture finer time points.

The *calendar-date granularity* is the unit of measure adopted for the time scale [4]. Variables of *calendar-date granularity* type have as domain the set $G = \{SUP, Y, M, D, H, Mi, S, INF\}$, where SUP stands for granularity greater than year granularity, Y stands for year granularity, M for month granularity, and so on; INF stands for granularity smaller than seconds. For the set **G** of granularities the functions *max()* and *min()* and the predicates $>$, $<$, $=$, are defined, according to the ordering criterion, by which $SUP > Y > M > D > H > Mi > S > INF$. In our model we don't face actually the well-known problem of representing the week granularity [9, 14, 15].

Let *instant* be a generic temporal point; its position on the time axis is specified at an arbitrary absolute-time granularity. Let *duration* be the generic distance between two instants. A duration can be specified at an arbitrary absolute-time granularity.

Inst is the domain of variables of *instant* type; **D** is the domain of variables of *duration* type. For variables of *instant* type we defined the functions *Inf()* and *Sup()*, returning values belonging to the set **Elem** of the elementary instants: an instant is unequivocally defined by two elementary instants. For instance, the instant expressed as "October 23, 1991" is characterized by two elementary instants expressible as "October 23, 1991; 0 hours, 0 minutes, 0 seconds" (lower bound) and "October 23, 1991; 23 hours, 59 minutes, 59 seconds" (upper bound). These elementary instants are the lower bound and the upper bound of the segment, on the time axis, containing all the elementary instants the particular temporal position can coincide with.

Similarly we define the functions *inf()* and *sup()* for variables of *duration* type: a duration is unequivocally defined by the lowest and the greatest distance between elementary instants the particular duration can coincide with.

Let *interval* be a closed set of contiguous elementary instants. **Itvl** is the domain of the variables of *interval* type. For variables of *interval* type we defined the following functions: *Start()*, *End()*, returning values belonging to the set **Inst** of the instants; *Dur()*, returning a value belonging to the set **D** of the durations.

An interval is unequivocally characterized by a starting instant, an ending instant, and a duration. Only by all the three functions *Start()*, *End()*, and *Dur()* it is possible to suitably quantify intervals given at different and heterogeneous granularities; for example, by these functions it is possible to identify an interval having a duration given at a granularity finer than that of the starting and ending instants: e.g. "on September 1994, for 24 minutes". On the other hand, it is possible to identify some intervals having the starting (ending) instant given at a granularity finer than the ending (starting) instant: "on

September 23, 1994 for seven months" or "for seven months until September 23, 1994" or "from September 23, 1994 to December 12, 1994 in the evening".

Let *temporal assertion* be a proposition asserted as true for an interval. The domain for the variables of *temporal assertion* type is the set **E**. A temporal assertion can be expressed by natural language as "the patient suffered from an episode of tachycardia lasting for 150 seconds on October, 26 1991, at 3:22 p.m.". Other examples of temporal assertions are given in the section 3.

For variables of *temporal assertion* type we defined the functions *Itvl()*, returning a value belonging to the set **Itvl** of the intervals, and *Prop()*, returning a value belonging to the set **P** of the propositions. By the only *temporal assertion* type we characterize the whole temporal information; such assumption implies that the considered temporal information is somehow atomic in respect with the truth value of the considered proposition on the assigned interval. The proposed model does not consider decomposition of a temporal assertion into more temporal assertions at a lower abstraction level.

2.3. Temporal relationships and operations

Some binary predicates are defined, using the above defined functions for variables of *instant* type: *temporally expressed as* (\approx), *equal at a predefined granularity* ($=_{g=X}$), *before* ($<$), *after* ($>$).

