Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases: A Proper Treatment of the Telic/Atelic Distinction

Paolo Terenziani and Richard T. Snodgrass

June 18, 2001

TR-60

A TIMECENTER Technical Report

Title | Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases:
A Proper Treatment of the Telic/Atelic Distinction

Copyright © 2001 Paolo Terenziani and Richard T. Snodgrass. All rights reserved.

Author(s) | Paolo Terenziani and Richard T. Snodgrass

Publication History | June 2001. A TimeCenter Technical Report.

TimeCenter Participants

Aalborg University, Denmark
Christian S. Jensen (codirector), Michael H. Böhlen, Heidi Gregersen, Dieter Pfoser, Simonas Šaltenis, Janne Skyt, Giedrius Slivinskas, Kristian Torp

University of Arizona, USA
Richard T. Snodgrass (codirector), Dengfeng Gao, Vijay Khatri, Bongki Moon, Sudha Ram

Individual participants
Curtis E. Dyreson, Washington State University, USA
Fabio Grandi, University of Bologna, Italy
Nick Kline, Microsoft, USA
Gerhard Knolmayer, Universty of Bern, Switzerland
Thomas Myrach, Universty of Bern, Switzerland
Kwang W. Nam, Chungbuk National University, Korea
Mario A. Nascimento, University of Alberta, Canada
John F. Roddick, University of South Australia, Australia
Keun H. Ryu, Chungbuk National University, Korea
Michael D. Soo, amazon.com, USA
Andreas Steiner, TimeConsult, Switzerland
Vassilis Tsotras, University of California, Riverside, USA
Jef Wijsen, University of Mons-Hainaut, Belgium
Carlo Zaniolo, University of California, Los Angeles, USA

For additional information, see The TimeCenter Homepage:
URL: `<http://www.cs.auc.dk/TimeCenter>`

Any software made available via TimeCenter *is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranty of merchantability and fitness for a particular purpose.*

The TimeCenter icon on the cover combines two "arrows." These "arrows" are letters in the so-called *Rune* alphabet used one millennium ago by the Vikings, as well as by their precedessors and successors. The Rune alphabet (second phase) has 16 letters, all of which have angular shapes and lack horizontal lines because the primary storage medium was wood. Runes may also be found on jewelry, tools, and weapons and were perceived by many as having magic, hidden powers.

The two Rune arrows in the icon denote "T" and "C," respectively.

Abstract

The analysis of the semantics of temporal data and queries plays a central role in the area of temporal databases. Although many different algebræ and models have been proposed, almost all of them are based on a point-based (snapshot) semantics for data. On the other hand, in the areas of linguistics, philosophy, and, recently, artificial intelligence, a debated issue concerns the use of an interval-based versus a point-based semantics. In this paper, we first show some problems inherent in the adoption of a point-based semantics for data, and argue that these problems arise because there is no distinction drawn in the data between *telic* and *atelic facts*. We then introduce a three-sorted temporal model and algebra which properly copes with these issues, and which achieves a great flexibility via the introduction of coercion functions for transforming relations of one sort into relations of the other at query time. We show that it is possible to extend SQL/Temporal in a minimal fashion to support this augmented algebra.

1 Introduction

Time plays an important role in real-world phenomena, and so there has been much work over the last two decades in incorporating time into data models, query languages, and database management system (DBMS) implementations. In particular, many extensions to the standard relational model were devised, and more than 2000 papers on temporal databases (TDBs) were published over the last two decades (cf., the cumulative bibliography in [Wu et al. 98], the **surveys in [McKenzie & Snodgrass 91, Tansel et al. 94, Özsoyoğlu & Snodgrass 95, Jensen & Snodgrass 99] and the** workshop proceedings in [IWITD 93, RATD 95, TDRP 99]).

An important issue about the treatment of temporal information in temporal databases is the semantics of data and queries. Many researchers realized the richness of the semantics of temporal data [Clifford & Tansel 85, Segev & Shoshani 87, Tansel 87] and provided various operators in different query languages [Snodgrass 87, Segev & Shoshani 87, Tansel 87, Bettini et al. 98]. Many works and surveys dealing with the semantics of temporal data modeling have been proposed within the temporal database literature (cf., e.g., [Bubenko 77, Ariav et al. 83, Clifford & Warren 83, Clifford & Ariav 86, Peckham & Marianski 88, Ling & Bell 90, Roddick & Patrick 92, Clifford & Isakowitz, 94, Jensen & Snodgrass 96]). In a recent paper, Toman [96] pointed out some problems connected with the definition of a clear semantics for the approaches where the validity times of tuples and attributes are encoded using time intervals. In fact, in most temporal models, time intervals (or sets of time intervals) are associated with tuples (or with attributes) instead of time points, but this is only a matter of efficient and compact implementation [Chomicki & Toman 98]. Many works (cf., e.g., [Clifford & Warren 83, Jensen & Snodgrass 96, Toman 98]) showed that, for instance, in SQL/Temporal [Snodgrass et al. 95b, 98], TSQL2 [Snodgrass et al. 95a], TSQL [Navathe & Ahmed 87, 89], HQL [Sardeghi 87], and TQuel [Snodgrass 87], data can be seen as a sequence of states indexed by time points. Thus, such approaches adopt a *point-based semantics for data*, in the sense discussed in Section 2 below.

A point-based semantics has significant limitations, which have been widely studied in the areas of philosophy, linguistics and artificial intelligence (henceforth: AI). However, the distinction between point-based and interval-based approaches in TDBs (see, e.g., [Toman 96, 98, Chomicki and Toman 98, Böhlen et al. 98, 00]) is an entirely different one than the distinction in linguistics and AI. In this paper we show how the latter distinction can be profitably incorporated into temporal models and query languages, and in fact we argue that this distinction is in some ways more central than the similarly-named but orthogonal distinction in temporal databases.

We first make clear the terminology we adopt in the paper, relating it to the current one in the area of TDBs (Section 2). Then, in Section 3 we sketch the previously-made distinction between two classes of facts, *telic* and *atelic* facts, emphasizing that the semantics of the association of facts to time depends on the class of fact being considered, and relating the point-based and interval-based semantics to these two classes. To the best of our knowledge, an analysis of the impact of the limitations of point-based semantics on temporal database models and algebræ which adopt it (on the basis of the telic/atelic distinction) has not been done, and is one of the main contributions of this paper.

In Section 4, we propose a “prototypical” temporal relational model which adopts tuple validity time timestamping and point-based semantics, and which is suitable to deal with atelic facts. Section 5 briefly describes an algebra for such a model, which is based on an algebra supporting SQL/Temporal. We then propose a new and original model (Section 6) and algebra (Section 7) which adopts an interval-based semantics in order to deal with telic facts in TDBs, plus the introduction of suitable coercion functions which allow one to transform telic relations into atelic ones, and vice versa, in order to add query expressiveness (Section 8). Section 9 sketches an extension of the SQL2/Temporal query language to deal effectively with both point-based and interval-based relations. Finally, in Section 11 we evaluate our approach in the context of other temporal models. Section 12 revisits the point-based versus interval-based controversy, and Section 13 describes conclusions and future work. Appendix 1 contains part of the proofs, Appendix 2 discusses in more detail the linguistic distinctions among different classes of facts (events), with specific focus on the telic/atelic dichotomy, and Appendix 3 sketches the relevance of such distinctions in different fields of research, including philosophy, cognitive science and artificial intelligence.

2 Preliminaries

The goal of this paper is to augment the usual point-based semantics with an interval-based semantics for data into temporal relational databases, motivated by analogous linguistic and philosophical distinctions. We chose to operate at the *algebraic level*, to adopt *tuple timestamping* and to focus on *valid time*. Moreover, for the sake of clarity and simplicity, we do not consider valid-time event relations [Snodgrass et al. 95a], and we assume discrete time and a unique granularity for all relations. In Sections 4 and 5, we introduce a relational model and algebra, an adaptation of that of SQL/Temporal, which should be taken as a paradigmatic example of the algebraic relational approaches using tuple timestamping and point-based semantics for data. However, our goal in this paper is not that of providing a minimal nor complete set of algebraic operators, in the sense of [Soo et al. 95, page 524]. The modifications we made to SQL/Temporal’s algebraic operators are primarily motivated by our goal of demonstrating relative weaknesses and strengths of point-based and interval-based semantics. We adopt the BCDM (Bitemporal Conceptual Data Model) [Jensen & Snodgrass 96]. We associate a set of time points with tuples. Our approach could also be adapted to cover attribute timestamping, as, e.g., in [Gadia 88], because the distinctions between attribute and tuple timestamping is not one of semantics [Jensen & Snodgrass 96].

To set the stage, we provide some central definitions. Before stating these definitions, it is important to point out three distinctions, which are at the core of our approach.

(i.) *representation versus semantics of the language*—The approach we propose concerns the temporal semantics of data and queries, independent of the representation one uses for time. This distinction is analogous, e.g., to the distinction between concrete and abstract databases emphasized by Chomicki [94].

(ii.) *data language versus query language*—The two should be differentiated [Chomicki & Toman 98]. For instance, both SQL/Temporal [Snodgrass et al. 95b, 98] and SQL/TP [Toman, 96] support time intervals in their data representation language; however, while SQL/Temporal's query language is based on time intervals, SQL/TP's one is based on time points.

(iii.) *data semantics versus query semantics*—In most cases, within the database community, the semantics of *data* is not distinguished from the semantics of the *query*. On the other hand, data have their own semantics, independently of any query language and query operators. This is the usual approach in AI and logics: logical formulæ have an intrinsic meaning, which can be formally defined in model-theoretic terms. Of course, queries are an operational way of making such a semantics explicit. However, a set of logical formulae has a semantics *per se,* even if no query is asked. Analogously, we will say that data in a database have a semantics, that we will call "semantics for data" (or semantics, for short). We note in passing that the conventional meaning of the term "data model" includes both data objects and operations [Tsichritzis & Lochovsky 82]. However, to emphasize that the objects may have a different temporal semantics than the operators, we use the term "data model" to denote only the objects, and "query language" to denote only the operations.

These distinctions will be emphasized and explicated throughout this paper.

2.1 Point-based versus Interval-based Semantics For Data

The fundamental tension examined here is point-based versus interval-based; this characterization may be applied, somewhat orthogonally, to the semantics for data and to the semantics for queries. We first consider data; query semantics will be examined in later sections.

Definition *Point-based semantics for data:* The data in a temporal relation is interpreted as a sequence of states (with each state a conventional relation: set of tuples) indexed by points in time. Each state is independent of every other state. Such temporal relations can be encoded in many different ways (data language). The following are three different encodings of the same information, within a point-based semantics, of John being married to Mary.

Example:

<John, Mary || {1,2,7,8,9}> $\in R$

< John, Mary || {[1-2],[7-9]}}> $\in R$

< John, Mary || [1-2]> $\in R$ and <John, Mary || [7-9]> $\in R$

These tuples encode that the indicated tuple is in the states indexed by the times 1, 2, 7, 8, and 9. The fact denoted by < John, Mary > is in certain states, in this case, 5 individual states, as follows.

1 → {< John, Mary >}

2 → {< John, Mary >}

7 → {< John, Mary >}

8 → {< John, Mary >}

9 → {< John, Mary >}

Definition *Interval-based semantics for data:* Each tuple in a temporal relation is associated with a set of time intervals, which are the temporal extents in which the fact described by the tuple occur. In this semantics the index is a time interval. Time intervals are atomic primitive entities, in the sense that they cannot be decomposed. Note, however, that time intervals can overlap; there is no total order on time intervals, unlike time points.

Example:

Let <John || {[10-20]}> represent the fact that John started to build a house at 10 and finished at 20. If an interval-based semantics is adopted, the interval [10-20] is interpreted as an atomic (indivisible) one.

[10,20] → {<John>}

This tuple does *not* imply that John built the house in [12-15], or at the time point 12, or at any other time interval different than [10-20].[1]

2.2 Related Terminology

Here we briefly relate the above central notions of point-based and interval-based semantics to various other terms that have been prominent in the literature.

- *Snapshot reducibility:* two relations are *snapshot reducible* if their respective snapshots at all points in time are identical [Jensen & Snodgrass 96]. This notion is applicable only to a point-based semantics, as the snapshot of a relation in an interval-based semantics is not relevant. (One could envision the snapshot at a point in time of interval-based data, but to do so requires coercing, either explicitly or implicitly, that data into point-based data. We discuss coercions in Section 8.)
- *Semantic mapping:* A (concrete) database DB_1 encodes an (abstract) database DB_2 if for every tuple in DB_2 at every time instant implies a tuple in DB_1 timestamped with an interval that includes that time instant [Chomicki and Toman 98]. Such a mapping implies that the data in the interval-stamped DB_1 is point-based.
- *Snapshot equivalence:* a temporal operator is *snapshot equivalent* to a nontemporal operator if all snapshots of the result of the temporal operator are identical to the nontemporal operator applied to the snapshot of the argument at the same time [Snodgrass 87]. As before, this notion is applicable only to a point-based semantics. An analogous definition (i.e., *weak equality*) was previously defined by Gadia [88].
- *Sequenced:* a temporal query is *sequenced* with respect to a similar nontemporal query (say, if the temporal query contains an additional keyword [Böhlen 00]) if the semantics of the temporal query is expressed as the semantics

---

[1] Again, also our definition of interval-based semantics for data is independent of the representation formalism. For instance, one could choose to represent time intervals as a set of points, and nevertheless adopt the interval-based semantics. If an interval-based semantics is adopted, <John || {{10,11,12,13,14,15,16,17,18,19,20}}> denotes exactly the same content as <John || {[10-20]}> above.

of the nontemporal query on each state of the underlying relation(s) [Snodgrass et al. 98]. Again, this notion applies only in a point-based semantics.

- *Current*: a temporal query is *current* if it accesses only the current state [Snodgrass et al. 98], which immediately implies that it is point-based.
- *Non-sequenced:* a temporal query is *non-sequenced* if the result is computed from data at potentially multiple points in time [Snodgrass et al. 98]. In contrast with the previous notions, non-sequenced applies equally to point-based and interval-based semantics. An example in the former case is a trend query requesting those salaries that have increased over the previous year. An example of the latter is a query on duration requesting those employees who have been with the company for at least five years. (In fact, all interval queries within the TSQL benchmark [Jensen et al. 93] are necessarily interval-based notions, as will become clear as we get further into our presentation.)
- *point-based query:* We delay discussion of this term, used often in TDBs [Toman 96, Böhlen et al. 98] until Section 12.
- *Upward* and *downward hereditary*: a fact f is *upward hereditary* if and only if when f holds on two overlapping intervals i_1 and i_2, then it holds on the interval $i_1 \cup i_2$; a fact f is *downward hereditary* if and only if when f holds on an interval i then f also holds in each sub-interval of i. For example, owning a house is both downward and upward hereditary, while building a house is not (see the discussion in Section 3). Both upward and downward hereditary hold if one adopts a point-based semantics for data, while the two properties do not hold if one adopts the interval-based one. It is important to notice, however, that upward and downward hereditary are semantic properties that are nevertheless related to the representation language one adopts. More emphatically, these properties are meaningless in case the data language does not support time intervals. These properties are important, since they underlie two commonly done operations on TDBs: *temporal coalescing* [Böhlen et al. 96], which can be performed only in case upward hereditary holds, and *temporal restriction* [Gadia 88], which is meaningful only in case downward hereditary holds.

We now return to the fundamental distinction that is the focus of this paper, that of point-based versus interval-based semantics of data.

3 Important Dichotomies

In this section, we sketch linguistic arguments of the inadequacy of point-based semantics to capture important aspects of natural language statements, structuring the presentation into four steps. First, in Section 3.1 we sketch a basic issue underlying many approaches in philosophy and linguistics, namely 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 [2]. In particular, the distinction between *telic* and *atelic* facts, which is the kernel of this paper, dates back to Aristotle [Aristotle], has strong cognitive evidence [Bloom et al. 80], and is at the basis of almost all approaches to the semantics of natural language sentences, starting from Vendler’s seminal work [67]. In Section 3.2,

---

[2] Henceforth, we use "fact" as a cover term to denote situations of the different classes (e.g., *states, activities, accomplishments* and *achievements* in [Vendler 67]). Our "facts" correspond, e.g., to *"situations"* in [Mourelatos 78].

we briefly present a seminal and basic analysis by Dowty [79, 86], demonstrating that the semantics of the association of facts to time depends on the classes of facts being considered. Section 3.3 sketches an argument argued effectively within the linguistic community (cf., e.g., [Bennet and Partee 72, Dowty 79, 86, Tichy 85]), namely that point-based semantics is not adequate to deal with the semantics of telic facts (while it works fine for atelic facts), for which an interval-based semantics is needed (i.e., a semantics which evaluates the truth of facts over time intervals [Dowty 86]). Section 3.4 emphasizes that, in accordance with the linguistic and philosophical analysis (cf., e.g., [Verkuyl 71, Bach 86, Moens & Steedman 88] and the collection in [CL 88]), the distinction between telic and atelic facts is not a rigid one: basically telic facts can be transformed into (or decomposed into, or viewed as) atelic ones, and vice versa.

