

Is Point-Based Semantics Always Adequate for Temporal Databases?

Paolo Terenziani

Dipartimento di Scienze e Tecnologie Avanzate

Universita' del Piemonte Orientale "Amedeo Avogadro", Corso Borsalino 54, Alessandria, Italy

e-mail: terenz@di.unito.it

Abstract

The analysis of the semantics of temporal data and queries plays a central role in the area of temporal relational databases. Although many different models and algebras have been proposed, almost all of them are based on a point-based (snapshot) semantics of the association of time to tuples/attributes. On the other hand, in the areas of Linguistics, Philosophy, and, recently, Artificial Intelligence, a very debated issue concerns the use of an interval-based vs. a point-based semantics. In this paper, we consider a slight adaptation of TSQL2's "consensus" algebra and model, discuss its semantics, and show some critical aspects arisen by the adoption of a point-based semantics.

1 Introduction

Time plays a very important role in real-world phenomena, so that, starting from the 80', the DataBase community devised many different approaches to model it. In particular, many extensions to the standard relational model were devised, and more than 2000 papers on temporal databases were published over the last two decades (see, e.g., the cumulative bibliography in [32] and the surveys in [17,20]). Recently, the TSQL2 approach has consolidated many years of results into a single "consensus" approach [24], which is currently being revised (becoming SQL/Temporal [25]) in order to be accepted by the ISO and ANSI standardization committees. An important issue about the treatment of temporal information in temporal databases (TDB for short) is the analysis of the semantics of data and queries.

A lot of researchers realized the richness of the semantics of temporal data, since often implicit semantics assumptions are used to answer queries, and/or provided

various operators for the users to code the semantic assumptions into the queries [4]. Many works and surveys dealing with the semantics of temporal data modeling have been proposed within the TDB literature (see, e.g., [10,16,21]). Moreover, [25] proposed a detailed analysis of the semantics of queries in SQL/Temporal. As shown by many researchers, temporal relational databases always adopt a *point-based semantics* (see, e.g., [10,16,25,28]) in which the validity of a tuple (or of an attribute, in case of attribute time-stamping) is evaluated on each time point (snapshot). Unfortunately, many researchers in Philosophy, Linguistics and Artificial Intelligence pointed out severe limitations on the point-based semantics, and introduced *interval-based semantics* to overcome them.

On the other hand, to the best of our knowledge, although all recent TDB relational models and algebras are based on a point-based (snapshot) semantics, the analysis of the implications of the adoption of a point-based semantics on such models and algebras, with specific attention to the limitations of such a semantics, has not been analysed yet. Such an analysis is the main issue of this paper. On the other hand, for the sake of brevity, in this paper we do not describe the constructive part of our approach, which is fully presented in [27]. In [27] we introduced a two sorted temporal algebra and data model dealing with two sorts of tables: *telic* tables, based on an interval-based semantics, an *atelic* tables, based on the "consensus" point-based one. Our approach is very powerful and flexible, since it also consists of coercion functions for transforming tables of one sort into tables of the other sort at query time. In [27], we also proved that reduction and equivalence with respect to the classical atemporal algebra hold for our two sorted temporal algebra, and that it is a consistent extension of TSQL2's algebra.

2 Telic vs. atelic facts, point-based vs. interval-based semantics.

A very basic issue underlying many approaches in Philosophy and Linguistics is the fact that the usual human way of "capturing" reality (i.e., of representing it, or of describing it through natural language expressions) involves a **distinction between different classes of facts**. For the sake of brevity, in the following we will discuss such a distinction considering only the analysis carried on within the linguistic community. Within this community, it is commonly agreed that natural language sentences can be classified within different *aktionsart classes* (e.g., *activities*, *accomplishments*, *achievements* and *states* in [30]; also called *aspectual classes* [9]) depending on their linguistic behaviour and/or semantic properties. For example, progressive forms cannot be applied to stative sentences, so that a sentence like "John is being tall" (*state*) is odd, while "John is running" (*activity*) is correct. Many approaches in the linguistic literature took into account semantic criteria to distinguish among different classes of facts. These criteria demonstrate that **the semantics of the association of facts to time depends on the classes of facts** being considered. For example, Dowty [12] proposed the following semantic criteria to distinguish between *states* and *accomplishments*.

"(a) A sentence ϕ is stative iff it follows from the truth of ϕ at an interval I that ϕ is true at all subintervals of I (e.g., if John was asleep from 1:00 to 2:00 PM, then he was asleep at all subintervals of this interval: *be asleep is stative*)

(b) A sentence ϕ is an accomplishment iff it follows from the truth of ϕ at an interval I that ϕ is false at all subinterval of I e.g., if John built a house in exactly the interval from September 1 to June 1, then it is false that he built a house in any subinterval of this interval: *build a house is an accomplishment*)" [12; pp. 42].

The property (a) for states has been often called **downward hereditary** in the TDB and AI literature (see, e.g., [1,4,22]). Notice that also **upward hereditary** holds over states: if John was asleep from 1:00 to 2:00 and from 2:00 to 3:00, then he was asleep from 1:00 to 3:00. Moreover, Dowty and other researchers in Linguistics also considered other classes that are less relevant in this paper, such as *achievements* (i.e., punctual telic facts; e.g., "being promoted") and *activities* (i.e., homogeneous durative dynamic facts; e.g., "walking") in [30].