The relationships between instants can be True, False, and Undefined. This feature does not, however, weaken the proposed temporal data model: even though it is possible to freely assign the value of an instant, the proposed model maintains all the properties of a similar temporal data model, in which the value of each temporal instant is expressed at a unique and a-priori defined granularity. It is also possible to define relationships between instants defined at different granularities. In some cases it may be not possible to establish the above defined relationships, according to the two classical truth values *true* or *false*. For example, the instants "on October 28, 1991" and "on October 1991" cannot be compared by the operator *before*. It is not possible to assign a logical value *true* or *false* as a result of this relationship. The operator *before* can also lead to three different values: T, F, U. Instants can be connected by the operator *equal at a predefined granularity*, so that it is possible to express relationships described in natural language by "are in the same year" and "are in the same month of the same year".

For variables of *duration* type the predicates *temporally expressed as* (\approx), *equal at a predefined granularity* ($=_{g=X}$), *minor* ($<$) are defined in the same way as for variables of *instant* type; in this case we refer to the variables of *elementary instant* type by the functions *inf()* and *sup()*.

The function *sum* (symbol +), having an instant and a duration as arguments, returns an instant; there are also

two functions *difference*, between two instants and between an instant and a duration. We finally defined the concept of *pseudo-distance* $d(a,b)$ between two variables a , b of *instant* type.

We extended the 13 temporal relationships defined by Allen in [5], to consider different absolute-time granularities (tab. I); we also defined some new temporal relationships, to consider *calendar-date granularity*-based properties and to allow comparisons between intervals also on the basis of their durations. Let \underline{a} , \underline{b} , be variables of *interval* type. The following predicates define relationships between intervals, by using the predicates and the functions previously introduced for variables of *duration* type and of *instant* type.

Relationship	Symbol	Symbol for inverse
\underline{a} temporally expressed as \underline{b}	\approx	\approx
\underline{a} contemporary to \underline{b} at granularity X	$=_{g=X}$	$=_{g=X}$
\underline{a} temporally specifies \underline{b}	tS	i tS
\underline{a} before \underline{b}	$<$	$>$
\underline{a} overlaps \underline{b}	O	iO
\underline{a} during \underline{b}	D	iD
\underline{a} starts as \underline{b} at granularity X	$S_{g=X}$	iS $_{g=X}$
\underline{a} finishes as \underline{b} at granularity X	$F_{g=X}$	iF $_{g=X}$
\underline{a} meets \underline{b} at granularity X	$M_{g=X}$	iM $_{g=X}$

Table I. Relationships between intervals based on relationships on starting and ending instants.

All the predicates defined for the variables of *interval* type are suitably extended to the variables of *temporal assertion* type: e.g., if E , F are variables of *temporal assertion* type and t is a variable of *interval* type, the predicate $E \approx t$ will be equivalent to $Itvl(E) \approx t$ and similarly the predicate $E \approx F$ will be equivalent to $Itvl(E) \approx Itvl(F)$.

2.4. Temporal constraints for the interval

After introducing temporal predicates, it is possible to specify temporal constraints for a variable of *interval* type. For a variable \underline{a} of *interval* type some relationships exist between the values returned by the three functions *Start()*, *End()*, and *Dur()*.

More precisely the following formula is always satisfied:

$$\begin{aligned} \forall \underline{a} \quad & (Start(\underline{a}) \approx End(\underline{a}) - Dur(\underline{a}) \vee \\ & End(\underline{a}) \approx Start(\underline{a}) + Dur(\underline{a}) \vee \\ & Dur(\underline{a}) \approx d(End(\underline{a}), Start(\underline{a}))) \end{aligned}$$

This constraint is related to the dependencies among the three entities characterizing an interval: given the ending and starting instants, the duration can be evaluated; given the starting instant and the duration, the ending instant can be evaluated; given the ending instant and the duration, the starting instant can be evaluated. In the modelization here presented, however, the granularities for starting and ending instants and for the duration of an interval can be different; in particular it is possible to specify a duration of an interval by a finer granularity even if the starting or ending instants of the interval are specified at a coarser granularity.