The first three points motivate (in our approach, as well as in many approaches in philosophy, linguistics and AI) the adoption of an interval-based semantics to deal with telic facts. The last point motivates a flexible approach in which the association of a point-based versus interval-based semantics to facts is not a rigid one.

Although the distinction between different types of facts according to their temporal properties dates back to the early approaches to philosophy, for the sake of brevity and clarity we choose to discuss this distinction (and the distinction between a point-based and an interval-based semantics) mainly on the basis of the recent approaches in linguistics and computational linguistics. In fact, we believe that paying attention to the suggestions of the linguistic analysis can be important and fruitful also in designing data models and query languages.

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” [88, 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 information/knowledge. We thus believe that making the semantics of (temporal) databases closer to the semantics of natural language in a further step towards the user-friendliness of (temporal) databases (cf., e.g., the discussions in [Moens 88, Clifford 88]). Moreover, such a “natural-language-based” organization could also simplify the attempt of building natural language interfaces to (temporal) databases (cf., e.g., [Copestake & Spark Jones 90, Androutsopoulos et al. 95]).

3.1 Different Classes of Facts

In the areas of linguistics and computational linguistics, it is commonly agreed that natural language sentences can be classified within different *aktionsart classes* (e.g., *activities, accomplishments, achievements* and *states* in [Vendler 67]; also termed *aspectual classes* [CL 88]) depending on their linguistic behavior 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 building a house” (*accomplishment*) is correct. Many approaches in the linguistic literature devised sets of linguistics tests and criteria to distinguish among different classes of facts, and to classify input sentences along such distinctions (cf., e.g., the collection in [CL 88]). However, many linguistic approaches also took into account *semantic criteria*, which are centered on the association of facts with time, which is the core issue of this paper.

3.2 Associating Facts with Time

Dowty [86] introduced linguistic tests to state the impact of the “aktionsart” on the semantics of linguistic propositions. Consider, e.g., the implication from the progressive aspect of verbs: “if ϕ is an activity verb [atelic verb in our terminology], then *X is now ϕing* entails that *X has ϕed*. If ϕ is an accomplishment verb [telic verb in our terminology] then *X is now ϕing* entails that *X has not yet ϕed*" [Dowty 79, page 59]. For instance, John is walking implies that John has walked, while John is building a house does not imply that John has built a house. According to this and many other tests, Dowty proposed the following semantic criteria to distinguish between *states, activities* 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 a stative)

(b) A sentence ϕ is an activity (or *energeia*) iff it follows from the truth of ϕ at an interval I that ϕ is true at all subintervals of I down to a certain limit in size (e.g., if John walked from 1:00 until 2:00 PM, then most subintervals of this time are times at which John walked; *walk* is an activity).

(c) A sentence ϕ is an accomplishment/achievement (or *kinesis*) iff it follows from the truth of ϕ at an interval I that ϕ is false at all subintervals of I (e.g., if John built a house in exactly the interval from September 1 until June 1, then it is false that he built a house in any subinterval of this interval: *build a house* is an accomplishment/achievement)" [Dowty 86, page 42].

The property (a) for states has been often termed *downward hereditary* in the AI literature (cf., e.g., [Allen 84, Shoham 87, Bettini et al. 98]). 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 linguistic researchers also considered other distinctions between sentences that are less relevant for this paper, and that will be discussed in Appendix 2.

3.3 Inadequacy of Point-Based Semantics

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. As already mentioned in Section 2, point-based semantics associates states (conventional relations) with time points; equivalently, this semantics associates the truth of facts with time points. This semantics works perfectly on stative facts: "John was asleep" in item (a) above holds exactly in all time points (equivalently, exists in the states indexed by all the time points) within 1:00 and 2:00 PM.

On the other hand, a point-based semantics is inadequate when dealing with accomplishments. For instance, given item (c) above, there is no specific time point *p* such that "John built a house" is true in *p*. The building of the house was achieved exactly in the *time interval* from September 1 to June 1 (and in no sub-interval or sub-point of it). This and analogous observations led most linguistic researchers, starting from pioneering papers [Bennet & Partee 72, Dowty 79] in the 1970's, to criticize the point-based semantics and to adopt an interval-based semantics to deal with the meaning of natural language sentences. We recall from Section 2 that facts in an interval-based semantics are associated with the time intervals in which they occurred, and time intervals are primitive non-decomposable entities.

3.4 Relationships Between Telic and Atelic Facts

Another important issue, which has a deep impact on the semantic model one has to adopt, and which has been pointed out by almost all the recent research about aktionsart classes, is the fact that aktionsart classifications are not rigid. They concern sentences *in neutral form* [Verkuyl 71] (e.g., in the present perfect), and it is possible (using, e.g., appropriate linguistic tools such as, e.g., progressive forms) to have different aktionsart views for the same fact (cf., e.g., [Vendler 67, Mourelatos 86, Dowty 86, Bach 86, CL 88, Moens & Steedman 88]). For instance, let us consider again the fact in item (c) above: "John built a house from September 1 to June 1" . Such a sentence describes an accomplishment, since downward hereditary does not hold. This means that, e.g., the answer to the query "*Did* John *build* a house on March 1st 1999 at 12:00?" is *no*. However, one can look an accomplishment from the inside, looking at the "pieces of activity" which compose it. A standard linguistic tool for obtaining this inner view is the use of the progressive: given (c), one can correctly say that "John *was building* a house on March 1st 1999 at 12:00" , or, in other words, the answer to the query "*Was* John *building* a house on March 1st 1999 at 12:00?" is *yes*.

Different authors used different terminologies and models to deal with this phenomenon. For instance, Moens and Steedman [88] based their explanation on the fact that accomplishments are *telic* (from the Greek: "telos" meaning "goal") in the sense that they are characterized by the fact that they reach a *culmination* (or, in other words, they are activities with an intrinsic goal, or telos -finishing the construction of the house in item (c)), while activities (and states) are *atelic* (from the Greek: 'a' as a prefix indicates negation), i.e., do not have an intrinsic culmination (consider, e.g., walking, looking around, owning a house, earning a given salary, etc.). These forms of transformations (termed *aktionsart coercions* in the following) have been graphically represented in Figure 1. In Figure 1 and in the following, atelic facts stand for Vendler's *states* and *activities*, and telic facts for Vendler's *accomplishments*. Given an accomplishment, one can look within the activities which lead to its culmination (thus, stripping out the culmination; arc -*culmination* in the figure), as in the query "*Was* John *building* a house on March 1st 1999 at 12:00?" .Analogously, one can easily associate a culmination to an activity (arc +*culmination* in the figure), viewing it as an accomplishment. For instance, "*John walked"* is not telic, but "*John walked from home to the church"* is telic [Verkuyl 71].

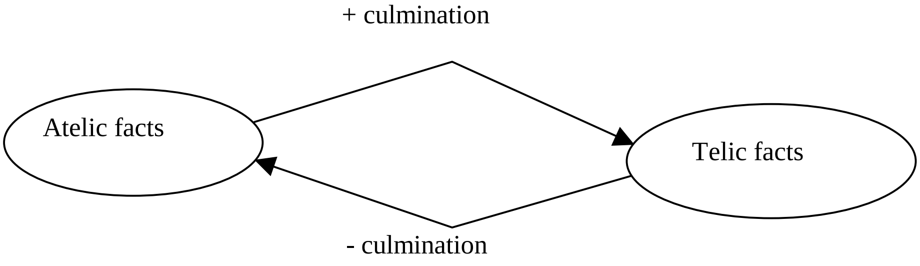

Figure 1. Akionsart coercions.

4 Atelic Temporal Model

Here, we first introduce a paradigmatic example of an approach based on point-based (snapshot) semantics (and on tuple timestamping), which is a simplification and an adaptation of SQL/Temporal's model (i.e., the BCDM). However, this choice is not critical, as most (if not all) extant *temporal models* are point-based in their data, though we hasten to add that many temporal *query languages* are a mixed of point-based and interval-based (atelic and telic). Our purpose here is to bring this central distinction to the fore, and rendering it explicit in both the data model and in the query language.

4.1 The Model

We assume that time is a linear, ordered and discrete set of time points $p \in \mathcal{P}$ (e.g., isomorphic to integers) [Jensen & Dyreson 98]. 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 in most TDB approaches (see below) we adopt a point-based semantics for data. Thus, we associate with each tuple an *atelic element*, which is the set of time points[3] representing the collected times when the fact holds.

Concerning the data model, in this paper we only deal with *valid time state relations* [Jensen & Dyreson 98]. For the sake of simplicity, we disallow *value-equivalent* tuples (tuples with mutually identical atemporal attributes), as in TSQL2 and BCDM. Disallowing value-equivalent tuples makes the model clearer, since the full story of a fact is contained in a single tuple. From a technical point of view, this simplification allows us to assume a set framework for the semantics.

We term these relations *atelic relations*, and term tuples in such relations *atelic tuples*, since they represent atelic facts. For instance, the STOCK^A relation below is an atelic relation representing stocks (we suppose that the company name is the key), their category, and their price over time (here and in the following tables we give the time in minutes for a specific hour). For example, the first tuple represents the fact that IBM had a price of $63 for four minutes (from 14 to 17).

STOCK^A

Name	**Category**	**Price**	VT
IBM	High-tech	63	{14,15,16,17}
IBM	High-tech	61	{12,13}
GM	Industrial	49	{10,11,12,16,17,18}

4.2 Impact of Aktionsart Distinctions

For the sake of clarity and brevity, in this paper we do not take into account all the aktionsart distinctions identified within the linguistic literature. In particular, we do not distinguish between states and activities, since both of them covers atelic facts (for further discussion of aktionsart distinctions, see Appendix 2). On the other hand, we believe that

---

[3] In BCDM [Jensen & Snodgrass 96], time is seen as a finite sequence of chronons, isomorphic to a sequence of natural numbers. The sequence of chronons can be thought of as representing a partitioning of the time line into indivisible segments. Using chronons instead of discrete time points does not imply a significant change in the approach described in our paper.

the distinction between durative *telic* facts (i.e., *accomplishments*, in Dowty's terms [79]) and durative *atelic* facts (i.e., *states* and *activities*) has a dramatic impact on the semantics of (relational) TDBs. The treatment of telic facts in a temporal relational algebra based on atelic elements (i.e., on a point-based semantics for data; see Section 2) encounters several problems. We illustrate these problems with an example, but we stress that the same problem arises whenever an atelic relation (i.e., a relation based on point-based semantics) is used to represent telic facts.

A phone call starting at time t_1 and ending at time t_2 is a durative telic fact. For instance, if John made a call to Mary from time 10 to time 12, he didn't make it from 10 to 11. Similarly, two consecutive calls, one from 10 to 12 (inclusive) and the other from 13 to 15 (inclusive), are clearly different from a single call from 10 to 15. 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 John to Mary starting at 10 and ending at 15 is cheaper that two consecutive calls taking the same amount of time. This implies that phone calls are telic facts.

Suppose that we use an *atelic* relation to represent telic facts such as phone calls. In particular, let us model the fact that John called Mary twice: once from 10 to 12 and once from 13 to 15 (all extremes are included). The linguistic analysis described in Section 3 tells us that telic facts cannot be correctly captured using models that are based on a point-based semantics for data, since such models are not sufficiently expressive. Consider the following atelic PHONE^A relation.

PHONE^A

Caller	**Called**	**VT**
John	Mary	{10,11,12,13,14,15}

The point-based semantics of this relation is given in Figure 2.

$$10 \rightarrow \{<\text{John, Mary}>\}$$
$$11 \rightarrow \{<\text{John, Mary}>\}$$
$$12 \rightarrow \{<\text{John, Mary}>\}$$
$$13 \rightarrow \{<\text{John, Mary}>\}$$
$$14 \rightarrow \{<\text{John, Mary}>\}$$
$$15 \rightarrow \{<\text{John, Mary}>\}$$

Figure 2. Point-based semantics for the table PHONE^A.

This atelic PHONE^A relation captures the fact that John was calling Mary at 10, 11, 12, 13, 14, and 15. This relation does not capture relevant information, namely, the fact that there were two distinct calls, one from 10 to 12 and the other from 13 to 15.

4.3 Representation Language versus Data Semantics

It is important to emphasize that the above loss of information is *not* due to the fact that the *language* we use to represent data does not support time intervals, but to the fact that the underlying *data semantics* is point-based. In other words, whenever the data semantics is point-based (independently of how the data language looks like), we risk a loss of information when dealing with telic facts. This is true for all approaches that use temporal elements to timestamp tuples; consider, SQL/Temporal, TSQL2, TSQL, HQL, and TQuel, which utilize temporal elements to timestamp tuples, and Gadia's [88] Homogeneous Relational Model, which uses temporal elements to timestamp attributes.

In fact, while temporal elements are sets of intervals, this is only a matter of data *representation language*, since the underlying data *semantics* is point-based. In other words, temporal elements are merely a notational representation for a set of time points. In such approaches, one can represent the above phone example as in relation PHONE2^A.

PHONE2^A

Caller	Called	VT
John	Mary	{[10-12],[13-15]}

However, while PHONE2^A is different from PHONE^A at the level of the representation language, it is identical to PHONE^A at the data semantics level, since both relations represent the identical content of that in Figure 2, namely, that John was calling Mary at 10, 11, 12, 13, 14, and 15 (formally, we may say that PHONE^A and PHONE2^A are *snapshot equivalent* [Jensen & Snodgrass 96]). Thus, the loss of information is the *same* even if we adopt *temporal elements* [Gadia, 88] instead of atelic elements.

To summarize, one central message of this paper is that, independently of whether the data *language* supports or not time intervals, if the underlying data *semantics* is point-based, we do not have sufficient expressive power to properly deal with the semantics of telic facts. Obviously, this lack of expressive power has a dramatic impact also on the results one may obtain when querying relations based on the point-based semantics (see Sections 6.3 and 8.2). To discuss this concretely, we now briefly present an atelic (point-based) algebra, show its inadequacies when querying telic facts represented as atelic data, and then extend this data model and algebra to treat telic facts properly.

5 Atelic Relational Algebra

We use a slight adaptation of the algebra of SQL/Temporal to exemplify atelic operators. These operators are designed to generalize the standard set operators to apply over atelic data. We also note that the definitions of some operators (e.g., intersection and temporal restriction) are similar to Gadia's *computational semantics* definitions [86, 88], adapted to the case of tuple timestamping.

First, we generalize the five operators, selection, projection, Cartesian product, difference and union, which extend the standard snapshot algebraic operators (i.e., σ, π, $\times$, -, $\cup$) to operate on atelic relations. We indicate by σ, π^A, $\times^A$ and $-^A$ the operators on atelic relations. Each takes one or more atelic relations as input, and produces an atelic relation as the result. This result is defined in terms of the analogous conventional operators. For example, each state of the result, at a particular time *t*, of the projection operator is defined as the conventional projection operator applied to the underlying state at time *t*. To effect this result, we define below how to compute the timestamp (as a atelic element) of the result of each of the operators. We denote with $Sch(R^A)$ the data attributes of an atelic relation R^A. Given a tuple *t* of a relation *R*, we denote by $t(R)$ the value of the data attributes in *t*, and by $t(\mathrm{VT})$ its validity time.

Selection. The atelic selection selects the tuples whose data part satisfies a condition *P* (which is a condition on the data part only), regardless of its temporal part. Hence the conventional (nontemporal) selection operator suffices for atelic data.