Obviously, the *aktionsart* distinctions above have a deep impact on the semantic framework one has to adopt to model the meaning of sentences and of the facts they describe. Point-based semantics evaluate the truth of sentences over time points. This semantics perfectly works on stative facts: "John was asleep" in item (a)

above is true exactly for all time points within 1:00 and 2:00 PM. On the other hand, point-based semantics seems to be inadequate to deal with accomplishments. For instance, given (b), there is no specific time point p such that "John built a house" is true in p . "John built a house" is true (or, in other words, occurred) exactly in the *time interval* from September 1 to June 1. This and analogous observations led most researcher in Linguistics, starting from the pioneering works in [3,11,12], to criticise **point-based semantics**, which **is not adequate to deal with the semantics of accomplishments** (while it works well for *states* and *activities*), **for which an interval-based semantics is needed** (i.e., a semantics which evaluates the truth of facts over time intervals [12]). E.g., in an interval-based semantics, *John built a house* in example (b) above is true in *exactly the interval from September 1 to June 1*.

The distinction between **telic facts** (roughly corresponding to Vendler's *accomplishments* and *achievements*) and **atelic facts** (roughly corresponding to Vendler's *states* and *activities*) also emerged in Philosophical field, dating back to Aristotle. Recently, Bach [2] pointed out that telic and atelic facts are two complementary ways of representing reality, and Bloom showed that the telic/atelic distinction has a very strong cognitive evidence [5]. Many AI approaches proposed formalisms distinguishing between different classes of facts (see, e.g. [1,14,26]). E.g., Allen [1] distinguished among states, activities (called *processes*) and accomplishments (called *events*). In his first order logic, Allen introduced three different predicates to associate facts to times, and used an axiomatic approach in order to model the temporal properties of these classes. In [1], the truth of facts is evaluated over time intervals (i.e., an interval-based semantics is adopted). Allen's approach started a whole area of AI approaches adopting time intervals as basic temporal primitives (see, e.g., the survey in [31]). Recently, the ontology devised within the CYC project by MCC distinguishes between Processes (atelic facts) and Events (telic facts) [18]. On the other hand, the impact of the telic/atelic distinction on the semantics of temporal (relational) DataBases have not been studied yet, and will be widely analysed in the rest of this paper.

3 Semantic implications of a point-based semantics on TDB's

The main goal of this paper is that of facing semantic issues related to the adoption of a point-based semantics into temporal relational databases. To investigate this issue, we chose to operate at the algebraic level, to adopt tuple timestamping and to focus on the validity time only. We introduce a data model and a relational algebra which should be conceived as a paradigmatic example of the algebraic relational

approaches using tuple timestamping and point-based semantics. Our model is a reduction (since we only consider validity times) and a slight adaptation of the BCDM simplified data model which has been introduced by Jensen and Snodgrass in order to clarify the semantics of TSQL2 [16]. As regards the algebra, the adaptations we made to TSQL2's algebraic operators are mainly motivated by our goal of showing weaknesses and merits of the point-based semantics. Thus, our goal in this paper is not that of providing a *minimal* and *complete* set of algebraic operators, in the sense of [24; page 524]. The treatment of transaction time, atemporal and valid time event tables [24] are outside the goals of this paper.

3.1 Semantics of the association of (validity) times to tuples

We assume that time is a linear, ordered and discrete set of time points $p \in \mathcal{P}$ (e.g., isomorphic to integers). A time interval i is a convex set of time points between a starting point i^- and an ending point i^+ (i.e., $\forall p \in \mathcal{P} \wedge i^- \leq p \leq i^+ \Rightarrow p \in i$, where $i^- \leq i^+$)¹. As in Gadia [13], we consider closed time intervals, and represent a time point p by $[p, p]$. Let denote I the set of time intervals. As in [17], we adopt the term *fact* for any statement that can be meaningfully assigned a truth value (i.e., that is either true or false), we represent facts with *tuples*, and we use the *valid time* of a fact (tuple) to represent the collected time when the fact is true.

As widely discussed in many papers, including [10,16,17,25,28] the *implicit semantics* of temporal relational models in the literature ([17,20]) is a "*point-based semantics*" (elsewhere called *snapshot semantics*), in the sense that the truth of tuples (facts) is evaluated over time points (or *chronoi*, in [16], see footnote 1). In most cases, time intervals (or sets of time interval) are associated to tuples (or to attributes) instead of time points (see, e.g., the surveys in [17,20]), but this is only a matter of efficient and compact implementation (see e.g., the distinction between *abstract* and *concrete* TDB's in [8]): for instance, in SQL/Temporal [25], TSQL2 [24], TSQL [19], and TQuel [23], stating that a tuple t has a time interval I as its validity time is just a shorthand to stating that the tuple t holds at any time point (or snapshot, or *chronon* [16]) p belonging to I . For example, given a tuple (fact) t , $t([3,6])$ (i.e., the association of the validity time $[3,6]$ to t) is just a *notational shorthand* for $t([3,3]) \wedge t([4,4]) \wedge t([5,5]) \wedge t([6,6])$ -henceforth, we