2.5. The set of temporal assertions

The set W of temporal assertions can be viewed, as regards the temporal dimension, as a network of temporal constraints between starting and ending elementary instants of intervals. These constraints fit into the class of Simple Temporal Problems (STP) [16]; in this case complete constraint propagation is tractable. There are algorithms able to compute the minimal network in $O(n^3)$ steps where n is the number of variables (starting and ending elementary instants) [16].

By relationships and properties defined for temporal assertions, it is possible to describe the characteristics of a set of temporal assertions. In defining predicates for temporal assertions, one or more assigned temporal assertions or one or more assigned intervals have been referred: now, we aim at using all that above defined properties, to highlight the qualities of a certain set W of temporal assertions. Let p, t be constants of *proposition* and of *interval* type, respectively, assuming an arbitrarily assigned value; let E be a variable of *temporal assertion* type. The basic formula, for a set W of temporal assertions, is:

$$\text{OCCURS}(p, t) \equiv_{\text{df}} \exists E (Prop(E) \rightarrow p \wedge Itvl(E) \approx t)$$

The formula $\text{OCCURS}(p, t)$ allows to establish if inside the set W a temporal assertion exists having the temporal part expressed exactly as t and the propositional part implying p . The condition on the temporal part allows to characterize completely the set W in respect with the temporal dimension of the temporal assertions belonging to the set. The implication predicate between propositions follows the classical two-valued logic. The interpretation of the formula OCCURS , then, can only assume the truth values T or F; we are able to separately manage the uncertainty coming from relationships between intervals and/or temporal assertions of the set W , by suitable formulas relying on the basic formula OCCURS .

This formula allows to rigorously define the typical properties of the set W of the temporal assertions. The existence of the temporal assertion E in W implies that in the same set W no other temporal assertion exists having the propositional part implying $Prop(E)$ or $\neg Prop(E)$ and being true in an interval not completely disjointed from

$Itvl(E)$. In order to avoid some inconsistency situations for the generic set W , we make the following restriction on the set W :

$$T(\forall r \forall y (OCCURS(r, y) \rightarrow \neg \exists F ((Prop(F) \rightarrow r \vee Prop(F) \rightarrow \neg r) \wedge \neg (Itvl(F) > y \vee Itvl(F) < y))))$$

2.6. Some formulas for temporal assertions

By using $\text{OCCURS}(p, t)$, we introduce some other formulas to establish the presence of the above defined temporal relationships between temporal assertions contained in W . For example, the formula $\text{BEGINS}(p, t, X)$ establishes if a temporal assertion, having the propositional part implying p , started the interval t at the granularity X : this formula establishes if p started contemporaneously to t at the granularity X and it finished during t . The formula $\text{BEGINS}(p, q, X)$ establishes the existence of two temporal assertions, the first starting at the granularity X as the second and finishing before the second finished, having the propositional parts implying p and q , respectively.

3. The clinical application of the temporal data model

To illustrate the power of the proposed temporal data model and to exemplify the differences between the TIME-NESIS model and a classical two-value logic, we consider an example from the clinical medicine. In this field, sentences related to symptoms, pathologies, measured parameters, and therapies are often used: the temporal dimension is expressed by different and heterogeneous modalities and at various granularities [2, 3].

Let us now consider a set of temporal assertions; the previously defined temporal formulas allow to suitably investigate relationships between temporal assertions with different granularities.

Let $\{G, H, I, L\}$ be the set W of temporal assertions, related in this case to a patient:

G : "In 1990 the patient took a calcium-antagonist for three months"

H : "At 4:45 p.m., October 15, 1990, the patient suffered from myocardial infarction"

I : "At 4:30 p.m., October 26, 1991, the patient's renal colic ended, it lasted five days"

L : "The patient suffered from an episode of tachycardia lasting for 150 seconds on October, 26 1991, at 3:22 p.m."