Cartesian product. The state-by-state interpretation above for atelic operators dictates that we adopt the intersection semantics for the atelic Cartesian product: the validity time of the resulting relation is intersection of the validity times of the tuples.

$$
\begin{aligned}
&Sch(R^A{}_1 \times^A R^A{}_2) \equiv Sch(R^A{}_1) \cup Sch(R^A{}_2)\\
&R^A{}_1 \times^A R^A{}_2 \equiv \{s \mid \exists t_1 \in R^A{}_1\ \exists t_2 \in R^A{}_2\ (s(R^A{}_1) = t_1(R^A{}_1) \wedge s(R^A{}_2) = t_2(R^A{}_2) \wedge s(\mathrm{VT}) = t_1(\mathrm{VT}) \cap t_2(\mathrm{VT})\ )\\
&\qquad\qquad\qquad \wedge s(\mathrm{VT}) \neq \emptyset\}
\end{aligned}
$$

Union. The conventional union relational operator ($\cup$) works fine for atelic tuples.

The intersection and union operators above are simply set intersection (over atelic elements, which are sets of time points) and set union (over sets of attributes).

Projection. Projection can be used to select data attributes (let *B* be the set of such attributes). For value-equivalent tuples, the union of the validity times must be performed, as in the case of atelic union above. This is a little more complex, because possibly many tuples are involved in the union.

$$
\begin{aligned}
&Sch(\pi^A{}_B(R^A)) \equiv B\ (B \subseteq Sch(R^A))\\
&\pi^A{}_B(R^A) \equiv \{s \mid \exists t_1 \in R^A\ (s(B) = t_1(B) \wedge (\forall t_2 \in R^A\ (t_2(\mathrm{B}) = s(B) \Rightarrow t_2(\mathrm{VT}) \subseteq s(\mathrm{VT})\ )\ )\\
&\qquad\qquad \wedge (\forall p \in s(\mathrm{VT})\ (\exists t_2 \in R^A\ t_2(B) = s(B) \wedge p \in t_2(\mathrm{VT})\ )\ )\ \ \}
\end{aligned}
$$

Difference. The difference operator between two atelic relations R_1 and R_2 gives as result a relation containing all the tuples of R_1 that are different in the data part from all tuples in R_2. Tuples with the same data part are dealt with considering the difference of their validity times.

$$
\begin{aligned}
&Sch(R_1{}^A -^A R_2{}^A) \equiv Sch(R_1{}^A)\ \ (Sch(R_1{}^A) = Sch(R_2{}^A))\\
&R_1{}^A -^A R_2{}^A \equiv \{s \mid (\ (\exists t_1 \in R_1{}^A\ (s(R_1{}^A) = t_1(R1^A) \wedge \neg\exists t_2 \in R_2{}^A\ t_1(R_1{}^A) = t_2(R_2{}^A)\ )\\
&\qquad\qquad \vee (\exists t_1 \in R_1{}^A\ \exists t_2 \in R_2{}^A\ (\ t_1(R_1{}^A) = t_2(R_2{}^A)\ \wedge\ s(R_1{}^A) = t_1(R_1{}^A) \wedge s(\mathrm{VT}) = t_1(\mathrm{VT}) - t_2(\mathrm{VT})\ )\ )\ )\\
&\qquad\qquad \wedge s(\mathrm{VT}) \neq \emptyset\ \}
\end{aligned}
$$

We now define an additional atelic operator, an adaptation of Gadia's temporal selection operator [88] (henceforth termed temporal restriction) to tuple timestamping.

Temporal Restriction. This operator restricts the validity time of the tuple to the input time interval I.

$$Sch(Restr^{A}_{[I]}(R^{A})) \equiv Sch(R^{A})$$

$$Restr^{A}_{[I]}(R^{A}) \equiv \{s \mid (\exists t \in R^{A} \ (s(R^{A}) = t(R^{A}) \ \wedge s(\mathrm{VT}) = (\mathrm{VT}) \cap \{I\}) \wedge s(\mathrm{VT}) \neq \emptyset \ \}$$

Note that the notation $\{I\}$ indicates the set of points contained in the interval I.

Temporal operators such as temporal selection that have explicit predicates that treat the timestamps as intervals are *not* included within our atelic algebra, since they essentially have an interval-based semantics. Consider, for instance, temporal selection based on duration (e.g., "select all phone calls that lasted at least 5 units"). In order to determine the duration of a fact in an atelic model, one first has to collect all the consecutive time points in which the fact holds in order to determine the maximal convex time interval(s) covering such points. Then, the durations of such intervals are considered. Thus, there is an (implicit) shift from time points to time intervals. In our approach, we make such a shift explicit, by supporting temporal selection operators only within the telic algebra (which operates on a data model based on interval-based semantics; see Sections 6 and 7).

Given this atelic model and algebra, and given the earlier discussion (such as in the phone call example) that showed that atelic/telic distinction is an important one, we now present a telic model and algebra.

6 Telic Relational Model

We first introduce a new model to support telic facts; the following section provides an algebra consistent with this model.

6.1 Telic Elements

In order to cope with the temporal issues discussed in Section 3, we need to introduce explicitly the notion of *time intervals*. A time interval i is a convex set of time points between a starting point i^{-} and an ending point i^{+}, that is,

$$\forall p \ p \in \mathcal{P} \ (i^{-} \leq p \leq i^{+} \Rightarrow p \in I \ ), \text{ where } i^{-} \leq i^{+}.$$

We indicate the domain of time intervals by $\mathcal{I}$. Moreover, we consider sets of time intervals (i.e., subsets of 2^{I}), termed *telic elements*. It is important to notice that, although we define time intervals as sets of time points, we regard them as *atomic* objects. Thus, a telic element may also contain meeting or overlapping intervals (for instance, {[10-15], [13-20]} is a perfectly reasonable telic element).

Two functions map atelic elements to telic elements and vice versa. *to-atelic* takes in input a telic element *TE* (i.e., a set of time intervals) and gives as output the atelic elements containing all the points belonging to the intervals in *TE*. *to-telic*, in contrast, takes as input an atelic element (a set of time points) and gives as output the telic elements containing the maximal convex intervals that cover exactly the time points in the atelic element.

to-atelic({[12-16], [15-17]}) = {12, 13, 14, 15, 16, 17}

to-telic({12, 13, 14, 15, 16, 17, 20, 21}) = { [12-17], [20-21] } Note that, given any atelic element $A \in 2^P$, *to-atelic*(*to-telic*(A)) = A. However, given a telic element $T \in 2^I$, it may be the case that *to-telic*(*to-atelic*(T)) $\neq T$. For example, if T = { [10,14], [12,16] }, then *to-atelic*(T) = {10,11,12,13,14,15,16}, so *to-telic*(*to-atelic*(T)) = {[10-16]} $\neq T$.

We regard time intervals as indivisible primitive entities, and we use the operations of union, complement and intersection over telic elements as a *restriction of standard set operators on the domain of time intervals*. For example,

{ [10-15], [20-30] } ∪ { [10-25], [35-40] } = { [10-15], [20-30], [10-25], [35-40] }

{ [10-15], [20-30] } ∩ { [10-25], [20-30], [35-40] } = { [20-30] }.

Notice that temporal coercion is not performed over time intervals by union, so that upward hereditary is not forced.

6.2 Telic Model

As discussed in Section 4, associating a tuple *t* with an *atelic* element $\{p_1, .., p_k\}$ means that the fact represented by the tuple *t holds* over all time points $p_1, .., p_k$, thus utilizing the point-based semantics. On the other hand, time intervals (and *telic* elements) are introduced in order to represent that the fact described by a tuple *t* is *accomplished* in a given time interval *i*, i.e., *t* starts at i^- and finishes (reaches its culmination, or telos) at i^+. For instance, in the phone call example, and considering the time interval [10-12], a tuple <John, Mary, { [10-12] }> means that John's call to Mary started at time 10 and finished at 12. Notice that, although it is true that John *was calling* Mary within any time point contained in [10-12], it would not be correct to state that John *accomplished* such a call at 11; downward hereditary does not hold. Analogously, upward hereditary does not hold when a telic interval is associated with a tuple. For instance, the tuple <John, Mary, { [10-12], [13-15] } represents two different episodes of John calling Mary, not to be confused or "merged" together.

In our telic relational model, a *telic relation* is a set of *telic tuples*, each with a validity time, which is a *telic element*. For the sake of simplicity we do not admit value-equivalent tuples, similarly to the atelic model.

6.3 Impact of Aktionsart on Telic Relations

We now show that adopting an interval-based semantics (i.e., associating tuples with telic elements), one can correctly capture the meaning of telic facts into relational relations. Let us consider again, e.g., the phone example discussed in Section 4.2. Instead of using an atelic relation (cf., the PHONE^A relation), let us now use a telic relation (say PHONE^T) to represent the fact that John called Mary from 10 to 12, and then from 13 to 15.

PHONE^T

Caller	**Called**	**VT**
John	Mary	{ [10-12], [13-15] }

PHONE^T is a telic relation, so that the validity time is a telic element, and interval-based semantics is used. This means that the time intervals [10-12] and [13-15] are interpreted as atomic temporal entities, and the semantics of PHONE^T can be expressed as in Figure 3 (contrast with Figure 2).

[10-12] → {<John, Mary>}

$[13\text{-}15] \rightarrow \{<\text{John}, \text{Mary}>\}$

Figure 3. Interval-based semantics for the relation PHONE^T.

Thus, the telic relation PHONE^T correctly models the information that there were two episodes of John calling Mary, one that occurred from 10 to 12, and the other from 13 to 15. [4]

More generally, telic elements (and interval-based semantics) are sufficiently expressive to model the semantics of telic facts, since they do not lose information concerning the different episodes, even in case such episodes overlap or meet in time. This fact is also evident when asking queries to telic relations: in Section 7.2, we will show that queries involving upward and downward hereditary properties yield the expected answers when telic facts are modeled using telic relations (and, thus, adopting an interval-based semantics for data).

7 Telic Algebra

As before, we first propose a specific set of operators, then we present some examples to illustrate how these operators can be used to effect downward and upward hereditary.

7.1 Algebraic operators

Now, we can define the operators on telic relations. The rationale underlying all the definitions is the following. In the interval semantics, each tuple occurs exactly in the time intervals in its validity time, and nowhere else. Notice that, in the definitions of the telic algebraic operators, set operators apply to telic elements (i.e., set of time intervals), while in the atelic algebra they operate on atelic elements (i.e., set of points).

Some of the telic operators have an identical definition as their atelic counterpart. In particular, (nontemporal) selection, which we showed in Section 5 applies to atelic relations, also works fine on telic relations. The formal definition of telic union ($\cup^T$) is similar to the definition of atelic union ($\cup$), with the added behavior that the telic union of two value-equivalent telic tuples t_1 and t_2 is a single tuple with a valid timestamp of the set union ($\cup$) of the two underlying telic elements t_1(VT) and t_2(VT). This similarity also applies to atelic ($\pi^A{}_B$) and telic ($\pi^T{}_B$) projection operators.

Cartesian product. Atelic Cartesian product necessarily requires that the two timestamps overlap (equivalently, that the two sets of time points are not disjoint), for each pair of tuples, effecting a point-based interpretation of facts. From a theoretical view, we could define telic Cartesian product similarly, but this would retain intervals only if they matched exactly, which seems artificial. The atelic Cartesian product can be seen as the counterpart of the “while” adverb in the temporal algebra, and “while” involve an atelic view of the facts to which it applies. Since forcing intersection is unnatural, we instead allow any of the Allen [83] predicates, and to make this predicate explicit in the operator, which can then be seen as the counterpart of the “before” or “meets,” etc. adverbs.

---

[4] Notice that, in some cases, one may need also to deal with telic facts that occur at overlapping time intervals.

There is also the issue of what timestamp to associate with the resulting tuples. For atelic Cartesian product, this decision necessarily is to intersect the two timestamps to determine the resulting timestamp, which doesn't work here. So we simply chose to return the timestamp of the left tuple. In the following, ϕ is a binary Allen predicate.

$$Sch(R^A{}_1 \times_\phi{}^A\ R^A{}_2) \equiv Sch(R^A{}_1\,) \cup Sch(R^A{}_2)$$

$$R^T{}_1 \times_\phi{}^T R^T{}_2 \equiv \{s \mid \exists t_1 \in R^T{}_1\ \exists t_2 \in R^T{}_2\ (\ s(R^T{}_1) = t_1(R^T{}_1) \wedge s(R^T{}_2) = t_2(R^T{}_2) \wedge\ \phi(\ t_1(\mathrm{VT}),\ t_2(\mathrm{VT})) \wedge\ s(\mathrm{VT}) = t_1(\mathrm{VT})\}$$

Telic difference has a similar difficulty: performing set difference between telic elements is artificial, since only repeated intervals will be affected. For this reason, we do not include a telic difference operator. Temporal restriction on telic relations also does not make sense, because the entire interval must be retained.

Temporal Selection. Unlike conventional selection, temporal selection takes a temporal predicate, ϕ, on telic intervals; examples of such predicates include duration and comparison with constants (see Section 5).

$$Sch(\sigma^T{}_\phi(R^T)) \equiv Sch(R^T)$$

$$\sigma^T{}_\phi(R^T) \equiv \{s \mid \exists t \in R^T\ (\ s(R^T) = t(R^T) \wedge s(\mathrm{VT}) = \mathrm{VT}') \wedge s(\mathrm{VT}) \neq \varnothing\}, \text{ where VT}' = \{i \in t(\mathrm{VT}) \mid \phi(i)\}.$$

7.2 Impact of Aktionsart Distinctions

In the following, we again consider the phone example. Suppose that we want to store the following data.

John made two calls to Mary, one from 10 to 12, and the other from 13 to 15;
Sue made two calls to Ann, one from 12 to 14, the other from 15 to 16;
Eric made a call to Paul from 14 to 16.

Figure 4. Sample Temporal Data.

We use a telic relation PHONE^T to represent such data, and also consider the corresponding atelic relation PHONE^A. Once again, notice that the distinction between telic and atelic relations is not due to the representation *language*, but to the (interval-based vs. point-based) *data semantics*. For instance, using temporal elements, one would have a relation which, from the syntactic point of view, looks like PHONE^T. However, this would be only a matter of notation, while the underlying content would be that shown in relation PHONE^A (in other words, as long as one adopts point-based semantics for data, and independently of how the representation formalism looks like, one will have relations that are *snapshot equivalent* to relation PHONE^A below).

PHONE^T

Caller	Called	VT
John	Mary	{ [10-12], [13-15] }
Sue	Ann	{ [12-14], [15-16] }
Eric	Paul	{ [14-16] }

PHONE^A

Caller	Called	VT
John	Mary	{10,11,12,13,14,15}
Sue	Ann	{12,13,14,15,16}
Eric	Paul	{14,15,16}

Now, we consider some prototypical queries for answering which a correct treatment of upward and downward hereditary plays a fundamental role. We will apply such queries to both PHONE^{A} and PHONE^{T}, showing that the case in which the atelic relation is used to model telic facts (such as phone calls) provides undesired results.

(1) Downward hereditary.

(Q1) $\mathit{Restr}^{A}_{\mathbf{[10,11]}}$ **(**PHONE^{A}**)**

The answer to Q1 is the tuple <John, Mary || {10,11}>. However, the restriction operator cannot be applied to a telic relation. Basically, since downward hereditary do not hold for telic facts, from the fact that John made a *complete* phone call to Mary from 10 to 12, it is wrong to conclude that he makes such a *complete* call from 10 to 11 (at most, one could conclude that John was calling Mary at 10 and 11, but not that he did a *complete* call at that time).

(2) Upward hereditary.

The treatment of upward hereditary causes even more severe problems to the point-based semantics. To illustrate, we provisionally apply temporal selection also to atelic relations, even if, in our approach, temporal selection only applies to telic ones. Notice that, actually, temporal selection is used in most temporal algebræ in the literature, that on the other hand cope only with point-based semantics data models. In Section 8, we will propose a clean solution to this problem, by introducing coercion functions.

Let us consider, for instance, the following query, asking for all information regarding phone calls lasting at least 5 units in the atelic relation PHONE^{A}.

(Q2) $\sigma^{T}_{\mathrm{duration(VT)} \geq 5}$ **(**PHONE^{A}**)**

The tuples <John, Mary || {10,11,12,13,14,15}> and <Sue, Ann, {12,13,14,15,16}> are given as output. This is due to the fact that, in the atelic relation, we have a loss of information concerning the start and end points of calls, since consecutive calls are "coalesced together". On the other hand, none of the calls in Figure 4 lasts at least 5 units.

Let us suppose now that the above query is applied to the telic relation PHONE^{T}.