¹ In BCDM [16], time is seen as a finite sequence of chronos, isomorphic to a sequence of natural numbers. The sequence of chronos can be thought of as representing a partitioning of the time line into equal-sized, indivisible segments. Using chronos instead of discrete time points do not imply any radical change in the issues discussed in this paper.

abbreviate $t([p,p])$ as $t(p)$. Put in an axiomatic way, this can be expressed as in At.1.

$$(At.1) \quad t(i) \Leftrightarrow \forall p \in \mathcal{P}, p \in i \Rightarrow t(p)$$

From At.1, both the so called downward hereditary and the so-called upward-hereditary of "t" trivially follow.

$$(down) \quad t(i) \Rightarrow (\forall p \in \mathcal{P}, p \in i \Rightarrow t(p))$$

$$(up) \quad (\forall p \in \mathcal{P} \ p \in i \Rightarrow t(p)) \Rightarrow t(i)$$

In TSQL2 [24] a set of time intervals is associated with tuples. Let \mathcal{Z}^I denote the sets of set of time intervals, i.e., Gadia's *temporal elements* [13]. Thus, the semantics of associating a temporal element $\{i_1, i_2, \dots, i_n\}$ to a tuple t is:

$$(At^*) \quad t(\{i_1, i_2, \dots, i_n\}) \Leftrightarrow t(i_1) \wedge t(i_2) \wedge \dots \wedge t(i_n)$$

For example, since Gadia and TSQL2 adopted a point-based semantics, $\{[90,92],[93,95]\}$ and $\{[90,95]\}$ denote the very same temporal element. In fact, stating that a tuple t holds over the two intervals $[90,92]$ and $[93,95]$ means that t holds over the snapshots 90, 91, 92, 93, 94, and 95, which is equivalent to stating that t holds over the time interval $[90,95]$.

3.2 Algebraic operators on temporal elements

Gadia defined the operations of union, intersection and complement over set of time intervals (these operators are also used, e.g., in TSQL2 and SQL/Temporal, and are indicated by \cup^A , \cap^A , \neg^A in this paper). Although an explicit formal definition is not provided by Gadia [13], it is clear that these operations are defined consistently with the point-based semantics. In practice, in Gadia's approach, as well as in all approaches using point-based semantics, one can interpret time intervals (and thus *temporal elements*) as a mere notation to represent sets of time points, and define union, intersection and complement on temporal elements as a *restriction of standard set operators to the sets of time points*. More formally, Gadia's set operators could be defined as follows (where \mathcal{P} is the domain of time points, and \mathcal{Z}^p stands for sets of time points):

$$\text{Def. } \cup^A \quad \forall S1, S2 \in \mathcal{Z}^p \quad S1 \cup^A S2 \equiv S1 \cup S2$$

$$\text{Def. } \neg^T \quad \forall S \in \mathcal{Z}^p \quad \neg^A S \equiv \neg S$$

$$\text{Def. } \cap^T \quad \forall S1, S2 \in \mathcal{Z}^p \quad S1 \cap^A S2 \equiv S1 \cap S2$$

For example, \cup^A is defined as the restriction of standard set union to the case where the elements of the domain are time points. Thus, $\{[10,12],[14,16]\} \cup^A \{[11,15],[20,21]\}$ gives as result $\{[10,16],[20,21]\}$, which stands for the set of points $\{10,11,12,13,14,15,16,20,21\}$.

Notice that temporal *coalescing* is performed over time intervals by union, so that upward hereditary is forced (see axiom Up).

3.3 Relational model

As in TSQL2's *historical tables*, in our model a **temporal table** is a table in which each tuple has a validity time, which is a set of time intervals (i.e., a *temporal element* [13]). Given the semantics above (see also [16]), the case in which just one time interval is associated with each tuple (see, e.g., [19]) can be seen as a subcase of the model above, in which only unitary sets of intervals are used.

The adoption of a point-based semantics has a crucial impact on the semantics of temporal tables. In particular, two temporal tables (relations) with the same atemporal attributes are *snapshot equivalent* if for all valid times, their snapshots are identical (see [16]). Thus, for instance, tables T1 and T2 in Figure 3 are two equivalent ways of expressing the very same content (which is abstractly shown in table T3).

T1	Atemporal attributes	VT
	t	{[90,92], [93,95]}
T2	Atemporal attributes	VT
	t	{[90,95]}
T3	Atemporal attributes	VT
	t	{90,91,92,93,94,95}

Figure.1 T1 and T2 are snapshot equivalent tables. The semantics of both tables is shown by T3.

For the sake of simplicity, as in TSQL2 [24] and BCDM [16] we disallow value-equivalent tuples (i.e., two tuples with mutually identical atemporal attributes). Finally, notice that, as discussed in [16; pp. 323], “*the natural extension of a conventional relation to a temporal relation encodes states* (i.e., atelic facts, in our terminology) *instead that events*” (i.e., telic facts). This is a consequence of the adoption of a point-based semantics for TDB's. The truth of tuples (facts) is evaluated on time points. Thus, the database collects a set of snapshots of the *mini-world* (see [16]) it represents. In other words, the mini-world is represented as a set of states (one for each temporal snapshot in the database).

3.4 Relational algebra

We use a slight adaptation of the algebra of TSQL2 [24, section 27]. Notice also that the definitions of some operators (e.g., intersection and temporal restriction) is very similar to Gadia's *computational semantics* definitions [13], adapted here to the case of tuple timestamping.