To exemplify how the model considers the temporal assertions, we use a formal notation adopting the following conventions:

- the tuple $\langle Prop(Z), Itvl(Z) \rangle$ represents the generic temporal assertion Z ;
- the tuple $\langle Start(y), Dur(y), End(y) \rangle$ represents the generic interval y ;
- the tuple $\langle Inf(a), Sup(a) \rangle$ represents the generic instant a ;

- the tuple $\langle \text{inf}(_a), \text{sup}(_a) \rangle$ represents the generic duration $_a$.
- Each elementary instant is described by the usual format Y/M/D/H/Mi/S.

The formal notation of the temporal assertions, above expressed in natural language, is:

G: $\langle \text{calcium-antagonist}, \langle \langle 90/1/1/0/0/0, 90/9/1/23/59/59 \rangle, \langle 0/3/0/0/0/0, 0/3/29/23/59/59 \rangle, \langle 90/4/1/0/0/0, 90/12/31/23/59/59 \rangle \rangle$

H: $\langle \text{myocardial_infarction}, \langle \langle 90/10/15/16/45/0, 90/10/15/16/45/59 \rangle, \langle 0/0/0/0/0/0, 0/0/0/0/0/0 \rangle, \langle 90/10/15/16/45/0, 90/10/15/16/45/59 \rangle \rangle$

I: $\langle \text{renal_colic}, \langle \langle 91/10/20/16/30/1, 91/10/21/16/30/59 \rangle, \langle 0/0/5/0/0/0, 0/0/5/23/59/59 \rangle, \langle 91/10/26/16/30/0, 91/10/26/16/30/59 \rangle \rangle$

L: $\langle \text{tachycardia}, \langle \langle 91/10/26/15/22/0, 91/10/26/15/22/59 \rangle, \langle 0/0/0/0/2/30, 0/0/0/0/2/30 \rangle, \langle 91/10/26/15/24/30, 91/10/26/15/25/29 \rangle \rangle$

By the previously introduced formulas some considerations can be drawn on the patient:

- **the tachycardia episode and renal colic finished in the same day, while renal colic started before the tachycardia episode:** the formula $\text{FINISHES}(\text{tachycardia}, \text{renal_colic}, D)$ returns T;
- **the tachycardia episode was not before the renal colic:** the formula $\text{BEFORE}(\text{tachycardia}, \text{renal_colic})$ returns F;
- **it may be that the infarction happened during the calcium-antagonist therapy:** the formula $\text{DURING}(\text{myocardial_infarction}, \text{calcium-antagonist})$ returns U;
- **it may be that the infarction happened after the calcium-antagonist therapy:** the formula $\text{AFTER}(\text{myocardial_infarction}, \text{calcium-antagonist})$ returns U; with the same meaning the formula $\text{U}(\text{AFTER}(\text{myocardial_infarction}, \text{calcium-antagonist}))$ returns T;

The design and the implementation of TIME-NESIS fit an object-oriented approach; in particular we designed and implemented each of the introduced variable types by a suitable abstract data type (ADT). By the object-oriented approach it is possible to hide the algorithmic details in implementing functions and predicates for the various abstract data types; e.g. we can adapt TIME-NESIS to many different calendric systems without modifying the ADT hierarchy. The types *elementary instant*, *instant* and *duration* can be hidden to the user of the temporal data model; the types *interval* and *temporal assertion* allow to completely model natural language sentences. By the inheritance property it is also possible to specialize the *temporal assertion* type to model more specific sentences. In the medical field, for example, we can suitably specify some specific temporal assertions like *diagnostic temporal assertion* or *therapy-related temporal assertion* or *symptom-related temporal assertion*. The ADT *temporal assertion set* models the set \mathbf{W} ; suitable functions allow to express the formulas previously introduced over \mathbf{W} . A suitable ADT can implement also the three-valued logic:

the abstract data type *bool3* manages all the extended logical operators. We designed and implemented the temporal data model by object-oriented techniques; we used a C++ based object-oriented DBMS, named ONTOS, in implementing the temporal data model [2]. We applied it to a clinical database prototype, based on ONTOS too, able to manage clinical data from follow up patients undergone to a coronary-artery angioplasty. These patients after a 2-3 day-long hospitalization go through some periodical follow-up visits aiming at verifying the efficacy of angioplasty. For these patients the narrative data have to be updated also during the follow-up period. Management of temporal aspect of these data is therefore relevant; the clinical history of these patients has to be managed to relate follow-up parameters with clinical events, such as pathologies or therapies [2].