(Q2′) $\sigma^{T}_{\mathrm{duration(VT)} \geq 5}$ **(**PHONE^{T}**)**

In such a case, no tuple is given as output, as desired (i.e., consistently with the data in Figure 4).

We have an analogous situation if we ask for information concerning phone calls that, e.g., follows one call from John to Mary.

(Q3) PHONE$^{A} \times^{T}_{\mathrm{After}}$ ($\sigma_{\mathrm{caller=John,called=Mary}}$ (PHONE^{A}))

(Q3′) PHONE$^{T} \times^{T}_{\mathrm{After}}$ ($\sigma_{\mathrm{caller=John,called=Mary}}$ (PHONE^{T}))

The answer to (Q3) is the empty relation, while the answer to (Q3′**) is the relation containing the tuples <Sue, Ann ||** {[15-16]}> and <Eric, Paul || {[14-16]}>, as desired.

In general, considering the benchmark for temporal query languages discussed in [Jensen et al. 93], we see that analogous problems arise for all *interval queries* (i.e., queries asking about durations, endpoints and relative positions of the validity times of tuples in the temporal benchmark) whenever relations using point-based semantics for data are used in order to represent some telic fact.

8 An Integrated, Three-sorted Algebra

In Section 6.3, we showed that our temporal model and algebra cope with tuples representing telic facts (accomplishments). In this section we show several examples motivating the fact that both telic models and operators and atelic ones are needed, as well as a flexible way of coercing telic relations to atelic ones, and vice versa (thus paralleling the coercion functions which occur in natual languages; see Section 3 and Appendix 2).

8.1 The Telic Model and Algebra Are Not Adequate for Atelic Facts

Unfortunately, the telic model and algebra in Sections 6 and 7, taken in isolation, are not powerful enough to deal with all types of facts, in particular, atelic facts. Actually, using the telic model and algebra to deal with atelic facts such as earning a given salary, owning a house, and so on, generate exactly the dual of the problems discussed in Sections 4.2 and 7.2. Both downward and upward hereditary properties hold for atelic facts; not considering them causes loss of information. Consider, for instance, the telic relation STOCK^T, which is the telic "counterpart" of the atelic relation STOCK^A in Section 4.1, using maximal convex intervals.

STOCK^T

Name	**Category**	**Price**	VT
IBM	High-tech	63	{ [14-17] }
IBM	High-tech	61	{ [12-13] }
GM	Industrial	49	{ [10-12], [16-18] }

Notice that it is part of the intended meaning of "stock prices" that stating that IBM price was more than 60 on [12-13], and then on [14-17] implies that it was more than 60 from 12 to 17 (upward hereditary); moreover, from the fact that the price of IBM was 63 from 14 to 17, one may correctly infer that it was 60 from 15 to 16 (downward hereditary). These semantic assumptions are automatically captured if the stock data are represented by an atelic relation (i.e., by a relation based on a point-based semantics for data). On the other hand, such assumptions no longer hold in case a telic relation (i.e., a relation based on interval-based semantics for data) such as STOCK^T is used to represent the same data. This loss of information becomes even more evident if we asks queries on the telic relation STOCK^T. For example, restriction cannot be applied to telic relations (cf. the discussion in Section 7.1), so that we cannot enforce downward hereditary. On the other hand, restriction on the atelic relation STOCK^A gives the (desired) result {<IBM, high-tech, 63 || {15, 16}>}.

(Q4) $Restr^A_{[15,16]}\,(\sigma_{\text{Name=IBM}}(\text{STOCK}^A))$

As regards upward hereditary, let us consider the queries (Q5) and (Q5′), **asking for stocks whose price was more than** 60 for at least 5 consecutive minutes (again, we suppose that temporal selection also applies to atelic relations).

(Q5) $\sigma^T_{\text{duration(VT)} \geq 5}\,(\pi^A_{\text{Name}}(\text{STOCK}^A))$

(Q5′) $\sigma^T_{\text{duration(VT)} \geq 5}\,(\pi^T_{\text{Name}}(\text{STOCK}^T))$

The answer to (Q5) is the relation containing the tuple <IBM ||{12,13,14,15,16,17}>. On the other hand, the answer to (Q5′) **is the empty relation, since downward hereditary does not apply to the telic case.**

Thus, we conclude that *both* telic and atelic models and operators are needed in order to correctly deal with the temporal phenomena pointed out in this paper.

8.2 Three-sorted relational model and algebra

To summarize, our temporal model consists of three sorts of temporal relations: atelic relations, in which the validity times of tuples are atelic elements (cf. Section 4.1), telic relations, in which the validity times of tuples are telic elements (cf. Section 6), plus standard atemporal relations. For instance, both the atelic relation STOCK^A in Section 4.1 and the telic relation PHONE^T in Section 7.2 may be present in a given temporal database.

Our algebra is a three-sorted one consisting of the atelic operators in Section 5 (which are, basically, the algebraic operators underlying SQL/Temporal), the telic operators in Section 7, plus the standard operators for the atemporal algebra.

Unfortunately, the three-partite view of data is too rigid in practice. First of all, operators are needed in order to transform atelic and telic relations into atemporal ones, and vice-versa.

We introduce three temporal functions, τ^A_p, $Transform^T_I$, and $Transform^A_p$, which allow one to obtain the conventional (i.e., atemporal) relation corresponding to a temporal relation, and vice-versa. Notice that we chose to define τ^A_p only on atelic relations, since, by definition, facts in telic relations only occur over time intervals.

Definition $\tau^A_p(R) \equiv \{s \mid \exists t\in R\ \exists i\in t(\mathrm{VT})\ (\ p\in i\ \wedge s(R) = t(R)\ )\}$

The $Transform^A$ (resp. $Transform^T$) function applies to an atemporal relation R and a given time point p (resp. a time interval I) and returns an atelic (resp. telic) relation, with the timestamp of every tuple of p (resp. I).

Definition 2.

$Transform^T_I(R) \equiv \{s \mid \exists t\in R\ (\ s(R) = t(R) \wedge s(\mathrm{VT}) = \{\ I\ \}\ )\ \}$

$Transform^A_p(R) \equiv \{s \mid \exists t\in R\ (\ s(R) = t(R) \wedge s(\mathrm{VT}) = \{\ p\ \}\ )\ \}$

8.3 Need for Further Flexibility: Telic/Atelic Coercion Functions

However, the above transformations do not suffice, since transformation functions that coerce telic relations into atelic ones, and vice versa, are needed. In particular, there are at least a specific and a general motivation for introducing coercion functions between telic and atelic relations. The specific motivation concerns the opportunity of applying temporal selection (and, in general, any “interval query” [Jensen et al., 93]) to atelic relations, as most temporal algebræ do.

The second, more general, motivation, which concerns also the coercion of telic relations into atelic ones, emerges again from the linguistic analysis. In fact, linguistics tell us that the distinction between telic and atelic facts is not at all a rigid one within natural languages (cf. the discussion in Section 3.4). The same flexibility would be greatly desirable in our temporal model, and would significantly increase the expressive power of our approach. Consider again, for instance, our phone call example (cf. relation PHONE^T), and the query “Who made calls during John’s calls to

Mary?” As shown in Section 6.3, phone calls can be regarded as telic facts. On the other hand, when stating “during John’s calls to Mary” we look inside the fact, *coercing* it into an atelic one. In other words, this query involves two different ways of looking at the tuples in relation PHONE^T. First, we have to consider John’s calls to Mary. Such calls must be interpreted as atelic facts, since we are not looking for calls occurred during *one* of John’s calls, but, more generally, *while John was calling Mary*. In other words, we are interested into calls occurred during [10-15], and not during one of [10-12], [13-15]. On the other hand, the calls during [10-15] we are asking for must be interpreted as telic facts, since we look for *each complete occurrence* of them which is fully contained in [10-15]. For example, we want Sue in our output, since Sue made a call in [12-14], which is during [10-15], regardless of the fact that Sue made also another consecutive call from 15 to 16.

We thus need more flexibility: although each base relation must be declared as telic or atelic, we need coercion functions to allow the temporal counterpart of linguistic aktionsart coercion [Moens & Steedman, 88], i.e., to transform telic relations into atelic ones, and vice versa. Other motivating examples for the introduction of temporal coercion functions are shown in Section 9.

In our approach, telic relations can be converted into atelic ones, and vice versa, using the operators α^A (α represents—in Greek—the initial letter of “aktionsart,” and ‘^{A}’ denotes “Atelic”; thus is a shorthand for: “coercing the aktionsart from atelic”) and α^T (‘^{T}’ denotes “Telic”). The definitions of these mapping functions can be easily given in terms of the functions *to-telic* and *to-atelic* provided in Section 6.1.

Given any telic relation R^T,

$$Sch(\alpha^T(R^T)) \equiv Sch(R^T)$$

$$\alpha^T(R^T) \equiv \{\, s \mid \exists t_1 \in R^T \,(\, s(R^T) = t_1(R_1^T) \wedge s(\mathrm{VT}) = \textit{to-atelic}(t_1(\mathrm{VT}))\,)\,\}$$

Given any atelic relation R^A,

$$Sch(\alpha^A(R^A)) \equiv Sch(R^A)$$

$$\alpha^A(R^A) \equiv \{\, s \mid \exists t_1 \in R^A \,(\, s(R^A) = t_1(R_1^A) \wedge s(\mathrm{VT}) = \textit{to-telic}(t_1(\mathrm{VT}))\,)\,\}$$

Example:

As a first example, let us consider the query (Q5) above, which incorrectly applies temporal selection to an atelic relation. Such a query can be correctly expressed as

$$\text{(Q5}''\text{)} \qquad \sigma^T_{\mathrm{duration(VT)} \geq 5}\,(\alpha^A(\pi^A_{\mathrm{Name}}(\mathrm{STOCK}^A)))$$

Moreover, the query “Who made calls during John’s calls to Mary” can be expressed in our algebra as

$$\text{(Q6)} \qquad \pi^T_{\mathrm{Caller}}(\mathrm{PHONE}^T \times^T_{\mathrm{During}} (\alpha^A(\sigma_{\mathrm{Caller=John,\ Called=Mary}}\,(\alpha^T(\mathrm{PHONE}^T)))))$$

In particular, the result of the subquery $\sigma_{\mathrm{Caller=John,\ Called=Mary}}\,(\alpha^T(\mathrm{PHONE}^T))$ is the atelic relation R_1^A = {<John, Mary ||{10,11,12,13,14,15}>}; $\alpha^A(R_1^A)$ gives as result the telic relation R_2^T = {<John, Mary ||{[10-15]}>}; the final result of the query is the telic relation {<John ||{[10-12],[13-15]}>, <Sue ||{[12-14]}>}.

8.4 Properties of the Three-Sorted Algebra

Reduction and equivalence are important properties for a temporal algebra, since they grant that a temporal algebra is a consistent extension of the atemporal classical one. We state both properties for our three-sorted algebra here; proofs may be found in Appendix 1.

Property 1. The atelic operators $\times^A$, π^A, and $-^A$ are snapshot reducible to the analogous conventional relational algebraic operators. As for one case, π^A , snapshot reducibility is stated as $\forall R^A \forall p\ (\tau^A_p\ (\pi^A\ (\mathrm{R}^A)) = \pi\ (\tau^A_p\ (R^A))\ )$.

Property 2. The equivalence property holds for the atelic operators $\times^A$, π^A, and $-^A$ and for the telic operators $\cup^T$ and π^T . As for one case, π^A , equivalence is stated as $\forall R^A \forall p\ (\mathit{Transform}\,(\pi^A(R^A), p) = \pi^A\ (\mathit{Transform}\,(R^A, p))\ )$.

It is also the case that the entire three-sorted algebra (considering also telic relations and operators, and coercion functions) is more expressive than just the atelic one.

Property 3. Given a telic relations R and indicating by Op^A and Op^T analogous (unary) atelic and telic operators, $\exists R^T\ (Op^A(\alpha^T(R^T)) \neq \alpha^T(Op^T(R^T)))$.

The last property states that, if the database consists of both telic and atelic relations, and one coerces all telic relations into atelic ones, the results obtained from queries can be different from those obtained distinguishing between telic and atelic relations. Throughout the preceding sections, and in the next section, we show many cases in which the distinctions between telic and atelic relations/operators is relevant, and in which adopting the proper types (telic or atelic) for relations/operators is important in order to obtain the desired results.

9 Extended SQL/Temporal

The preceding sections focused on extensions to a temporal algebra to add support for both telic and atelic tables. We now show how these concepts can be added to a calculus-based temporal query language, SQL/Temporal [Snodgrass et al. 95b, 98]. As we'll see, only a few new constructs are needed. The specifics (and the adoption of SQL/Temporal) are not as important; the core message is that incorporating the distinction between telic and atelic data into a user-oriented query language is not difficult.

SQL/Temporal augments the existing **`CREATE TABLE`** statement to support valid-time state tables. A valid-time state table, as befits its name, is an atelic table. One way to extend this language further is to allow the definition of telic tables, with the following syntax, adding one more keyword, **`TELIC`**.

```
CREATE TABLE B AS TELIC(DAY)
```

where **`DAY`** is the granularity desired for the timestamp.

SQL/Temporal augments the existing **`SELECT`** statement (as well as existing statements for modifications, views, cursors, integrity constraints, and assertions) with two optional prefix keywords, **`NONSEQUENCED`** and **`VALIDTIME`**. Without these keywords, the query (modification, view, etc.) is interpreted as a current query, on the current state of the table. As such, it performs an implicit temporal restriction, and thus no telic tables should participate in a current query. With the `VALIDTIME` keyword, the query is interpreted as a sequenced query, effectively applying the query separately at every point in time. Again, this is relevant only for atelic tables, and effects an atelic semantics. The third type of query, a nonsequenced query, is specified with the prefix `NONSEQUENCED VALIDTIME`. Such queries interpret the implicit timestamp as just another column of the table, yielding a nontemporal table.

One way to extend this language to effect a telic semantics is to substitute our new `TELIC` keyword for `VALIDTIME`, as a prefix.

We also need coercion constructs. Nonsequenced queries effect a conversion of the underlying tables to atemporal. Since telic tables are more restricted in the types of queries they can participate in, coercion from telic to atelic is important, and so an extension of the `FROM` clause is useful here: `FROM A AS VALIDTIME`. That suggests the coercion the other way: `FROM A AS TELIC`.

Alternatively, coercions could be implicitly invoked whenever the type of query (telic or atelic) did not match the type of an underlying table. However, we prefer explicit coercions, so that the reader understands clearly the semantics of the query, and so that the query processor can detect places where a coercion was not needed, as well as illegal coercions. So in the following we will assume that all coercions must be explicit.

There is one place where implicit coercions can be allowed. We can inherit from TSQL2 explicit coalescing [Böhlen et al. 96] of a telic timestamp, stated as e.g., `FROM A (VALIDTIME)`; this would be expressed in our algebra as $\alpha^A(\alpha^T(A))$.

10 Examples

In the following we assume that the temporal database contains the atelic relation STOCK^A (Section 4.1) and the telic relation PHONE^T (Section 7.2) and provide some examples of queries and answers in our approach. In these examples, we express the queries both in the relational algebra and in the extended SQL/Temporal query language. The objective is to show how queries, both simple and complex, can be expressed in the algebra and in SQL/Temporal.

The tables are first created using the following statements.

```
CREATE TABLE STOCK (Name, Category, Price) AS VALIDTIME(MINUTE)
CREATE TABLE PHONE (Caller, Called) AS TELIC(MINUTE)
```

The first few queries were given earlier. “Who made phone calls to whom during 10-11?”

(Q1) $\mathit{Restr}^A_{[10,11]}\ \mathbf{(}\alpha^T\ (\mathrm{PHONE}^T\mathbf{))}$

This atelic query is easy to express in conventional SQL/Temporal, using an expression after the `VALIDTIME` prefix, and using `AS VALIDTIME` in the `FROM` clause to coerce PHONE^T from telic to atelic.

```
VALIDTIME PERIOD(10-11) SELECT * FROM PHONE AS VALIDTIME
```

“Who made phone calls after John called Mary?” Here after treats the phone calls as the telic data it is.