First, we define the operators of selection, projection, cartesian product, difference and union, which extend the standard snapshot algebraic operators (i.e., $\sigma, \pi, \times, -, \cup$) to operate on *temporal tables*. We indicate by $\sigma^{AT}, \pi^{AT}, \times^{AT}, -^{AT}, \cup^{AT}$ the operators on temporal tables. In the following, we indicate with $Sch(R)$ the data attributes of a temporal table R. Given a tuple t of a temporal table R, we indicate by $t(R)$ the value of the data attributes in t (i.e., $t(R)$ is a shorthand for $t(Sch(R))$), and by $t(VT)$ its validity time.

(Atemporal) Selection. (Atemporal) selection selects the tuples whose data part satisfies a condition ϕ (which is a condition on the data part only), regardless of its temporal part.

$$Sch(\sigma_{\phi}^{atemp}(R)) = Sch(R)$$

$$\sigma_{\phi}^{atemp}(R) = \{t \mid t \in R, \phi(t(R))\}$$

Cartesian product. As in TSQL2, we chose to adopt the intersection semantics for the cartesian product. Thus, the validity time of the resulting tables is the intersection of the validity times of the tuples.

$$Sch(R1 \times^{AT} R2) = Sch(R1) \cup Sch(R2)$$

$$R1 \times^{AT} R2 = \{s \mid \exists t1 \in R1, \exists t2 \in R2, s(R1)=t1(R1), s(R2)=t2(R2), s(VT) = t1(VT) \cap^A t2(VT), s(VT) \neq \emptyset\}$$

Union. The union operator \cup^{AT} makes the union of two temporal tables R1 and R2. Tuples with different data part are put unchanged into the resulting table. On the other hand, Gadia's union \cup^A is performed on the validity times of tuples with equal data part.

$$Sch(R1 \cup^{AT} R2) = Sch(R1) \cup Sch(R2)$$

$$R1 \cup^{AT} R2 = \{s \mid (\exists t1 \in R1, \text{Not } \exists t2 \in R2 \ t1(R1)=t2(R2), s(R1)=t1(R1), s(VT) = t1(VT)), \text{ or } (\exists t2 \in R2, \text{Not } \exists t1 \in R1 \ t1(R1)=t2(R2), s(R1)=t2(R2), s(VT) = t2(VT)), \text{ or } (\exists t1 \in R1, \exists t2 \in R2 \ t1(R1)=t2(R2), s(R1)=t1(R1), s(VT) = t1(VT) \cup^A t2(VT))\}$$

Projection. Projection can be used to select data attributes. In case the data parts of the tuples obtained by projection on the attributes B is different, the validity times are left unchanged by Projection. On the other hand, for each tuple t such that there is at least another tuple t' in R such that $t(B)=t'(B)$ ($R(B)$ indicates the restriction of a table R to the attributes in B) the union of the validity times must be performed, as in the case of union above. More precisely, projection is defined as follows (where $\forall t'_1, \dots, t'_k \in R \ P(t'_1, \dots, t'_k)$ is a shorthand for indicating all and only the tuples of R such that $P(t'_1, \dots, t'_k)$ holds):

$$Sch(\pi_B^{AT}(R)) = B \ (B \subseteq Sch(R))$$

$$\pi_B^{AT}(R1) = \{s \mid (\exists t1 \in R, \text{Not } \exists t2 \in R \ t1(B)=t2(B), s(B)=t1(B), s(VT)=t1(VT)) \text{ or } (\forall t'_1, \dots, t'_k \in R, \ t'_1(B)=\dots=t'_k(B), s(B)=t'_1(B), s(VT)=t'_1(VT) \cup^A \dots \cup^A t'_k(VT))\}$$

Difference. The difference operator between two periodic tables R1 and R2 gives as result a table containing all the tuples of R1 which are different in the data part from all tuples in R2. Tuples with the same data part are dealt with considering the difference of their validity times. More formally:

$Sch(R1 \text{ } ^{AT} R2) = Sch(R1) \text{ } (Sch(R1) = Sch(R2))$
 $R1 \text{ } ^{AT} R2 = \{s \mid (\exists t1 \in R1, \text{ Not } \exists t2 \in R2 \text{ } t1(R1)=t2(R2),$
 $s(R1)=t1(R1), s(VT) = t1(VT)) \text{ or } (\exists t1 \in R1 \exists t2 \in R2$
 $t1(R1)=t2(R2), s(R1)=t1(R1), s(VT) = t1(VT) \text{ } ^A t2(VT),$
 $s(VT) \neq \emptyset\}$

Temporal Restriction. We adapt Gadia's temporal selection operator [13] (henceforth called temporal restriction and indicated by $Restr^{AT}_{[I]}$, where I is a time interval) to the case of tuple timestamping. Notice that temporal restriction is the only temporal operator introduced in [13], and restricts the validity time of the tuple to the input time interval I.