4. Discussion and conclusion

Two features, in general, distinguish TIME-NESIS from the other proposals dealing with the time granularity in the database field [7, 10, 15, 17]: 1) the use of a three-valued logic in managing different time granularities, and 2) the capability to identify a time interval by different granularities both for the starting instant, the duration, and the ending instant.

The choice in TIME-NESIS about the *abstraction granularity* on the temporal notions is similar to what proposed in most contributions from temporal constraint based approach [8, 16] and from temporal databases [3, 18]: we defined a unique entity, the *temporal assertion*, to model a proposition having a temporal dimension. TIME-NESIS is not able to represent at the data level temporal information having different abstraction granularities, as in [2, 6, 17, 19]. However, TIME-NESIS, applied to the temporal data modeling at database level, could enable higher abstraction level systems - as those described in [6, 17] - to perform more powerful abstractions on the clinical data, also considering mixed granularities and uncertainty in the temporal references.

As regards the *absolute-time* and *calendar-date granularities*, our approach is similar to those in [3, 7] allowing to express the uncertainty in the time reference that is not only limited to the calendar-date granularity. TIME-NESIS, unlike the Gadia's proposal, is not able to represent the valid time as a finite union of intervals [13]. TIME-NESIS, unlike Sarda's and Wiederhold's proposals in [10, 18], does not limit to consider a same granularity level, for both the temporal dimensions, into a relation or, equivalently, into a set of temporal assertions: a set of temporal assertions can consider intervals at different and mixed granularities.

TIME-NESIS uses a hierarchical logical structure - elementary instants, instant, duration, interval - to give both sound computational properties and abstract representation to the temporal dimension of information. This approach is similar to that in [7]. TIME-NESIS is the only to support the capability to identify, at a higher

abstraction level, a time interval by different granularities both for the starting instant, the duration, and the ending instant. This feature also allows to completely use the theory proposed by Dechter and colleagues and by Console and colleagues in [8, 16] about temporal constraints, where, at a lower abstraction level, constraints between time points represent intervals at a different granularity.

In respect to the *IOUs* (Interval Of Uncertainty) and *IOC* (Interval Of Certainty) of Das and colleagues or to the *variable intervals* of Console and colleagues, our *instant* and *interval* types allow to semantically distinguish in a clearer manner the set of time points, a generic time point can coincide with, from the set of time points composing an interval [3, 8].

As regards the ability to express *relative* and *absolute temporal dimension*, TIME-NESIS lacks in representing sentences like "the angina began after the patient finished her daily exercise routine"; we can only represent sentences referring to the absolute time (calendric time). This approach is quite usual in the field of temporal databases and in medical decision support systems [3, 6, 18]; however, TIME-NESIS allows to define on intervals, instants and duration a very wide number of new temporal relationships, also considering absolute-time and calendar-date granularities.

Few works faced the problem of the *management of the uncertainty in temporal relationships* [7, 17]; these few contributions are coming from the field of the temporal reasoning and are based on the adoption of some modal operators. TIME-NESIS allows to manage at the level of database modeling some degrees of uncertainty in temporal relationships (between temporal assertions, intervals, instants, durations), by a three-valued logic. The approach of Pernici and colleagues in [7, 17] has some differences from what performed by the three-valued logic of TIME-NESIS: more particularly, in LATER a formula satisfying a MUST query satisfies also the corresponding MAY query. In TIME-NESIS if a *istrue* (T) formula returns the truth value *True*, the corresponding *isundefined* (U) formula returns the truth value *False*. In other words, TIME-NESIS restricts the meaning expressed by the sentence "it is possible" to consider only the formulas that are uncertain. The Gadia's approach in [13] is based on a three-valued logic too. Gadia deals with uncertainty related to the set-based relationships between *partial temporal elements*; he focused on the existence of a tuple (or of the values of their attributes) in a relation more than directly with the uncertainty coming from temporal relationships [13].