(Q33′) $\mathrm{PHONE}^T \times^T_{\mathrm{After}} (\sigma_{\mathrm{caller=John,called=Mary}}\ (\mathrm{PHONE}^T))$

Everything is in telic, which makes this query easy to express in our augmented SQL/Temporal.

```
TELIC SELECT P1.Caller, P1.Called
FROM PHONE AS P1, PHONE AS P2
WHERE P2.Caller='John' AND P2.Called='Mary'
```

```
            AND VALIDTIME(P1) AFTER VALIDTIME(P2)
```

The `VALIDTIME()` function returns the implicit timestamp of the tuple, for use in predicates such as `AFTER`.

Let’s now look at some complex queries, to see what can be expressed. A coercion is required in the query, “Who made at least one complete call (during the time) when IBM’s price was more than $60?”

$$\text{(Q7)} \qquad \pi^T_{\text{Caller}}(\text{PHONE}^T \times^T_{\text{During}} \alpha^A(\pi^A_{\text{Name}}(\sigma_{\text{Name=IBM, Price>60}}(\text{STOCK}^A))))$$

Here PHONE^T is already telic, but STOCK^A needs to be converted, but only after the select and project operators, and so is done as a nested query. The *telic* result is: {<John || {[13-15]}>, <Sue ||{[12-14], [15-16]}>, <Eric || {[14-16]}>}

This example shows the importance of having both telic relations (in this case: PHONE^T), on which temporal coercion must not be performed, and atelic ones (in this case: STOCK^A), with temporal coercion. Notice that, since STOCK is an atelic relation, the projection $\pi^A_{\text{Name}}(\sigma_{\text{Name=IBM, Price>60}}(\text{STOCK}^A))$ makes the point-union of the validity times of value-equivalent tuples. In other words, temporal coalescing is applied, so that the (atelic) tuple <IBM, {12,13,14,15,16,17}> is returned as the result of the projection. Thus, the tuple <John || {[13-15]}> is included in the final result, even though [13-15] is not included in any of the time intervals in which IBM was more than $60 (i.e., [12-13], [14-17]) considered in isolation. On the other hand, since PHONE^T is telic, no temporal coercion is performed on the validity time of the tuple <John, Mary || {[10-12], [13-15]}>. Thus, the tuple <John || {[13-15]}> is in the final result, as desired. Finally, notice that the coercion α^{T} must be used to apply the interval query “during” (i.e., $\times^{\text{T}}_{\text{During}}$) to the atelic relation $\pi^A_{\text{Name}}(\sigma_{\text{Name=IBM, Price>60}}(\text{STOCK}^A))$. Moreover, using telic Cartesian product, we return the time intervals of phone calls as the validity time of the result.

This query may be expressed in our SQL/Temporal extension as follows.

```
TELIC SELECT Caller
FROM PHONE, (VALIDTIME SELECT Name FROM STOCK
             WHERE Name = 'IBM' AND Price >60) AS TELIC A
WHERE VALIDTIME(PHONE) DURING VALIDTIME(A)
```

The outermost query is a telic query (required since PHONE^T is telic); the nested query is atelic, with its result coerced to telic.

“Tell me about stocks whose price was $61 during the time when John was calling Mary.”

$$\text{(Q8)} \qquad \alpha^A(\sigma_{\text{Price=61}}(\text{STOCK}^A)) \times^T_{\text{During}} \alpha^A(\alpha^T(\sigma_{\text{Caller=John, Called=Mary}}(\text{PHONE}^T)))$$

This query exemplifies the need of coercion from telic to atelic. In fact, we have an inner view (“was calling”) of a telic relation (PHONE^T). Since PHONE^{T} is telic, the result of $\sigma_{\text{Caller=John,Called=Mary}}$ (PHONE^T) is the telic relation {<John, Mary || {[10-12], [13-15]}>}. However, in (Q3), we are not interested to distinguish different occurrences of John’s calls to Mary, but we see calling Mary as an atelic fact, looking for the overall time in which John was calling her. We thus apply the α^T coercion operator, obtaining the atelic tuple {<John, Mary {10,11,12,13,14,15}>}. Thus, the tuple <IBM, High-tech ||{ [12-13] }> is in the result even if IBM price was $61 in the time interval [12-13], which is not contained in any of John’s calls taken in isolation. Moreover, notice that the $\times^T_{\text{During}}$ operator only applies to telic tuples, thereby explaining the need of the coercions α^{A} to both sides of the operator. The telic result is thus: {<IBM, High-tech ||{[12-13]}>}.

This (rather complex) query can be expressed in the SQL/Temporal extension as follows.

```
TELIC SELECT *
FROM (VALIDTIME SELECT * FROM STOCK WHERE Price =61) AS TELIC S,
```

```
         (TELIC SELECT * FROM PHONE
          WHERE Caller=’John’ AND Called=’Mary’)(VALIDTIME) AS P
    WHERE VALIDTIME(S) DURING VALIDTIME(P)
```

The outermost query is telic, as is the second nested query; the first nested query is atelic (then coerced into telic). The coalescing is expressed as “`(VALIDTIME)`”.

The central point is that the user needs to be aware of whether a table is telic or atelic, and also how that data is to be manipulated. If the query language does not support this distinction explicitly, all manner of problems arise, as discussed in Section 3.

11 Other Potential Approaches

Our approach provides a general solution to the difficulties of mixing point-based and interval-based data. However, another tack is to see if existing language facilities can be exploited to solve the same problem. In this section, we examine briefly some of these alternatives.

11.1 1NF

Let us suppose to impose a temporal 1NF [Gadia 88] as, e.g., in TSQL and in HQuel, so that just one time interval is associated with each tuple, instead of a temporal element. In such approaches, for instance, relation PHONE^{T} could be represented by relation PHONE^{1NF} below.

PHONE^{1NF}

Caller	Called	VT
John	Mary	[10-12]
John	Mary	[13-15]
Sue	Ann	[12-14]
Sue	Ann	[15-16]
Eric	Paul	[14-16]

However, if one adopts the point-based semantics for data, this transformation alone does not solve the problem. For example, the first two tuples of PHONE^{1NF} still carry on the content that John was calling Mary on 90, 11, 12, 13, 14, and 15. And, in fact, the distinction between 1NF with respect to Not-1NF is relevant at the representation level, but not at the conceptual and semantic level. On the other hand, one can use single interval timestamping as above, and *never* perform coalescing of value-equivalent tuples, as in SQL/Temporal. In such a case, one cannot use a set-based semantics (as we did in this paper), since the treatment of the semantics of duplicates involves additional complications. More importantly, an approach which never performs coalescing has the same kind of problems discussed in Section 8.2 when dealing with the use of telic relations and operators to model atelic tuples, since upward and downward hereditary would never hold. Basically, any "homogeneous" approach in which upward and downward hereditary hold on all relations (as in TSQL2), or do not hold in any relation (as in SQL/Temporal), will not be satisfactory.

11.2 Static Semantic Properties of Relations

A "static" approach in which one specifies once and forever which relations are atelic (i.e., both upward and downward hereditary hold) or telic (i.e., both upward and downward hereditary do not hold) also presents some drawbacks. In particular, the examples in Sections 8.3 and 9 show that approaches that always perform coalescing (as TSQL2) or never perform it (as SQL/Temporal), as well as approaches which have to fix *a priori* on which relations/attributes coalescing has to be performed and on which not, with no possibility of changing this property at query time (cf., e.g., [Bettini et al., 98]) do not properly cope with the atelic/telic dichotomy.

11.3 Using Additional Attributes and Surrogates

Another approach is to design database schemas in such a way that no value-equivalent tuples occur in the base relations, so that at least the problem of dealing with upward hereditary (temporal coalescing) vanishes. For instance, in our phone-call example, one could add an additional attribute containing the sequence number of calls for each phone number to ensure that relations contain no value-equivalent tuples.

PHONE^{ATTR}

Caller	Called	Seq	VT
John	Mary	1	[10,12]
John	Mary	2	[13,15]
Sue	Ann	1	[12,14]
Sue	Ann	2	[15,16]
Eric	Paul	1	[14,16]

However, in the example above and in many practical cases, forcing that no value-equivalent tuples occur in base relations requires the introduction of new attributes which are often scarcely useful and informative. This imposes a strong requirements over database designers, who are forced to introduce attributes which in some cases are not significant by themselves (and are not part, e.g., of the entity relationship model), but must be added just to ensure that no data-equivalent tuples belong to the same base relation. Moreover, notice also that this would be only a partial solution to the problems discussed in this paper, which would arise again for derived relations. For instance, if one projects away the additional Seq attribute, one obtains again the original relation, with all the limitations discussed throughout this paper. [5]

---

[5] Of course, the problems related to an indiscriminate use of upward hereditary do not occur in case one first applies temporal operators and aggregate functions, and only later performs projection. But, again, we are imposing strong constraints not only on the design of the model, but also on the definition of legal queries.

Finally, one could also introduce a surrogate attribute, which is used to keep the identity of the fact (i.e., it is a key attribute) and is hidden to the user [Jensen & Snodgrass 96]. Notice that, in such a way, one models exactly the semantics of telic tuples, since the surrogate attribute will represent explicitly the different “episodes” (occurrences) of a given telic fact. Thus, this solution can be simply conceived as a possible partial implementation of our telic model, to avoid upward hereditary (or, in other words, to prevent temporal coalescing). However, such a partial implementation should be augmented to deal with the other aspects of our three-sorted algebra. In particular, in Section 8 we showed that both telic and atelic relations (and operators) are useful, and that an *a priori* and fixed-forever distinction between telic and atelic relations is too restrictive, so also that coercion functions should be available.

11.4 Temporal Interpolation Approaches

We see some analogies between our approach and the TDB approaches dealing with *temporal interpolation*. Elsewhere we [Jensen & Snodgrass 96] have described the temporal interpolation problem as the problem of devising suitable techniques to derive information for times for which no information is stored, on the basis of related information holding at different times. For example, Clifford and Warren [84] proposed a *continuity assumption*, which states that a value for a given attribute of a given tuple holds until a new value is explicitly recorded. Segev and Shoshani [87] adopted a point-based semantics for data, looking at attributes as a function from time points to values, and introduced six different types of interpolation operators that compute the value of such a function at a given time point, given its values over other points. Recently Bettini et al. [98] proposed to explicitly associate with each table a specification of the assumptions on the semantics of temporal attributes (e.g., persistence of data), expressed in a formal language. At query time, such specifications are automatically merged with the user’s query in order to provide the correct results. Bettini also considered *interval assumptions*, which mainly involve the conversion of values across different granularities, including upward and downward hereditary. However, upward and downward hereditary are only studied in the context of evaluating the values of attributes whose validity time is expressed at different time granularities. Moreover, in their approach, one has to state once and for all the properties of a given table, which cannot be changed at query time (unless a granularity change is made).

11.5 Summary

This discussion of alternative solutions to the point/interval quandary examined four possibilities: INF, static declaration of point or interval semantics for data, using additional attributes or surrogates, and using temporal interpolation facilities. In the first two cases, the problem was only partially solved. In the last two cases, it may be possible to express what is desired, but requires significant effort to fit this distinction into a formalisms which was not designed with this purpose in mind.

Instead, we feel that the atelic/telic distinction is so central that it should be given first-class status in both the data model and query language, especially as doing so requires so few changes to either.

12 The Point-based Versus Interval-Based Controversy Revisited

In a recent paper, Toman [98] pointed out some problems connected with the definition of a clear semantics for the approaches where the validity times of tuples/attributes are encoded using time intervals (as, for example in TSQL2 and SQL/Temporal): “this approach lead to a tension between the syntax of the query languages and their intended semantics: the data model and the semantics of the language are point-based, while temporal attributes refer to the actual encoding for sets of time intervals (e.g., interval endpoints)" (page 212). Toman [96] proved that for every point-based query there is an equivalent interval-based query, and vice versa[6], a statement that seems to fly in the face of the atelic/telic distinction, and indeed of the entire discussion of the present paper. Even more interestingly, we fully agree with Toman's statement. The resolution of the paradox is that Toman always assumes a point-based semantics *for the data*. His "interval-based temporal database" in our terminology is a point-based semantics data model represented by an interval-based encoding. So a rephrasing of that statement, in our terminology, is that for every point-based query, there is an equivalent interval-based query over *point-based data* represented by an interval-based encoding, a statement which seems quite reasonable.

Toman then proposed a new model and language which are more purely point-based; the only appearance of intervals is in the specific encoding that he used to implement of this language; no notion of interval appears either in the data model nor in his SQL/TP (for time-point) query language. We agree that having the interval-based nature of the encoding appear in a restricted way in the query language raises problems; however, we feel that it is more appropriate to emphasize the distinction between atelic and telic facts, rather than jettisoning telic facts altogether, thereby allowing data with a semantics such as in Figure 3 and queries such as atelic queries on telic data and telic queries on telic and atelic data, as given in Section 10.

Chomicki and Toman [98] clearly point out the distinction between abstract and concrete temporal databases, and between data and query language. Furthermore, the framework for multidimensional time they introduce is a very general and powerful one, and could be applied to model both atelic and telic elements. In particular, time points in our approach might correspond to their 1-dimension points, and our telic intervals to their 2-dimensional points. On the other hand, they do not propose any specific treatment of the telic/atelic dichotomy, so that, for instance, no counterpart of our coercion function is taken into account. Thus, we believe that the considerations we made in Section 11.5 also apply to their approach.

In ATSQL2 and when using temporal statement modifiers in general [Böhlen et al. 00], the data semantics is purely point-based (atelic) and the query semantics is almost entirely atelic (except for duration and interval comparison predicates). However, there is some leeway in choosing a representation of the result, as there are potentially many snapshot-equivalent representations of the result. Their notion of *interval preservation* selects among these representations that which best preserves the underlying intervals (this notion was first introduced by Böhlen et al. [98]). Through their *non-restrictiveness* property, they allow the interval timestamps to be converted into values of an explicit attribute, thereby enabling interval-based queries to be simulated in conventional SQL, often with difficulty, as SQL has little notion of time.

13 Conclusions and Future Work

The analysis of the semantics of temporal data and queries plays a central role in the area of TDBs, since *"data explicitly stored in a temporal database are often associated with certain semantics assumptions"* [Bettini et al. 98, page 277]. Although many different models have been proposed in the TDB literature, almost all of them are based on a

---

[6] Specifically, Toman [96] in Theorem 5.5 restricted his attention to a particular form of interval query, which in our terminology correspond to telic queries,

point-based (snapshot) semantics for the association of tuples (or attributes) to time. On the other hand, in the areas of linguistics and philosophy, many approaches stressed the fact that point-based semantics is useful for atelic facts, but is not adequate to cope with telic facts, for which an interval-based semantics is needed. In this paper, we introduced an original three-sorted sorted model and algebra which properly copes with both telic and atelic facts, and which achieves a great flexibility via the introduction of coercion functions for transforming tables of one sort into tables of the other, at query time. Reduction and equivalence with respect to the classical atemporal algebra hold for our temporal algebra.

Our overall approach makes the following contributions.

(i) Both the data model and query language are quite expressive, since they emphasize the telic/atelic dichotomy, which has not to this juncture been dealt with by any other temporal database approach. In particular, we offer a general approach to handling telic facts, either in isolation or in conjunction with atelic facts.

(ii) We clarify several subtle issues concerning the adoption of a point-based versus interval-based semantics, making also clearer the distinction between data language and data semantics, and between query and data semantics. We do so by exploiting the deep understanding of these issues offered by the philosophic and linguistic literature.

(iii) We show how to add the atelic/telic distinction to temporal data models and query languages. While we chose to adapt SQL/Temporal's user language and algebra to deal with atelic relations, this choice is by no means restrictive, as long as one starts with an algebra that uses point-based semantics for data.

(iv) Despite the fact that we add the expressiveness needed in order to properly cope with telic facts, the changes in the query language are minor. This is an interesting property for users, who does not have to learn too many different syntactic constructs with respect to SQL/Temporal. Once again, we would have had a similar result also choosing most of the other temporal query languages in the literature.

We feel that the atelic/telic distinction is so central that it should be given first-class status in both the data model and query language, especially as doing so requires so few changes to either.

As future work, we would like to extend our approach to be even more comprehensive. An analysis of the linguistic analysis provides other relevant distinctions between types of propositions besides the telic/atelic one discussed in this paper. In particular, in Section 2 we briefly considered all Vendler's aktionsart classes, taking into account also *activities* and *achievements*. While the treatment of *activities* in TDBs does not seem to involve any deep departure from the "classical" point-based models (cf. our "prototypical" model in Section 3.1, and the discussion in Appendix 2), let us consider now *achievements*. In our approach (as in the case of point-based classical approaches) achievements could be dealt with via the extension of the model to consider also punctual telic tables, in which the validity time is a set of time points, each one representing the time when an individual occurrence of the fact took place. This is similar, e.g., to the use of *validity time event tables* in TSQL2. Thus, a fourth temporal sort could be introduced in our temporal model, and the algebraic operators and coercion functions should be extended accordingly.