$Sch(Restr^{AT}_{[I]}(R1)) = Sch(R1)$

$Restr^{AT}_{[I]}(R1) = \{s \mid (\exists t1 \in R1, s(R1)=t1(R1), s(VT) =$
 $(t1(VT) \cap^A \{I\}), s(VT) \neq \emptyset\}$

Temporal Selection. We introduce a temporal selection operator which is analogous to the atemporal selection above, but imposes a condition on the validity time of tuples. As in [6], our temporal selection selects the tuples whose validity time contains at least a time interval which satisfies a temporal condition ϕ (which is a condition on the validity time only), regardless of its data part. The validity time of the resulting tuples contains only the intervals in the validity time which satisfies ϕ . This operator is a slight modification of temporal selection in TSQL2 [24], in which the validity time of the selected tuples is left unchanged in the output.

$Sch(\sigma^{AT}_{\phi}(R)) = Sch(R)$

$\sigma^{AT}_{\phi}(R) = \{s \mid \exists t \in R, s(R)=t(R), s(VT)=TE, TE \neq \emptyset\}$
 where $TE = \{i \in t(VT) \mid \phi(i)\}$

Left (right) temporal join. This operator is introduced in order to be able to compare the validity times of different tuples. It takes in input two tables R1 and R2 and a comparison operator ϕ (e.g., Before) and gives as output the concatenation of the input tuples $t \in R1$ and $t' \in R2$ such that there is at least an interval $i \in t(VT)$ and an interval $i' \in t'(VT)$ such that $\phi(i, i')$ holds. The validity time of the output tuples is the set of intervals $i \in t(VT)$ which satisfy the condition in case of left temporal join (the set of intervals $i' \in t'(VT)$ which satisfy the condition in case of right temporal join). In the following, only left temporal join is presented (right temporal join is analogous).

$Sch(l\text{-join}^{AT}_{\phi}(R1, R2)) = Sch(R1) \cup Sch(R2)$

$l\text{-join}^{AT}_{\phi}(R1, R2) = \{s \mid \exists t \in R1 \exists t' \in R2, s(R1)=t(R1),$
 $s(R2)=t(R2), s(VT)=TE, TE \neq \emptyset\}$

where $TE = \{i \in t(VT) \mid \exists i' \in t'(VT) \text{ such that } \phi(i, i')\}$

4 On the impact of aktionsart distinctions on TDB: some semantic issues

The treatment of telic facts in the temporal relational algebra based on point-based semantics arise several problems. The most important are, in our opinion, an

incorrect use of *downward* and *upward hereditary* and the problem of *countability* of tuples.

It is important to stress that the problems below concern *data expressiveness* of point-based TDB models, and not their *query expressiveness* (point-based vs. interval-based query languages have been widely compared, e.g., in [7,8]). In the following, we exemplify these problems using a very simple example of DataBase, and by showing that different “usual” queries on such a DB provide unexpected (and not correct) answers, due to the fact that *data* are interpreted using a point-based semantics (or, in other words, because the desired *temporal data cannot be properly represented* within a DB model based on such a semantics).

Our example concerns phone calls. An event such as making a phone call starting at a given time $t1$ and ending in $t2$ is a telic durative fact (accomplishment). For instance, if I made a call from 100 to 110, I didn't make it from 102 to 105. Similarly, two consecutive calls, one from 90 to 92 (included) and the other from 93 to 95, are clearly different from one single call from 90 to 95². However, such a difference *cannot be captured in a point-based TDB model* (unless one introduces additional ad-hoc attributes –see section 5). For example, let us consider a TDB (based on the model in section 3) consisting of the table PHONE in Figure 2. The attributes Caller and Called represent the phone number of the called and of the receiver of the call respectively. In PHONE, phone numbers are represented by n_i and validity times endpoints are represented directly by natural numbers (instead that by dates).

PHONE

Caller	Called	VT
n1	n2	{[90,92],[93,95]}
n7	n8	{[94,120]}
n6	n4	{[90,120]}
n3	n5	{[90,95]}

Figure 2

(1) **Downward hereditary** implicitly holds on the data model (if point based semantics is assumed). This is not correct in the case of the PHONE table, and leads, e.g., to obtaining wrong answers to queries such as AQ1 below. Notice that AQ1 is a way of asking query Q1 using temporal restriction (the only operator provided by Gadia in [13] in order to perform temporal selection).

² For instance, in the current Italian phone system, the cost of a call is evaluated as the sum of a fixed amount, which has to be paid for each single call (independently of its duration) plus a variable amount, depending on many factors, including duration and distance. Thus, one call from n1 to n2 starting at 90 and ending at 95 is cheaper than two consecutive calls taking the same amount of time.

(AQ1) $\pi_{\text{Caller}}^{\text{AT}}(\text{Restr}_{\{[90,95]\}}^{\text{AT}}(\text{PHONE}))$

(Q1) Who made a call starting at 90 and ending at 95?

For the sake of clarity, let us suppose for the moment that the table PHONE only contains the second, the third and the fourth tuples. Then, the result of the query (AQ1) on the table in Figure 2 is (ANSW1)

(ANSW1) $\{ \langle n7, \{[94,95]\} \rangle, \langle n6, \{[90,95]\} \rangle, \langle n3, \{[90,95]\} \rangle \}$.

Notice, however, that only n3 made a call starting exactly at 90 and ending exactly at 95. On the other hand, n6 is included in the results, since downward hereditary implicitly holds on the data model, so that if n1 was making a call from 90 to 120, it was also making it in the subinterval [90,95]. Analogously, n7 was calling in the subinterval [94,95], so that it is included in the resulting table³.