Acknowledgments

This work was partially supported by contributions from many sources: MPI 40% Italian National Project for Medical Informatics; MPI 40% Italian National Project for the Bio-Engineering of Cardiovascular System; Departments of Biomedical Engineering of the Politecnico di Milano.

References

- [1] Maiocchi R, Pernici B. Temporal Data Management Systems: A comparative View, IEEE Trans. on Knowledge and Data Eng. 1991; 3(4): 504 - 524.
- [2] Combi C, Pincioli F, Pozzi G. Temporal Clinical Data Modeling and Implementation for PTCA Patients in an OODBMS Environment, Proceedings of Computers in Cardiology, IEEE Computer Society Press, Los Alamitos, 1994: 505-508.
- [3] Das AK, Musen MA. A Temporal Query System for Protocol-Directed Decision Support, Methods of Information in Medicine 1994; 33: 358-370.
- [4] Montanari A, Pernici B. Temporal Reasoning, in [11]: 534 - 562.
- [5] Allen J. Towards a General Theory of Action and Time, Artificial Intelligence 1984; 23: 123-154.
- [6] Shahar Y, Musen MA. A RESUME: A Temporal-Abstraction System for Patient Monitoring, Computers and Biomedical Research 1993; 26: 255-273.
- [7] Brusoni V, Console L, Pernici B, Terenziani P. LATER: a General Purpose Manager of Temporal Information, 8th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October 1994.
- [8] Console L, Torasso P. Temporal Constraint Satisfaction on Causal Models, Information Sciences 1993; 68: 1-32.
- [9] Clifford J, Rao A. A Simple, General Structure for Temporal Domains, In: C. Rolland, F. Bodart and M. Leonard (eds) Temporal Aspects in Information Systems. Amsterdam: Elsevier Science Publishers, North-Holland, 1988: 17-28.
- [10] Sarda N. Extensions to SQL for Historical databases, IEEE Trans. on Knowledge and Data Eng. 1990; 2(2): 220 - 230.
- [11] Tansel AU, Clifford J, Gadia S, Jajodia S, Segev A, Snodgrass R. Temporal Databases: Theory, Design, and Implementation, The Benjamin/Cummings Publishing Company, Redwood City, CA, 1993.
- [12] Haack S. *Philosophy of Logics*, Milano, Franco Angeli, 1983 (in Italian).
- [13] Gadia SK, Nair SS, Poon YC. Incomplete Information in relational temporal databases, in Proceedings of the 18th VLDB Conference, Vancouver, Canada 1992, pp. 395 - 406.
- [14] Corsetti E, Montanari A, Ratto E. Dealing with Different Time Granularities in Formal Specifications of Real-Time Systems, The Journal of Real-Time Systems 1991; 3(2): 191-215.
- [15] Wang XS, Jajodia S, Subrahmanian VS. Temporal Modules: An Approach Toward Federated Temporal Databases, ACM SIGMOD Record 1993; 22: 227 - 236.
- [16] Dechter R, Meiri I, Pearl J. Temporal Constraint Networks, Artificial Intelligence 1991; 49: 61-95.
- [17] Maiocchi R, Pernici B, Barbic F. Automatic Deduction of Temporal Information, ACM Transactions on Database Systems 1992; 17(4): 647 - 688.
- [18] Wiederhold G, Jajodia S, Litwin W. Integrating Temporal Data in a Heterogeneous Environment, in [25]: 564-579.
- [19] Shoham Y. Temporal Logics in AI: Semantical and Ontological Considerations, Artificial Intelligence 1987; 33: 89-104.