We also would like to consider the possibility of extending our approach to deal with other relevant temporal properties, such as temporal persistence [Bettini et al. 98], interpolation functions [Segev & Shoshani 87, Jensen & Snodgrass 96, Bettini et al. 98]. We want to extend these approaches to take into account the impact of aktionsart distinctions on the semantics of TDBs.

Another interesting issue concerns the treatment of imprecise temporal information [Dyreson & Snodgrass 98] and of periodic times in TDBs. We recently devised an approach to deal with approximate dates and durations, as well as qualitative constraints such as "the validity time of tuple t_1 contains the validity time of t_2" stored as temporal data in the relational tables [Brusoni et al. 95, 99]. In such an approach, we only considered "standard" atelic temporal tables. In the future, we would like to extend such an approach to cover also telic tables.

We also proposed [Terenziani 99, 01] a semi-symbolic approach to the treatment of periodicity in temporal relational databases, which extends and improves the results of Niezette & Stevenne [92]. We strongly believe that extension of such an approach to consider also telic time intervals could greatly improve the expressiveness of the approach, making it suitable to deal also with the so-called "nearly-periodic events" [Tuzhilin & Clifford 95].

ACKNOWLEDGEMENTS

We thank Christian S. Jensen for stimulating discussions, and National Science Foundation Grant EIA-008012 for partial support of this research.

14 REFERENCES

[Aksu 78] A. Aksu. Aspect and Modality in the Child's acquisition of the Turkish Past Tense. Dissertation. Univ. of California at Berkeley, 1978.

[Allen 83] J.F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM* 26(11):832–843, 1983.

[Allen 84] J.F. Allen, "Towards a General Theory of Action and Time," *Artificial Intelligence* 23:123–154, 1984.

[Allen 91] J.F. Allen, "Time and Time Again: The Many Ways to Represent Time," *International Journal of Intelligent Systems* 6(4), 341–355, 1991.

[Androutsopoulos et al. 95] I. Androutsopoulos, G.D. Richie and P. Thanisch, "Experience using TSQL2 in a natural language interface," in [RATD, 95], 113–132, 1995.

[Antinucci & Miller 73] F. Antinucci and R. Miller. "How children talk about what happened," *Journal of Child Language 3*, 167–189, 1973.

[Ariav et al. 83] G. Ariav, J. Clifford and M. Jarke, "Panel on Time and Databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Jose, 243–245, 1983.

[Aristotle] Aristotle, **The Categories, on Interpretation. Prior Analytics**. Cambridge, MA, Harvard University Press.

[Bacchus et al. 91] F. Bacchus, J. Tenenberg, J. Koomen. "A Non-reified Temporal Logic," *Artificial Intelligence* 52, 87-108, 1991.

[Bach 86] E. Bach, "The algebra of events," *Linguistics and Philosophy* 9, 5–16, 1986.

[Bennet & Partee 72] M. Bennet and B. Partee, "Tense and Discourse Location In Situation Semantics," distributed 1978 by the Indiana University Linguistics Club, Bloomington.

[Bertinetto 86] P.M. Bertinetto. **Tempo, Aspetto e Azione nel Verbo Italiano**. Accademia della Crusca. Firenze. 1986.

[Bettini et al. 98] C. Bettini, X.S. Wang and S. Jajodia, "Temporal Semantic Assumptions And Their Use In Databases," *IEEE Transactions on Knowledge and Data Engineering* 10(2), 1998.

[Bloom et al. 80] L. Bloom, L.Lifter and J. Hafitz, “Semantics of Verbs and the Developments of Verb Inflection in Child Language,” *Language* 52(2):386–412, 1980.

[Böhlen et al. 96] M. Böhlen, R. T. Snodgrass and M. Soo, “Coalescing in Temporal Databases,” in *Proceedings of the International Conference on Very Large Databases*, 180–191, 1996.

[Böhlen et al. 98] M. Böhlen, R. Busatto and C. S. Jensen, “Point- Versus Interval-Based Data Models,” in *Proceedings of the IEEE International Conference on Data Engineering*, pp. 192–200, 1998.

[Böhlen et al. 00] M. Böhlen, C. S. Jensen and R. T. Snodgrass, “Temporal Statement Modifiers,” to appear in *ACM Transactions on Database Systems*, 25(4), December, 2000.

[Bronckart & Sinclair 73] J.P. Bronckart, and H. Sinclair. “Time, tense and aspect,” *Cognition 2*, 107–130, 1973.

[Brusoni et al. 95] V. Brusoni, L. Console, B. Pernici and P. Terenziani, “Extending Temporal Relational Databases to Deal With Imprecise and Qualitative Temporal Information,” in [RATD, 95], 3–22, 1995.

[Brusoni et al. 99] 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*, to appear.

[Bubenko 77] J.A. Bubenko, “The Temporal Dimension In Information Modelling,” In **Architecture and Models in Data Base Management Systems,** G. Nijssen, ed., North Holland, 1977.

[Bull 60] W.E. Bull. **Time, Tense, and the Verb. A Study in Theoretical and Applied Linguistics, with Particular Attention to Spanish**. Berkley. Los Angeles. 1960.

[Carlson 81] L. Carlson. Aspect and Quantification. In: Tedeschi and Zaenen (eds.) *Syntax and Semantics 14*: *Tense and Aspect*. New York, Academic Press, 31–64, 1981.

[CL 88] Special Issue on Tense and Aspect, B.L. Webber (guest ed.), *Computational Linguistics* 14(2), 1988.

[Chomicki 94] J. Chomicki, “Temporal Query Languages: A Survey,” in *Proceedings of the International Conference on Temporal Logic,* Springer-Verlag (LNAI 827), pp. 506–534, July, 1994.

[Chomicki and Toman 98] J. Chomicki and D. Toman, Chapter 3, “Temporal Logic in Information Systems”, Logics for Databases and Information Systems, J. Chomicki and G. Saake (eds.), **pages 31–70, March 1998.**

[Clifford 88] J. Clifford, “Natural Language Querying of Historical Databases,” *Computational Linguistics* 14(4), 1988.

[Clifford & Ariav 86] J. Clifford and G. Ariav, “Temporal Data Management: Models And Systems,” Chapter 12 of **New Directions for Database Systems**, ABLEX Publishing Norwood, pp. 168–185, 1986.

[Clifford & Isakowitz 94] J. Clifford and T. Isakowitz, “On the Semantics of (Bi)Temporal Variable Databases,” M. Jarke, J. Bubenko and K. Jeffrey (eds.), in *Proceedings of the Fourth International Conference on Extending Database Technology,* 215–230, 1994.

[Clifford & Tansel 85] J. Clifford and A.U. Tansel, “On an Algebra for Historical Relational Databases: Two Views,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, S. Navathe (ed.), Austin, TX, 247–265, 1985.

[Clifford & Warren 83] J. Clifford and D.S. Warren, “Formal Semantics for Time in Temporal Databases,” *ACM Transactions on Database Systems* 8(2), 214–254, 1983.

[Copestake & Spark Jones 90] A. Copestake and K. Spark Jones, “Natural language Interfaces to Databases,” *The Knowledge Engineering Review* 5(4), 225–249, 1990.

[Coseriu 79] E. Coseriu. Verbalinhalt, Aktanten, Diathese. Zur japanische ukemi-bildung, In: K. Ekawa, K.H. Rensch, and W. Bethge (eds.) **Sprache und Sprachen**. Tübingen, 35–55, 1979.

[Dahl 81] Ö. Dahl. "On the definition of telic-atelic (bounded-unbounded) distinction," in: Tedeschi and Zaenen (eds.) *Syntax and Semantics 14*: *Tense and Aspect*. New York, Academic Press, 1981.

[Declerk 79] R. Declerk. "Aspect and the bounded/unbounded (telic/atelic) distinction," *Linguistics 17*, 761–794, 1979.

[Dyreson & Snodgrass 98] C. Dyreson and R. T. Snodgrass, "Supporting Valid Time Indeterminacy," *ACM Transactions on Database Systems* (23)1: 1–57, March, 1998.

[Dowty 79] D. R. Dowty. **World Meaning and Montague Grammar**. D. Riedel, Dordrecht, 1979.

[Dowty 86] 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.

[Emerson 90] A.E. Emerson, "Temporal and Modal Logic," in: Van Leeuwen (ed) *Handbook of Theoretical Computer Science*, Vol. B, pp. 997-1072, Elsevier Science Publishers, 1990.

[Franconi et al. 93] E. Franconi, A. Giorgi, F. Pianesi, "Tense and Aspect: A Mereological Approach," in *Proc. 13th International Joint Conference on Artificial Intelligence*, Chambery, France, 1222–1228, 1993.

[Gadia 86] S.K. Gadia, "Weak Temporal Relations," In *Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pp. 70–77, New York, MA, 1986.

[Gadia 88] S.K. Gadia, "A Homogeneous Relational Model and Query Languages for Temporal Databases," *ACM Transactions on Database Systems* 13(4), 418–448, 1988.

[Gadia et al. 92] S.K. Gadia, S. Nair and Y.C. Poon, "Incomplete Information in Relational Temporal Databases," in *Proceedings of the International Conference on Very Large Databases*, Vancouver, Canada, 1992.

[Galton 91] A. Galton, "A Critical Examination of Allen's Theory of Action and Time," *Artificial Intelligence* 42, 159–188, 1991.

[Garey 57] H. Garey, "Verbal Aspect in French," *Language* 33, 91–110, 1954.

[Guarino 95] N. Guarino, "Formal Ontology, Conceptual Analysis and Knowledge Representation," in [HCS, 95], 625–640, 1995.

[Heinämäki 78] O. Heinämäki. **Semantics of English Temporal Connectives**. Indiana Univ. Linguistic Club, 1978.

[Hobbs 95] J. Hobbs, "Sketch of an Ontology Underlying the Way We Talk About The World," in [HCS, 95], 819–830, 1995.

[Hwang & Schubert 93] C.H. Hwang and L.K. Schubert, "EL: A Formal, Yet Natural, Comprehensive Knowledge Representation," in *Proceedings AAAI'93*, Washington, DC, 1993.

[HCS 95] Special Issue on "The Role of Formal Ontology in the Information Technology," *International Journal of Human-Computer Studies* 43, N.Guarino and R. Poli, guest editors, Academic Press, 1995.

[Ikegami 81] Y. Ikegami. **'Acrivity'-'accomplishment'-'achievement'. A language that can't say *I burned it, but it didn't burn* and one that can**. Linguistic Agency Univ. of Trier, series A, n.87.

[IWITD 93] R. T. Snodgrass (ed), *Proceedings of the International Workshop on an Infrastructure for Temporal Databases*, Arlington, Texas, 1993.

[Jensen et al. 93] C. Jensen, J. Clifford, S.K. Gadia, F. Grandi, P.P. Kalua, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, S.S. Nair, E. Peressi, B. Pernici, E.L. Robertson, J.F. Roddick, N.L. Sarda, M.R. Scalas, A. Segev, R.T. Snodgrass, A. Tansel, R. Tiberio, A. Tuzhilin, and G.T.J. Wuu, "A Consensus Test Suite of Temporal Database Queries" , in *Proceedings of the International Workshop on an Infrastructure for Temporal Databases,* *Arlington, TX,* R. T. Snodgrass, editor, June 14–16, 1993, pp. QQ1–QQ28; see also Technical Report R 93-2034, Dept. of Mathematics and Computer Science, Aalborg University, Denmark, November 1993.

[Jensen & Dyreson 98] C. S. Jensen and C. E. Dyreson (eds), “A Consensus of Glossary of Temporal Database Concepts—February, 1998 Version,” in **Temporal Databases: Research and Practice**, O. Etzion, S. Jajodia and S. Sripada (eds), Springer-Verlag, pp. 367–405, 1998.

[Jensen & Snodgrass 96] C.S. Jensen and R.T. Snodgrass, “Semantics of Time-varying Information,” *Information Systems* 21(4), 311–352, 1996.

[Jensen & Snodgrass 99] C. Jensen and R.T. Snodgrass, “Temporal Data Management,” *IEEE Transactions on Knowledge and Data Engineering* 11(1), 36–44, 1999.

[Jespersen 24] O. Jespersen. **The Philosophy of Grammar**. Allen and Unwin, London, 1924.

[Jonson 81] M.R. Jonson, “A unified temporal theory of tense and aspect,” in: Tedeschi and Zaenen (eds.) *Syntax and Semantics 14*: *Tense and Aspect*. New York, Academic Press, 145–175, 1981.

[Kenny 63] A. Kenny. **Action, Emotions and Will**. Humanities Press. 1963.

[Klein 74] H.G. Klein. **Tempus, Aspekt, Aktionsart**. Tübingen, 1974.

[Koubarakis 93] M. Koubarakis, “Representation And Querying Temporal Databases: The Power of Temporal Constraints,” in *Proceedings of the IEEE International Conference on Data Engineering*, 327–334, 1993.

[Kowalski & Sergot 86] R. Kowalski, M. Sergot, “A Logic-Based Calculus Of Events,” *New Generation Computing* 4, pp. 67–95, 1986.

[Lenat & Guha 94] D.B. Lenat and R.V. Guha, **Building Large Knowledge-Based Systems: Representation and Inference in the Cyc project**. Addison Wesley Publishing Company, 1994.

[Ling & Bell 90] D.H.O. Ling and D.A. Bell, “Taxonomy Of Time Models In Databases,” *Information. Software Technology* 32(3), 215–224, 1990.

[McCarthy 68] J. McCarthy, “Programs With Common Sense,” **Semantic Information Processing**, L.M. Minsky (ed.), 403–418, MIT Press, Cambridge, MA, 1968.

[Moens & Steedman 88] M. Moens, M. Steedman, “Temporal Ontology and Temporal Reference,” *Computational Linguistics* 14(2), 15–28, 1998.

[McKenzie & Snodgrass 91] L.E. McKenzie and R.T. Snodgrass, “Evaluation of Relational Algebras Incorporating the Time Dimension in Databases,” *ACM Computing Surveys* 23(4), 501–543, 1991.

[Moens 88] M. Moens, “Temporal Databases and Natural Language,” in *Proceedings on Temporal Aspects in Information Systems*, C. Rolland, F. Bodart and M. Leonard, eds, Elsevier Science, 1988.

[Mommer 86] K.E. Mommer. Theoretical issues concerning inherent aspect and the perfect in English, Cebaari and Swahili. Ph.D. Dissertation, Northwestern University, Evanston, Illinois, 1986.

[Mourelatos 78] A. Mourelatos, “Events, Processes and States,” *Linguistics and Philosophy* 2, 415–434, 1978.

[Navathe & Ahmed 87] S. Navathe and R. Ahmed, “TSQL- A Language Interface For History Databases” , in *Proceedings of the Conference on Temporal Aspects in Information Systems*, C. Rolland, F. Bodart and M. Leonard, eds, Elsevier Science, AFCET, 113–128, France, May 1987.

[Navathe & Ahmed 89] S. Navathe and R. Ahmed, “A Temporal Relational Model and a Query Language” , *Information Sciences* 49, 147–175, 1989.

[Niezette & Stevenne 92] M. Niezette and J.-M. Stevenne, “An Efficient Symbolic Representation of Periodic Time” , *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, Baltimore, Maryland, Nov. 1992.

[Özsoyoğlu & Snodgrass 95] G. Özsoyoğlu and R.T. Snodgrass, “Temporal and Real Time Databases: A Survey,” *IEEE Transactions on Data and Knowledge Engineering* 7(4), 513–532, 1995.

[Peckham & Marianski 88] J. Peckham and F. Marianski, “Semantic data models,” *ACM Computing Surveys* 20(3), 153–189, 1988.

[RATD 95] J. Clifford and A. Tuzhilin (eds.), **Recent Advances in Temporal Databases**. Proceedings of the International Workshop on Temporal Databases, Workshop in Computing Series, Springer Verlag, 1995.

[Roddick & Patrick 92] J.F. Roddick and J.D. Patrick, “Temporal Semantics In Information Systems: A Survey,” *Information Systems* 17(3), 249–267, 1992.