(2) **Upward hereditary** holds on the data model. This is not correct when dealing with telic facts such as phone calls. For example, let us consider also the first tuple in Figure 2, stating that n1 made two consecutive calls to n2 which meets in time, the first from 90 to 92, and the second from 93 to 95. This fact *cannot be properly represented within point-based semantics*, where the first tuple in PHONE is interpreted as stating that n1 called n2 from 90 to 95. As a consequence, the answers to queries such as AQ1 and AQ1' are ANSW3 and ANSW4 respectively:

(ANSW3) $\{ \langle n1, \{[90,95]\} \rangle, \langle n7, \{[94,95]\} \rangle, \langle n6, \{[90,95]\} \rangle, \langle n3, \{[90,95]\} \rangle \}$

(ANSW4) $\{ \langle n1, \{[90,95]\} \rangle, \langle n3, \{[90,95]\} \rangle \}$

One would have analogous problems when trying to consider the duration of phone calls: for instance, the results for query (AQ2) is again ANSW4

(AQ2) $\pi_{\text{Caller}}^{\text{AT}}(\sigma_{\text{duration}=5}^{\text{AT}}(\text{PHONE}))$

(Q2) Which numbers made phone calls lasting 5 units?

Finally, within the snapshot semantics, the answer to the query (AQ3) is empty. In fact, since $\{[90,92],[93,95]\}$ is interpreted as the interval [90,95], the time interval [94,120] does not follow any interval in the validity time of the first tuple, so that the tuple $\langle n7, \{[94,120]\} \rangle$ is not part of the result.

(AQ3) $\pi_{\text{Caller}}^{\text{AT}}(\text{l-join}_{\text{After}}^{\text{AT}}(\text{PHONE}, (\sigma_{\text{atemp}}^{\text{AT}}(\text{PHONE}))))$
caller=n1, called=n2

³ Notice that, if the temporal algebra also contains a temporal selection operator such as σ^{AT} in section 3, plus the functions Start and End which returns the starting and ending points of the intervals in the temporal element, one can express query Q1 as shown by (AQ1'), which provides the desired results (see ANSW2).

(AQ1') $\pi_{\text{Caller}}^{\text{AT}}(\sigma_{\text{Start}=90, \text{End}=95}^{\text{AT}}(\text{PHONE}))$
 (ANSW2) $\{ \langle n3, \{[90,95]\} \rangle \}$.

(Q3) Which numbers made at least a call after one of the calls of n1 to n2?

(3) **Countability.** A further problem concerns **countability**. Consider, e.g., query Q4

(Q4) How many calls were made by n1?

In the point-based semantics, $\{[90,92],[93,95]\}$ is interpreted as $\{[90,95]\}$, so that the answer to Q4 would be: 1.

5 Impact of the limitations point-based semantics

When evaluating the impact of the limitations we proposed, one could argue that

- (i) They might look very marginal, in case the distinction between telic (accomplishments) and atelic (states and activities) durative facts is not relevant, or if the treatment of telic facts is useless in temporal databases
- (ii) even in the case of durative telic facts, the problems in section 4 are mostly “technical” problems which only occur for tuples having more than a time interval in their validity times.

The relevance of the distinction between telic and atelic facts has been already discussed and motivated in section 2, where we showed that, according with the philosophical and linguistic and cognitive analysis, telic facts play a fundamental role in our way of approaching and describing the world. Therefore, we believe that also the field of temporal databases could greatly benefit from an approach which can deal with both telic and atelic facts, especially if one considers the current trend of dealing with office automation, guidelines, schedules of activities and so on (which mostly involve telic agentive facts) within DB's.

From the semantic point of view, in this paper we showed that the treatment of *atelic facts* corresponds to the adoption of a *point-based semantics*, since the truth of facts is evaluated over time points. From the point of view of representing a data/knowledge, it corresponds to a *status-based representation* (see [16]). On the other hand, the treatment of *telic facts* usually implies the use of an *interval-based semantics* (see [3,11,12]). From the point of view of representing a data/knowledge, it corresponds to an *event-based representation*. This means that, for instance, the temporal databases representing (possibly time-varying) properties of entities (e.g., salary, taxes etc. of employees) can be easily and naturally captured by temporal databases based on a point-based semantics. On the other hand, such approaches have the problems discussed in section 4 when dealing with durative events in office automation, or in workflows, guidelines and protocols; the telephone example we discussed in this paper is just one prototypical case of a family of similar cases.

Point (ii) might be more controversial. Until now, we pointed out certain problems of the point-based semantics considering tuple timestamping in which a set of intervals (i.e., a temporal element) can represent the validity time of a tuple. However, one could argue that all problems vanish in the case that a unique time interval is associated to each tuple, and that unique-interval timestamping can always be achieved by adding additional attributes (e.g., adding an ID number for each phone call). However, from one side this means that, whenever dealing with tables representing telic facts, DB designers would be forced to add undesired *ad-hoc* attributes, which only exist in order to avoiding the problems discussed above in this paper. From the other side, projection should be restricted, since the additional ad-hoc attributes can never be removed (otherwise, such problems re-appear). More generally, in [27] we showed that problems like the ones we discussed in section 4 have to be faced even in the case of unique-interval tuple timestamping (i.e., if we impose a temporal 1NF – see [13] as, e.g., in TSQL [19]). This is in accordance with Jensen and Snodgrass' analysis in [16], where the authors have pointed out that the distinction between 1NF with respect to Not-1NF is relevant at the representation level, but *not at the conceptual and semantic level* [16].

