Extending Temporal Databases to Deal with Telic/Atelic Medical Data

Paolo Terenziani[1], Richard T. Snodgrass[2], Alessio Bottrighi[1], Mauro Torchio[3], and Gianpaolo Molino[3]

[1] DI, Univ. Piemonte Orientale "A. Avogadro", Via Bellini 25, Alessandria, Italy
terenz@mfn.it, alessio@unipmn.it
[2] Department of Computer Science, University of Arizona, Tucson, AZ 85721, USA
rts@cs.arizona.edu
[3] Lab. Informatica Clinica, Az. Ospedaliera S. G. Battista, C.so Bramante 88, Torino, Italy
{mtorchio, gmolino}@molinette.piemonte.it

Abstract. In the area of Medical Informatics, there is an increasing realization that temporal information plays a crucial role, so that suitable database models and query languages are needed to store and support it. In this paper we show that current approaches developed within the database field have some limitations even from the point of view of the data model, so that an important class of temporal medical data cannot be properly represented. We propose a new three-sorted model and a query language that overcome such limitations.

1 Introduction

Time plays a very important role in many real-world phenomena. For example, in the area of medicine, an explicit management of the time when symptoms took place and clinical actions were taken is needed to model the patients' status (e.g., for diagnostic or therapeutic purposes [12]). Thus several data models used to capture clinical data provide suitable supports to explicitly deal with time (consider, e.g., [18], [13], [14], [7]). Over the last two decades, the database community has devised many different approaches to model the validity time of data (i.e., the time when the data holds [20]). In particular, many temporal extensions to the standard relational model were developed, and more than 2000 papers on temporal databases have published (see the cumulative bibliography in [28] and recent surveys [17], [5], [16], [11]). Recently, the TSQL2 approach has consolidated many years of results into a single "consensus" approach [21], which in revision as SQL/Temporal [22]) has been proposed to the ISO and ANSI standardization committees. Such database approaches are domain-independent, so that they can be profitably exploited also to model temporal data in medical applications. However, recently, some papers pointed out that the lack of specific supports makes the task of managing medical temporal data quite complex. For instance, O'Connor et al. implemented Chronus II, a temporal extension of the standard relational model and query language with specific features to make the treatment of clinical data more natural and efficient [15].

S. Miksch et al. (Eds.): AIME 2005, LNAI 3581, pp. 58 – 66, 2005.

In this paper, we focus on temporal relational models and query languages, showing that current approaches have some limitations from the data model point of view, so that relevant temporal phenomena in the medical field cannot be adequately modeled. We then propose a three-sorted model and a query language that overcome such a limitation.

2 Data Models and Data Semantics

As mentioned, many different database approaches have been devised in order to provide specific support to the treatment of time. Although there are remarkable differences between the alternative approaches, basically all of them adopt the same data semantics: the data in a temporal relation is interpreted as a sequence of states (with each state a conventional relation: a set of tuples) indexed by points in time, with each state independent of every other state (see, e.g., the discussions in [9], [10], [17][22], [27]). We will call such a semantics *point-based*, in accordance with the terminology adopted in artificial intelligence, linguistics and mathematical logic (but not in the database area, where "point-based semantics" has a different interpretation [27], [5] and is often used in relation to the semantics of the *query language* [26]).

It is important to clarify that in this paper we focus on *data semantics*, and we sharply distinguish between *semantics* and *representation language*; our distinction is analogous to the distinction between *concrete* and *abstract* databases emphasized by [6]. For instance, in many approaches, such as SQL/Temporal, TSQL2, TSQL, HQL, and TQuel, and Gadia's [8] Homogeneous Relational Model, a *temporal element* (a set of time intervals) is associated with each temporal tuple (or attribute), but this is only a matter of representation language, while the semantics they adopt is point based [27]. However, a model based on a point-based interpretation of temporal data has severe expressive limitations. In section 3, we will substantiate this claim by considering an example in the medical field. We take TSQL2 as a representative example, but analogous problems arise in the other temporal relational approaches in the DB literature (since all these approaches assume a point-based semantics).

3 Limitations of Data Models Grounded on the Point-Based Semantics

We illustrate these problems with an example, but we stress that the same problem arises whenever data models using the *point-based* semantics are utilized to model a whole class of data (namely, Aristotle's class of *telic* data [1]; see the discussion below).

Let us consider, e.g., drug intravenous infusion (henceforth, "i.v." for short). In some cases, the administration event might stop suddenly (e.g., if the i.v. line falls out) and be resumed immediately. In other cases, two successive i.v. (to be distinguished) of a given drug may be prescribed to the same patient, with no time gap between them. In both cases, the two different i.v. infusions (again, with no temporal gap between them) must be recorded, since the process of restoring a phleboclysis requires medical staff intervention and is costly. On the other hand, notice that the biological effect of the drug is only slightly (if at all) influenced by a short interruption of the i.v. (except in few well-known cases). This is the reason why, at the level of granularity of minutes (that we choose for the whole clinical sample database), we model interruptions as instantaneous events (i.e., the interruption is simply expressed by stating that the i.v. ends on a time granule and re-starts in the next one).

From a technical point of view, if a patient *X* had two i.v. infusions of the drug *Y*, one starting at 10:00 and ending at 10:50, and the other from 10:51 to 11:30 (all extremes included), we cannot say that:

(i) *X* had a (complete) i.v. at time 10:31 (that is, a particular minute; thus, *downward inheritance* [19] does not hold);
(ii) *X* had a one-hour-and-a-half-long i.v. of drug *X* (thus, *upward inheritance* [19] does not hold);
(iii) *X* had just one i.v. (i.e., i.v. events are *countable*, and must be kept distinct one from another)

In accordance with Aristotle [1] and with the linguistic literature, we term ***telic*** *facts* those facts that have and *intrinsic goal* or *culmination*, so that the three above properties do ***not*** hold, and ***atelic*** facts (e.g., *"patient X having a high temperature"*) facts for which all the three implications (i)-(iii) above hold [8]. The importance of properly dealing with telic facts have been widely recognised in many different areas, spanning from artificial intelligence to philosophy, from cognitive science to linguistics [26].