[Šabršula 63] J. Šabršula, “La signification des verbes français et les problèmes d’aspect. (étude comparative: langue français et langues slaves),” *Beiträge zur Romanische Philologie, 1*. 166–179, 1963.

[Sardeghi 87] R. Sardeghi, A Database Query Language for Operations on Historical Data, Ph.D. Dissertation, Dundee College of Technology, 1987.

[Schubert & Hwang 89] L.K. Schubert and C.H. Hwang, “An Episodic Knowledge Representation For Narrative Texts,” In *Proceedings First International Conference on Knowledge Representation and Reasoning (KR’89)*, 444–458, Toronto, Canada, 1989.

[Segev & Shoshani 87] A. Segev and A. Shoshani, “Logical Modelling of Temporal Data,” U. Dayal and I. Traiger (eds.), in *Proceedings ACM SIGMOD International Conference on Management of Data*, 454–466, San Francisco, CA, 1995.

[Shoham 87] Y. Shoham, “Temporal Logics in AI: Semantical and Ontological Considerations,” *Artificial Intelligence* 33, 89–104, 1987.

[Snodgrass 87] R.T. Snodgrass, “The Temporal Language TQuel,” *ACM Transactions on Database Systems* 12(2), 247–298, 1987.

[Snodgrass et al. 93] R.T. Snodgrass, S. Gomez and L.E. McKenzie, “Aggregates in the Temporary Query Language TQuel,” *IEEE Transactions on Knowledge and Data Engineering* 5(5), 826–842, 1993.

[Snodgrass et al. 94] R.T. Snodgrass, I. Ahn, G. Ariav, P. Bayer, J. Clifford, C. Dyreson, F. Grandi, L. Hermosilla, C. Jensen, W. Käfer, C. Leung, N. Lorentzos, Y. Mitsopoulos, J. Roddick, M. Soo, S. Sripada, “An Evaluation of TSQL2,” Technical Report, available at `FTP.cs.arizona.edu/tsql/tsql2/sql3`.

[Snodgrass et al. 95] R.T. Snodgrass (editor), I. Ahn, G. Ariav, D. Batory, J. Clifford, C. Dyreson, R. Elmasri, F. Grandi, C. Jensen, W. Käfer, N. Kline, K. Kulkarni, C. Leung, N. Lorentzos, J. Roddick, A. Segev, M. Soo, S.M. Sripada, **The Temporal Query Language TSQL2**, Kluwer Academic Pub., 1995.

[Snodgrass et al. 95b] R.T. Snodgrass, M.H. Böhlen, C.S. Jensen and A. Steiner, “Adding Valid Time—Part A,” ANSI X3H2-95-485, available at `FTP.cs.arizona.edu/tsql/tsql2/sql3`.

[Snodgrass et al. 98] R.T. Snodgrass, M.H. Böhlen, C.S. Jensen and A. Steiner, “Transitioning Temporal Support in TSQL2 to SQL3,” in **Temporal Databases: Research and Practice**, O. Etzion, S. Jajodia and S. Sripada (eds.), LNCS 1399, Springer Verlag, 150–194, 1998.

[Soo et al. 95] M.D. Soo, C.S. Jensen and R.T. Snodgrass, “An Algebra for TSQL2,” chapter 27 of [Snodgrass et al. 95].

[Steedman 77] M. Steedman, “Verbs, Time and Modality,” *Cognitive Science 2*, 216–234, 1977.

[Sten 52] H. Sten. **Les temps du verbe fini (indicatif) en français moderne**. København. 1952.

[Tansel 87] A.U. Tansel, “A Statistical Interface for Historical Relational Databases,” in *Proceedings of the IEEE International Conference on Data Engineering*, 538–546, Los Angeles, CA, 1987.

[Tansel et al. 94] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R.S. Snodgrass, eds., **Temporal Databases: Theory, Design and Implementation**, Benjamin/Cummings, 1994.

[TDRP 98] O. Etzion, S. Jajodia and S. Sripada (eds.), **Temporal Databases: Research And Practice**, LNCS 1399, Springer Verlag, 1998.

[Terenziani 95] P. Terenziani, “Towards a Causal Ontology Coping with the Temporal Constraints Between Causes and Effects,” in [HCS,95], 847–863, 1995.

[Terenziani & Torasso 95] P. Terenziani and P. Torasso, “Time, Action-Types and Causation: An Integrated Analysis,” *Computational Intelligence* 11(3):529–552, 1995.

[Terenziani 97] P. Terenziani, “Integrating Calendar-Dates and Qualitative Temporal Constraints in the Treatment of Periodic Events,” *IEEE Transactions on Knowledge and Data Engineering* 9(5), 763–783, 1997.

[Terenziani 99] P. Terenziani, “A Relational Algebra for User-Defined Periodicity,” In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, 494–500, Florence, Italy, September 1999.

[Terenziani 01] P. Terenziani “Symbolic User-defined Periodicity in Temporal Relational Databases. To appear in IEEE Transactions on Knowledge and Data Engineering, 2001.

[Tichy 85] P. Tichy, “Do We Need Interval Semantics?,” *Linguistics and Philosophy* 8:263–282, 1985.

[Toman 96] D. Toman, “Point vs. Interval-based Query Languages for Temporal Databases,” in *Proceedings of the ACM Symposium on Principles of Database Systems*, pp. 58–67, June, 1996.

[Toman 98] D. Toman, “Point-based Temporal Extensions of SQL and Their Efficient Implementation,” **Temporal Databases: Research and Practice**, O. Etzion, S. Jajodia and S. Sripada (eds.), LNCS 1399, Springer Verlag, 211–237, 1998.

[Tsichritzis & Lochovsky 82] D. C. Tsichritzis and F. H. Lochovsky, **Data Models**, Prentice-Hall, 1982.

[Tuzhilin & Clifford 95] A. Tuzhilin and J. Clifford, “On Periodicity in Temporal Databases,” *Information Systems* 20(8): 619–639, December 1995.

[Vendler 67] Z. Vendler, “Verbs and times,” in **Linguistics in Philosophy**, Cornell University Press, New York NY, 97–121, 1967.

[Verkuyl 71] H.J. Verkuyl. **On the Compositional Nature of The Aspects**. Dorderect, Riedel, 1971.

[Verkuyl 93] H.J. Verkuyl. *A Theory of Aspectuality.* Cambridge University Press. 1993.

[Vila 94] L. Vila, “A Survey of Temporal Reasoning in Artificial Intelligence,” *AI Communications* 7(1), 4–128, 1994.

[Wu et al. 98] Y. Wu, S. Jajodia and X.S. Wang, “Temporal Database Bibliography Update,” 338–366, in [TDRP, 98].

Appendix 1: Proofs

Property 2. The equivalence property holds for the algebra defined in this paper.

In this appendix, we just sketch some significant part of the proof. We prove Property 2 by proving the following equalities:

Given two standard (atemporal) tables T_1 and T_2, if both T_1 and T_2 are transformed into atelic tables:

$$\forall p\in \mathcal{P}\ (Transform^A{}_P(OP(T_1, T_2)) \ = \ OP^A(Transform^A{}_P\ (T_1), Transform^A{}_P\ (T_2)))$$

If both T_1 and T_2 are transformed into telic tables:

$$\forall I\ (\ Transform^T{}_I(OP(T_1, T_2)) = OP^T(Transform^T{}_I\ (T_1), Transform^T{}_I\ (T_2)))$$

In case we transform the input tables T_1 and T_2 into atelic tables, our algebra corresponds to SQL/Temporal’s one, for which equivalence holds.

Let us consider the case when both T_1 and T_2 are transformed into telic tables. For example, we report the proof for the union ($\cup^T$).

Property 2.1: $\forall I\ (\ Transform^T{}_I\ (T_1 \cup T) = Transform^T{}_I\ (T_1) \cup^T Transform^T{}_I\ (T_2)\ )$
where $Sch(T_1) = Sch(T_2) = T$.

The proof is articulated in two steps. First, we show
$Transform^T{}_I\ (T_1 \cup T_2) \subseteq (Transform^T{}_I\ (T_1) \cup^T\ Transform^T{}_I\ (T_2))$, i.e.,
2.1.1 $\forall t\in Transform^T{}_I\ (T_1 \cup T_2) \Rightarrow t\in (Transform^T{}_I\ (T_1) \cup^T (Transform^T{}_I\ (T_2))$
Then, we prove the opposite containment:
$Transform^T{}_I\ (T_1 \cup T_2) \supseteq (Transform^T{}_I\ (T_1) \cup^T Transform^T{}_I\ (T_2))$, i.e.,
2.1.2 $\forall t\in (Transform^T{}_I\ (T_1) \cup^T Transform^T{}_I\ (T_2)) \Rightarrow t\in Transform^T{}_I\ (T_1 \cup T_2)$

Proof of 2.1.1:
Let t be any tuple such that $t\in Transform^T{}_I\ (T_1 \cup T_2)$. Then, by definition of $Transform^T$, t has validity time t(VT) = S_I, and there is a tuple s such that $s\in (T_1 \cup T_2)$ and s(T) = t(T). By definition of $\cup$, $s\in (T_1 \cup T_2)$ if and only if one of the following cases hold:
(i) $s\in T_1$ and $\neg\exists s'\in T_2$ such that s(T) = s'(T),
(ii) $s\in T_2$ and $\neg\exists s'\in T_1$ such that s(T) = s'(T), or
(iii) $\exists s_1\in T_1$, $\exists s_2\in T_2$ such that s_1(T) = s_2(T) .
In case (i), by definition, $Transform^T{}_I\ (T_1)$ contains a tuple t' such that t'(T) = s(T), and t'(VT) = I. Since (i) holds, $Transform^T{}_I\ (T_2)$ does not contain any tuple t'' such that t''(T) = s(T) = t'(T). Thus, by definition of $\cup^T$, t' belongs to $(Transform^T{}_I\ (T_1) \cup^T Transform^T{}_I\ (T_2))$. Notice that t' is such that t'(T) = s(T) = t(T) and t'(VT) = S_I = t(VT).
Case (ii) is analogous to case (i) above.

In case (iii), by definition, $Transform^{T}_{I}\,(T_1)$ contains a tuple t_1 such that $t_1(\mathrm{T}) = s(\mathrm{T})$, and $t_1(\mathrm{VT}) = I$, and $Transform^{T}_{I}(T_2)$ contains a tuple t_2 such that $t_2(\mathrm{T}) = s(\mathrm{T})$, and $t_2(\mathrm{VT}) = I$. Thus, by definition of $\cup^{T}$, there is a tuple t' that belongs to $(Transform^{T}_{I}\,(T_1) \cup^{T} Transform^{T}_{I}\,(T_2))$ such that $t'(\mathrm{T}) = t_1(\mathrm{T}) = t_2(\mathrm{T}) = s(\mathrm{T}) = t(\mathrm{T})$ and $t'(\mathrm{VT}) = t_1(\mathrm{VT}) \cup t_2(\mathrm{VT}) = I \cup I = I = t(\mathrm{VT})$.♦

Proof of 2.1.2:
Let t be any tuple such that $t \in (Transform^{T}_{I}\,(T_1) \cup^{\mathrm{T}} Transform^{T}_{I}\,(T_2))$. By definition of $\cup^{T}$,
$t \in (Transform^{T}_{I}\,(T_1) \cup^{T} Transform^{T}_{I}\,(T_2))$ if and only if one of the following cases hold:
(i) $\exists s \in Transform^{T}_{I}\,(T_1)$ such that $s(\mathrm{T}) = t(\mathrm{T})$ and $\neg\exists s' \in Transform^{T}_{I}\,(T_2)$ such that $s'(\mathrm{T}) = t(\mathrm{T})$,
(ii) $\exists s \in Transform^{T}_{I}\,(T_2)$ such that $s(\mathrm{T}) = t(\mathrm{T})$ and $\neg\exists s' \in Transform^{T}_{I}\,(T_1)$ such that $s'(\mathrm{T}) = t(\mathrm{T})$, or
(iii) $\exists s_1 \in Transform^{T}_{I}\,(T_1)$ such that $s_1(\mathrm{T}) = t(\mathrm{T})$ and $\exists s_2 \in Transform^{T}_{I}\,(T_2)$ such that $s_2(\mathrm{T}) = t(\mathrm{T})$.

In case (i), notice that, by definition of $Transform^{T}_{I}$, $s(\mathrm{VT}) = I$ and, by definition of $\cup^{T}$, $t(\mathrm{VT}) = s(\mathrm{VT}) = I$. $s \in Transform^{T}_{I}\,(T_1)$ implies that there is a tuple $s_1 \in T_1$ such that $s_1(\mathrm{T}) = s(\mathrm{T})$, and $\neg\exists s' \in Transform^{T}_{I}\,(T_2)$ such that $s'(\mathrm{T}) = t(\mathrm{T})$ implies that there is no tuple $s_2 \in T_2$ such that $s_2(\mathrm{T}) = s'(\mathrm{T}) = t(\mathrm{T})$. Thus, $T_1 \cup T_2$ contains a tuple u such that $u(\mathrm{T}) = s_1(\mathrm{T}) = s(\mathrm{T}) = t(\mathrm{T})$. Thus, $Transform^{T}_{I}\,(T_1 \cup T_2)$ contains a tuple v such that $v(\mathrm{T}) = u(\mathrm{T}) = t(\mathrm{T})$ and $v(\mathrm{VT}) = S_I = t(\mathrm{VT})$.

Case (ii) is analogous to case (i) above.

In case (iii), notice that, by definition of $Transform^{T}_{I}$, $s_1(\mathrm{VT}) = S_I$, $s_2(\mathrm{VT}) = S_I$. Moreover, by definition of $\cup^{\mathrm{T}}$,
$t(\mathrm{VT}) = s_1(\mathrm{VT}) \cup s_2(\mathrm{VT}) = I \cup I = I$. By definition of $Transform^{T}$, $s_1 \in Transform^{T}_{I}\,(T_1, S_I)$ implies that there is a tuple $s_1' \in T_1$ such that $s_1'(\mathrm{T}) = s_1(\mathrm{T})$, and $s_2 \in Transform^{T}_{I}\,(T_2, S_I)$ implies that there is a tuple $s_2' \in T_1$ such that
$s_2'(\mathrm{T}) = s_2(\mathrm{T})$. Thus, by definition of $\cup$, $T_1 \cup T_2$ contains a tuple u such that $u(\mathrm{T}) = s_1'(\mathrm{T}) = s_1'(\mathrm{T}) = t(\mathrm{T}) = s_2(\mathrm{T}) = s_2'(\mathrm{T})$. Thus, $Transform^{T}_{I}\,(T_1 \cup T_2)$ contains a tuple v such that $v(\mathrm{T}) = u(\mathrm{T}) = t(\mathrm{T})$ and $v(\mathrm{VT}) = I = t(\mathrm{VT})$.♦

Appendix 2: Aktionsart Classes in the Linguistic Literature

In this Appendix, we briefly sketch some issues about aktionsart classes and distinctions, from the standpoint of linguistics and computational linguistics, mainly focusing on the telic/atelic dichotomy. Obviously, the discussion below is by no means intended to be complete.

Starting from the 70's, almost all approaches in linguistics and computational linguistics have agreed that natural language propositions can be classified in different *aktionsart classes* (also termed *aspectual classes* [CL 88]) depending on their linguistic behavior and/or semantic properties. Unfortunately, the is no general agreement on which are the classes, and which are the tests to distinguish among them. However, Vendler's distinction between *activities, accomplishments, achievements* and *states* [67] is probably the most influential work about linguistic aktionsart, which is at the bases of most subsequent works, and is a quite standard milestone to evaluate and compare different approaches. Vendler's acktionsart scheme can be grasped intuitively by considering some of the examples Vendler classified under each class

ACTIVITIES: run (around, all over), walk (and walk), swim (along), push (a cart)

ACCOMPLISHMENTS: run a mile, paint a picture, recover from illness

ACHIEVEMENTS: recognize, find, win (the race), start, stop, resume, die

STATES: desire, want, love, dominate

Vendler's achievements denote propositions having an instantaneous character. They capture either the inception or the climax of an act. They cannot in themselves occur over or throughout a temporal stretch. In contrast, accomplishments have a duration intrinsically. They denote happenings which occupy a period of time, and which involve some intrinsic end or conclusion (termed *telos*, i.e., goal). Accomplishments are not "homogeneous": *"in case I wrote a letter in an hour, I did not write it, say, in the first quarter of that hour"* [Vendler 67, page 101]. Activities denote happenings occupying a period of time, this time strech being inherently indefinite. Differently from accomplishments, Vendler's activities are essentially homogeneous (i.e., *"any part of the activity is of the same nature of the whole"* [Vendler 67, page 101]). Finally, states are propositions denoting states of affairs which may endure or persist over stretches of time, and differ from accomplishments and activities in that they *"cannot be qualified as actions at all"* [Vendler 67, page 106].