Another way for trying to avoid the problems we discussed in section 4 is that of imposing unique interval timestamping in the data model, and *never* performing coalescing of value-equivalent tuples, as SQL\Temporal [25]. In such a case one cannot use a set-based semantics (as we did in this paper), since *duplicates* must be dealt with. However, an approach which never performs coalescing has dual problems with respect to the problems discussed in section 4, when dealing with atelic facts. E.g., given the table WORK, representing the salaries of the employees, and supposing that no temporal coalescing is performed (e.g., by temporal projection), the answers to the queries Q5: "Who worked all the time between 85 and 95?" and Q6: "Who worked continuously for at least 30?" is the empty table.

WORK

Employee	Salary	VT
Bob	30K	{[75,90]}
Bob	40K	{[91,110]}

Figure 3

6 Conclusions and Discussions

The analysis of the semantics of temporal data plays a central role in the area of temporal databases, since "data explicitly stored in a temporal database are often associated with certain semantics assumptions" [4, page 277]. Although almost all temporal relational approaches in the TDB literature are based on a point-based

(snapshot) semantics for the association of tuples (or attributes) to time (see, e.g., the discussions in [10,13,16,25,28]), not all the semantic implications involved by such adoption have been fully investigated by the specialised TDB literature. In this paper, we investigated some of these implications, taking into account the basic distinction between telic and atelic facts, as emerging by the linguistic (and philosophical) literature. We first considered a slight adaptation of TSQL2's "consensus" algebra and model, pointed out some semantics implications due to the adoption of a point-based semantics (e.g., axioms Up and Down in section 3.1, and the definitions of Gadia's operators in section 3.2) and showed some critical problems arisen by the adoption of a point-based semantics when dealing with telic facts in temporal DB's. Moreover, although we only considered tuple timestamping, Jensen and Snodgrass also argued that "*the distinction <between attribute and tuple timestamping> is not one of semantics*" [16, pp. 324]. Thus, our criticism could also be extended to the approaches using attribute timestamping, as, e.g., in [13]).

As regards related works, we pointed out throughout the paper that our temporal algebra and model is a slight adaptation of TSQL2's one. Within the AI literature, many works discussed the point-based vs. interval-based semantics issue (see, e.g., [1,14,15,26,29]). Many approaches investigated different aspects of the semantics of TDB's (see, e.g., [4,10,16,21,25,28]). In particular, Chomicki and Toman [7,8] discussed the point vs. interval issue in TDB's, focusing on the semantics of *queries*. On the other hand, in this paper we mainly focus on the semantics of *data* models. Finally, some TDB approach dealing with *temporal interpolation* [16] (consider, e.g., [4]) have faced problems such as upward and downward hereditary in order to *derive information for times for which no information is stored, on the basis of related information holding at different times*. However, in [4], upward and downward hereditary have been only studied for evaluating the values of attributes whose validity time is expressed at different time granularities.

Throughout this paper, we have often used linguistic motivations in order to point out several problems of temporal DataBase models based on point-based semantics. Thus, we believe that it is important, before the end of the paper, to stress why we believe that paying so much attention to the linguistic issues is very important. Basically, we strongly agree with Moens and Steedman's claim that "*effective exchange of information between people and machines is easier if the data structures that are used to organize the information in the machine correspond in a natural way to the conceptual structures people use to organize the same information*" [9; page 26]. Moreover, as widely accepted within the Linguistic

and philosophical communities, we also think that, in some sense, *natural language "mirrors" the human way of looking at the world and of organizing data/knowledge*. Thus, we believe that providing (temporal) DataBases with facilities for representing data/knowledge which reflect the way such data/knowledge is organised in linguistic sentences is the best way to obtain human-oriented and user-friendly (temporal) DBMS. Moreover, such a "natural-language-based" organization could also simplify the attempt of building natural language interfaces to (temporal) databases.

We believe that the major original contribution of this paper is that of bringing to the attention of the TDB community some limitations of the point-based semantics when coping with the semantics of *data* in TDB's, and of grounding such an analysis on the telic/atelic dichotomy, which has emerged from the philosophical and linguistic communities and is at the basis of many AI approaches. As sketched in the introduction, our overall approach also consists of a constructive part (which is not described here for the sake of brevity), where we defined a two sorted algebra and model reconciling point-based and interval based semantics for TDB's (see [27]).

Acknowledgements

The initial impulse to working on this paper arose from a very stimulating discussion I had in Tucson with C. Jensen and R.T. Snodgrass, in which we started to debate some of the differences between the usual (point-based) semantics adopted in TDB and the interval-based one used by many AI approaches. I am very grateful to them for their very deep insights and their initial suggestions.