It is worth noticing that Vendler's approach is not at all the first work considering aktionsart distinctions. Modern linguistics tend to attribute the first aktionsart distinctions to Aristotle. In his *Metaphysics*, Aristotle distinguished between *kineseis* (*telic* facts; accomplishments in Vendler's terminology) and *energeiai* (*atelic* facts; states and activities in Vendler's terminology). Bertinetto [86, pages 93-94] points out that the semantic implications of telicity were explicitly noticed (as regard ancient Latin) by Plinio il Vecchio in his introduction of *Naturalis Historia*. In the Linguistic literature, precursors of the Vendler's typology appeared already in the nineteenth century (see, e.g., Jespersen's two-fold distinction between *conclusive* vs. *nonconclusive* [Jespersen 24]). In 1957, the year Vendler's original article appeared, Garey [57] presented a classification scheme for French verbs under the rubrics telic and atelic. Basically, Vendler's aktionsart classes where also maintained in Mourelatos' taxonomy [78], while Kenny [63] merged all telic propositions (i.e., achievements and accomplishments) into one single class (both approaches used a different terminology to denote classes). Further aktionsart distinctions where introduced by other linguists. For example, Moens and Steedman [88] have extended the telic/atelic dichotomy to instantaneous propositions. In fact, Moens and Steedman distinguished between *culminations* (Vendler's achievements) and *points*, that are instantaneous events not usually viewed as leading to a relevant change in the world (e.g., *to hiccup*). Other aktionsart distinctions can be found, e.g., in the special issue [CL 88].

However, it is important noticing that, despite some relevant differences, almost all systems of aktionsart classes in the linguistic literature consider the telic/atelic dichotomy (besides the above references, consider also, e.g., [Declerk 79, Dahl 81]; the telic/atelic dichotomy has been also expressed using a different terminology: consider, e.g., the *Perfective* vs. *Imperfective*, *Acyclic* vs. *Cyclic*, *Conclusive* vs. *Non-conclusive* dichotomies discussed in [Sten 52], [Bull 60], and [Šabršula 63] respectively). It is also worth noticing that aktionsart distinctions (and, in particular, the telic/atelic distinction) appear to be an intrinsic feature of many different natural languages. For instance, Bertinetto [86] adopted a class scheme very close to Moens and Steedman's one (except for the fact that Bertinetto distinguishes between two sub-classes of states—*permanent* vs. *non-permanent*—and two subclasses of achievements – *reversible* vs. *non-reversible*) to deal with the Italian language. Aktionsart distinctions are used in the analysis of different languages, ranging from German [Klein 74], Dutch [Verkuyl 93] and French [Garey 57] to Kikuyu [Jonson 81], from Japanese [Ikegami 81, Coseriu 79] to Swahili [Mommer 86]. Of course, each language has its peculiar features. For example, Bertinetto [86] noticed that, as regards telicity, English and Japanese are at some extreme points in an ideal scale (and Romance languages are in the middle): English tend to maintain telicity as much as possible, while Japanese easily lose it. For example, in Japanese, one can correctly say [Ikegami 81]

Moyashita keredo, moenakatta

(literally: I burnt it, but it didn't burn)

The telic vs. atelic distinctions are so relevant that, in several cases, natural languages provide complementary pairs of verbal lexemes in order to stress the telic vs. atelic behaviour of semantically very similar propositions. Consider, for instance, English (*eat/eat up*, corresponding to the atelic action of eating and to the telic situation in which one has finished to eat respectively), German (*backen/verbacken*: cooking in a oven vs. finishing the action of cooking), Italian (*dormire/addormentarsi*: to sleep, vs. falling asleep), ancient Latin (*facio/perficio*: to be doing vs. to finish doing). This phenomenon is very relevant especially in Slavic languages where, in many cases, aspectual and aktionsart distinctions are expressed at the morphological level, adding prefixes (e.g., *po, pro, na*; sometimes, also suffixes are used) to verbal lexemes. For example, in Russian, one can say

On pisal pisma	(meaning: He was writing letters, but he didn't finish)
On napisal pisma	(meaning: He wrote the letters – he finished them)
On napisal pismo	(meaning: He wrote a letter – he finished it).

Recently, some approaches in cognitive science, considering also different languages, pointed out that the aktionsart distinctions (and, in particular, the telic/atelic one) play a fundamental role in the acquisition of verbal paradigms by children (see, e.g., [Antinucci & Miller 73, Bloom et al. 80] as regards English, [Bronckart & Sinclair 73] as regards French, [Aksu 78] as regards Turkish).

Within the linguistic literature, many different series of tests have been introduced in order to distinguish among aktionsart classes. Many of them are linguistic tests, involving, e.g., *co-occurrence* with other verbs (e.g., *force, persuade, finish, stop*) or with temporal adverbials (e.g., *in X time, for X time, at* t_X, *from* t_X *to* t_Y), or with aspectual forms such as the *progressive* form, or with specific tenses. For example, Dowty [79] notes that, in

English, only non-states (i) occur in the progressive, (ii) occur as complements of *force* and *persuade*, (iii) occur as imperatives, (iv) co-occur with the adverbs *deliberately* and *carefully*. Semantics tests are more closely related to the temporal semantics of verbs and propositions, looking at the semantic entailment of linguistic propositions. For example, Dowty's [79] criteria to distinguish between telic and atelic facts are reported below:

(a) If *V* is an activity verb, then *x V-ed for y time* entails that at any such time *x V-ed* was true. If *V* is an accomplishment verb, then *x V-ed for y* does not entail that *x V-ed* was true during any time within *y* at all.

(b) If *V* is an activity verb, then *x is (now) V-ing* entails that *x has V-ed*. If *V* is an accomplishment verb, then *x is (now) V-ing* entails *x has not (yet) V-ed*.

Moreover, a debated issue within the linguistic literature concerns the "scope" of the classification. Recent approaches tend to agree that the aktionsart classification does not concern verbs alone (or, even, "typical uses" of verbs), but propositions conveyed by linguistic sentences (see, e.g., [CL 88]). In fact, many linguists noticed that the aktionsart class of a proposition does not only depend on its verb, but also on its *case frame* (and, in particular, its subject and object, if any), and on the temporal adverbials and the aspect of the sentence. For instance, Verkuyl [93] noticed that the the fact that a *noun phrase* in a sentence denote a *"specified quantity"* of objects (or a non-specified quantity) has a deep impact on the telicity of the proposition denoted by the sentence. For instance, *Judith ate a sandwich* is telic, while *Judith ate sandwiches* is atelic. Verkuyl also proposed a compositional approach to derive a logical representation of the meaning of a sentence (and, thus, also its aktionsart properties) from its verb and its nominal constituents. Analogously, Moens and Steedman [88] proposed a compositional approach to determine the aktionsart of a sentence on the basis of its verb, aspect and temporal adverbials. For instance, in Moens and Steedman's model, progressive form naturally applies to activities (termed *processes* in their teminology). Whenever it applies to an accomplishment (*culminated process*), it coerces it to an activity, by stripping out its *culmination*. Thus, for instance, *"Roger ran a mile"* denotes an accomplishment, and in *"Roger was running a mile"*, the progressive form coerces it into an activity. Analogously, the *in* adverbial naturally applies to accomplishments (*"John ate an apple in 2 minutes"*). When applied to achievements (*culminations*), it turns them into accomplishments by involving their *preparatory process* (e.g., *"Laura reached the top in two hours"*); when applied to activities, it adds them a culmination, turning them into accomplishments. For instance, Moens and Steedman notice that *"John ran in four minutes"* is a correct English sentence (denoting an accomplishment) in a context where John habitually runs a particular distance, such as a measured mile.

To conclude this sketch on aktionsart distinctions in linguistics, let us quote Bertinetto [86, page 114-115, translated from Italian] *"Obviously, the substantial agreement between different authors, operating with different methodologies and on different languages, may induce us to think that Aktionsart classifications are a truthful mirroring of the extra-linguistic reality"*.

Finally, it is worth commenting why, in this paper, we chose to focus our attention on the telic/atelic dichotomy only, without considering the other aktionsart distinctions emerged from the linguistic literature. Let us consider, as a reference approach, Vendler's distinction between states, activities, accomplishments and achievements.

Basically, achievements denote punctual telic facts. As we mentioned in the concluding section, we can easily deal with achievements in our approach by extending it with the introduction of a counterpart of *validity time event relations* (i.e. relations having a single time point in their validity time) [Snodgrass et al. 95] into our model.

On the other hand, we deliberately chose to simplify our model throwing away the distinction between states and activities. Basically, our main concern within this paper regards the correct treatment of the temporal semantics of facts. In fact, it seems to us that, while there are obvious differences between states (which are inherently static, even if they may be persistent *"being small"* or not *"being rich"*) and activities (which involve some form of action), the semantic implications they entail are very close. In other words, the meaning of associating a range of time to a state and to an activity is quite similar, so that we think that the point-based semantics can correctly capture the temporal semantics of both classes. In particular, let us consider the upward and downward hereditary properties, which are at the core of the semantic distinctions in this paper. It is usually agreed that the upward hereditary property holds for activities (as for states). For instance, if John ran from 10:10 to 10:20, and from 10:20 to 10:25, then one can correctly conclude that John ran from 10:10 to 10:25. On the other hand, whether the downward hereditary property (also termed *homogeneity* property within the linguistic context) holds or not for activities is an open issue within the Linguistic literature. Many approaches state that downward hereditary also holds for activities, provided that *"relevant moments"* are considered [Heinämäki 78]. On the other hand, other approaches insisted that activities are not homogeneous at all, since, e.g., one can say that John worked from 1:00 to 2:00 even if he stopped, say, at 1:33 and started to work again at 1:37. For instance, this is the position by [Dowty 86] (see the semantic criteria presented in Section 3 of the paper – point b).

However, in most approaches to TDBs, only *exact* temporal information is dealt with (cf., however, [Gadia et al. 92, Dyreson & Snodgrass 93, Koubarakis 93, Brusoni et al. 99]). Thus, we can imagine that, in the case above, one can associate the set of time points spanning from 1:00 to 1:33, and the points from 1:37 to 2:00 to the tuple representing the sentence "John worked". In other words, it seems to us that downward hereditary holds on activities, provided that one describes their validity time in a very accurate way.

Extensions to deal with "relevant moments" would involve the introduction of a probabilistic model to cope with the association of facts to times. This is, of course, a major change from the approach commonly adopted within the TDB literature, which can be worth exploring in our future work.

Appendix 3: Telic and Atelic Facts in Artificial Intelligence (sketch)

In Section 3.4 we sketched an important issue, emerging both from the philosophical and the linguistic literature: facts in the world can be classified into different classes, and the semantics of the association of facts to time depends on the classes of the facts.

In Appendix 2, we described some aktionsart distinctions carried on within the linguistic literature. In Section 2, we also notice that, within the linguistic community, it is widely recognized that the semantics of the association of facts to time depends on the aktionsart classes of the facts. However, it is important to notice that Steedman emphasized that the about distinctions are not about verbs or verb groups, nor even about things that exist in the world, but rather about *descriptions* of the world [Steedman 77, page 217]. Thus, these distinctions *"….. are* *conceptual tools of great usefulness in the philosophy of action, the philosophy of mind, in ontology generally, as well as in linguistics …. "* [Mourelatos 78, page 194].

Since "one of the most crucial problems in any computer system that involves representing the world is the representation of time" [Allen 91, page 341], this issue has had a significant impact on the recent AI literature. In AI, many different techniques have been used in order to model the association of facts to time, such as, e.g.,

- *reification* (see, e.g., [Allen 84, Galton 91] and the criticism in [Bacchus 91])
- *episodic variables/ontological promisquity* (see, e.g., [Schubert & Hwang 89], [Hobbs 95])
- modal *temporal logics* (see, e.g., the survey in [Emerson 90]).

For example, Schubert and Hwang [89] introduced an episodic constant in order to represent explicitly any fact (termed *episode*) in the world. For instance, in Schubert's approach a fact such as "John called Mary from 10 to 12" could be represented as *phone-call*(e_1,*John*,*Mary*), where e_1 is a constant that uniquely identify the fact. Thus, in such an approach, one can easily distinguish between facts of the same type (and with the very same description), even if they occur in meeting, overlapping or equal intervals of time. [7]

Coming back to the core distinction between *telic* and *atelic* facts, it was first taken into account within the Philosophical community, dating back to Aristotle [Aristotle], from whom we derived the terminology. Going forward to recent philosophical approaches, Bach [86] pointed out that telic and atelic facts are somehow two complementary ways of representing reality. In particular, Bach showed that the dichotomy between atelic based view and the telic based view of the facts in the world is just a counterpart of the *mass-nouns* versus *class-nouns* dichotomy. In the same way as one can say that an object is composed by pieces of material (in turn, each piece of material could be conceived as a smaller object, at another level of granularity), a telic fact is composed by atelic ones.

These complementary ways of representing reality have also had a substantial impact on the AI community, where there is a long and still ongoing debate on whether it is better to model reality as a sequence of different *states* (atelic based representation)[8], or as a sequence of different *events* (telic based representation). For instance, McCarthy's Situation Calculus [McCarthy 68] is a typical example of the state based representation, while the Event Calculus [Kowalski & Sergot 86] an example of the event based representation. The discussion of the relative merits of the two approaches would lead us far away the main goals of this paper. However, it is importat to notice that, also in the AI field, many have stressed the fact that the status based and the event based ways to represent reality are complementary, and in many cases one needs a flexible approach in which *both* ways can be adopted (consider, e.g., [Allen 84, Franconi et al. 93, Galton, 91, Terenziani & Torasso 95]). For example, in his seminal approach, Allen [84] distinguished among *states*, *activities* (termed *processes*) and *accomplishments*

---

[7] In our telic tables, we adopt the *interval-based semantics* [Bennet & Partee 72; Dowty 79,86] for the association of a time interval $[i^-,i^+]$ to the fact described by a given tuple. In Schubert's approach, such a semantic is easily modeled by stating that there is an "*episode*" of t which started at i^- and ended at i^+. Analogously, the association of a tuple to a telic element $\{i_1,i_2,...,i_n\}$ can be captured by stating that there are different independent episodes of t, one starting in i_1^- and ending in i_1^+, one starting in i_2^- and ending in i_2^+, etc.

[8] Notice that, as discussed in [Jensen & Snodgrass 96; pp. 323], "*the natural extension of a conventional relation to a temporal relation encodes states instead that events*". In fact, using the point-based sematics (as in BCDM) the database collects a set of snapshots of the *mini-world* [Jensen & Snodgrass 96] it represents. In other words, the mini-world has a status-based representation, since it is represented as a set of *states*, one for each temporal snapshot (time point) in the database.

(termed *events*). In his first-order *reified* logic, Allen introduced three different predicates to associate facts to times, and used an axiomatic approach to model the downward hereditary property of states and the fact that accomplishments can be decomposed into activities. It is also important to remark that, in [Allen 83, 84], the truth of facts (represented by logical predicates) is evaluated over time intervals, and not over time points (i.e., an *interval-based semantics* is adopted). Following Allen's influential approach, many AI approaches chose to adopt time intervals as basic temporal primitives (cf., e.g., the surveys in [Allen 91, Vila 94]).

Moreover, in the last years, the increasing need of sharing knowledge has motivated the appearance of approaches proposing high-level domain-independent *ontologies* (cf., e.g., the discussion in [Guarino 95]). Many of these approaches included (at least) the above distinction between *telic* and *atelic* facts. A relevant example is the ontology devised within the CYC project, a project at MCC in Austin and Palo Alto started in 1989, which aims at encoding "the hundreds of millions of facts and heuristics that comprise human consensus reality" [Lenat & Guha 94]. In such an ontology, they distinguish between *processes* (atelic facts) and *events* (telic facts) and model the fact that "Process is to Events as Stuff is to Individual Objects" [Lenat & Guha 94, pp.187].