References

- [1] J.F. Allen: "Towards a general theory of action and time", *Artificial Intelligence* 23, 123-154, 1984.
- [2] E. Bach, "The algebra of events", *Linguistics and Philosophy* 9, 5-16, 1986.
- [3] M. Bennet and B. Partee: "Tense and discourse location in situation semantics", Indiana University Linguistics Club, Bloomington, 1978.
- [4] C. Bettini, X.S. Wang, and S. Jajodia: "Temporal semantic assumptions and their use in databases", *IEEE Trans. On Knowledge and Data Engineering* 10(2), 1998.
- [5] L. Bloom, L.Lifter, J. Hafitz: "Semantics of verbs and the developments of verb inflection in child language", *Language* 52(2), 386-412, 1980.
- [6] V. Brusoni, L. Console, B. Pernici, and P. Terenziani: "Qualitative and Quantitative Temporal Constraints and Relational Databases: Theory, Architecture, and Applications", *IEEE Transactions on Knowledge and Data Engineering* 11(6), 948-968, 1999.
- [7] J. Chomicki and D. Toman, "Point-Based vs. Interval-Based Temporal Query Languages", *Proc. ACM PODS*, 58-67, 1996.
- [8] J. Chomicki and D. Toman, "Temporal Logic in Information Systems" in J. Chomicki and G. Saake (eds.) *Logics for Databases and Information Systems*, Kluwer Academic, 1998.
- [9] *Computational Linguistics* 14(2), Special Issue on Tense and Aspect, B. Webber (guest ed.), 1988.
- [10] J. Clifford and D. Warren: "Formal semantics for time in temporal databases", *ACM Transactions on Database Systems* 8(2), 214-254, 1983.
- [11] D. Dowty, "Word meaning and Montague grammar", D. Riedel, Dordrecht, 1979.
- [12] D. Dowty, "The effects of the aspectual class on the temporal structure of discourse", *Tense and Aspect in Discourse, Linguistics and Philosophy* 9(1), 37-61, 1986.
- [13] S.K. Gadia: "A homogeneous relational model and query languages for temporal databases", *ACM Transactions on Database Systems* 13(4), 418-448, 1988.
- [14] A. Galton: "A critical examination of Allen's theory of action and time", *Artificial Intelligence* 42, 159-188, 1991.
- [15] E. Hajnicz, "Time Structures", LNAI 1047, Springer Verlag, 1995.
- [16] C.S. Jensen, R.T. Snodgrass: "Semantics of Time-varying information", *Information Systems* 21(4), 311-352, 1996.
- [17] C. Jensen, and R.T. Snodgrass: "Temporal Data Management", *IEEE Transactions on Knowledge and Data Engineering* 11(1), 36-44, 1999.
- [18] D.B. Lenat and R.V. Guha: "Building Large Knowledge-Based Systems. Representation and inference in the Cyc project", Addison Wesley Publishing Company, 1994.
- [19] S. Navathe, and R. Ahmed: "A temporal relational model and a query language", *Information Sciences* 49, 147-175, 1989.
- [20] G. Ozsoyoglu and R.T. Snodgrass: "Temporal and real time databases: a survey", *IEEE Transaction On Data and Knowledge Engineering* 7(4), 513-532, 1995.
- [21] J.F. Roddick, J.D. Patrick: "Temporal semantics in information systems: a survey", *Information Systems* 17(3), 249-267, 1992.
- [22] Y. Shoham, "Temporal logics in AI: semantical and ontological considerations", *Artificial Intelligence* 33, 89-104, 1987.
- [23] R.T. Snodgrass: "The temporal language TQuel", *ACM Transactions on Database Systems* 12(2), 247-298, 1987.
- [24] R.T. Snodgrass (editor), I. Ahn, G. Ariav, D. Batory, J. Clifford, C. Dyreson, R. Elmasri, F. Grandi, C. Jensen, W. Kafer, N. Kline, K. Kulkarni, C. Leung, N. Loretzos, J. Roddick, A. Segev, M. Soo, S.M. Sripada: "The temporal query language TSQL2", Kluwer Academic, 1995.
- [25] R.T. Snodgrass, M.H. Bohlen, C.S. Jensen, and A. Steiner: "Transitioning temporal support in TSQL2 to SQL3", in O. Etzion, S. Jajodia, and S. Sripada (eds.): *Temporal Databases: Research and Practice*, LNCS 1399, Springer Verlag, 150-194, 1998.
- [26] P. Terenziani and P. Torasso: "Time, action-types and causation: an integrated analysis", *Computational Intelligence* 11(3), 529-552, 1995.
- [27] P. Terenziani: "Reconciling point-based and interval-based semantics in temporal relational databases: a two-sorted algebra with coercion functions", Technical Report, Dipartimento di Informatica, Universita' di Torino, 2000.
- [28] D. Toman: "Point-based temporal extensions of SQL and their efficient implementation", in O. Etzion, S. Jajodia,

- and S. Sripada (eds.): *Temporal Databases: Research and Practice*, LNCS 1399, Springer Verlag, 211-237, 1998.
- [29] J. Van Benthem, "*The logic of Time*", D. Riedel, 1983.
- [30] Z. Vendler: "Verbs and times", *Linguistics in Philosophy*, Cornell University Press, New York NY, 97-121, 1967.
- [31] L. Vila: "A survey of temporal reasoning in artificial intelligence", *AI Communications* 7(1), 4-128, 1994.
- [32] Y. Wu, S. Jajodia, and X.S. Wang: "Temporal Database Bibliography Update", 338-366, in O. Etzion, S. Jajodia, and S. Sripada (eds.): *Temporal Databases: research and practice*, LNCS 1399, Springer Verlag, 1998.