Now, let us use a standard (i.e., point-based) temporal DB model to deal with therapies. For concreteness, we use the bitemporal conceptual data model (BCDM) [10] (which is the model upon which TSQL2 is based [21]), in which the validity time is denoted with sets of time points. (As an aside, even if we chose to use time intervals in the *representation* language, as in Figure 3, the problem discussed below would still occur, due to the point-based semantics, as we'll discuss shortly. Hence, the use of BCDM is not restrictive: the same problem arises for any data model that is based on the point-based semantics.)

For example, let us model the afore-mentioned patient *X*, who is named John. Consider the following temporal relation PHLEBO^A, modeling also the facts that John had an i.v. of drug *Z* from 17:05 to 17:34, that Mary had two i.v. infusions of *Z*, one from 10:40 to 10:55 and the other from 10:56 to 11:34, and finally that Ann had an i.v. from 10:53 to 11:32.

PHLEBO^A

P_code	Drug	VT
John	*Y*	{10:00,10:01,...,10:50,10:51,…,11:30}
John	*Z*	{17:05, 17:06, …., 17:34}
Mary	*Z*	{10:40,…,10:55,10:56,…,11:34}
Ann	*Z*	{10:53, 10:34, …, 11:32}

Fig. 1. Relation PHLEBO^A

This relation captures, among other facts, the fact that the drug *Y* was given by i.v. to John from 10:30 to 11:30. Formally, this semantics can be modeled as a function from time points to the tuples holding over such time points (see Figure 2).

10:00 → <John, *Y*>
10:01 → <John, *Y*>
……

Fig. 2. Point-based semantics of the relation PHLEBO^A in Figure 1

On the other hand, this relation does not capture other relevant information, namely, the fact that there were two distinct i.v. infusions, one ending at 10:50 and another starting at 10:51. Such a loss of information becomes clear and explicit if temporal queries are considered. The most important problems arise, in our opinion, in case of queries involving *upward inheritance* and *countability* of tuples. Again, we will use the TSQL2 query language, just to be concrete, but we stress that such a choice is not restrictive.

(1) Upward inheritance holds on all data models based on point-based semantics. Since the semantics implies the validity of tuples over each point in the validity time, it implies the validity on the whole time interval covering all of them[1]. This is not correct when dealing with telic facts such as i.v. Consider the following query over the relation PHLEBO^A in Figure 1, where a relational table based on point-based semantics is used to model telic facts.

(Q1) *Who had one i.v. of Y lasting more than 1 hour?*

```
SELECT P_CODE
FROM PHLEBO^A (PERIOD) AS P
WHERE CAST(P AS INTERVAL HOUR) > 1
        AND Drug = 'Y'
```

Answer_1: {<John ¦ {10:00, …., 11:30}>}

Since John's two i.v. infusions of *Y* cannot be distinguished at the semantic level, their validity time is "merged together", so that the above tuple is reported as output. Analogous problems when considering qualitative temporal constraints between validity times, such as, e.g., the *"after"* predicate in Q2, following.

(Q2) *Who had an i.v. starting after one of the i.v. infusions of Y to John?*

```
SELECT P2.P_CODE
FROM PHLEBO^A (PERIOD) AS P, P AS P2
WHERE P.P_CODE='John' AND P.Drug='Y'
      AND P PRECEDES P2
```

Answer_2: <John ¦ {17:05, …., 17:34}>

Notice that since the tuples <John ¦ {[10:51,..,11:30]}>, <Mary ¦ {[10:56,…,11:34]}> and <Ann ¦ {[10:53,…,11:32]}> are not reported as output, even if they follow one of John's infusions (the one which ended at 10:50).

---

[1] From the technical point of view, within temporal Databases approaches, upward hereditary is introduced by performing *temporal coalescing* [4] over value-equivalent tuples.

(2) Countability. Since there is no way to distinguish, at the semantic level, temporally contiguous value-equivalent tuples, contiguous telic facts are "merged together", and one loses the correct count. Consider the following query.

(Q3) *How many i.v. did John have?*
```
SELECT COUNT(P)
FROM PHLEBO^A (PERIOD) AS P
WHERE P_CODE='John'
```
Answer_3: 2

In fact, in the point-based semantics, {[10:00,…,10:50], [10:51,…,11:30]} is interpreted as the set of points {10:00,…,11:30}.

It is important to notice that these problems are not related to the *representation* language, but to the underlying (point-based) *semantics*. Indeed, several alternative implementations (*representations*) are possible, each maintaining the same (point-based) semantics [21]. The same consideration also concerns the adoption of first normal form [21], in which each timestamp is restricted to be a period, with timestamps associated with tuples. As long as the underlying semantics is point-based, *each possible representation* of the (telic) event that John had two i.v. infusions of *Y*, one from 10:00 to 10:50 and the other from 10:51 to 11:30, is equivalent to the first tuple in $PHLEBO^A$, and conveys the same content shown in Figure 2, i.e., that John had an i.v. of *Y* in each time point within the whole span of time starting at 10:00 and ending at 11:30.

4 A Three-Sorted Model

Notice once again that the above appear whenever a *telic* event (roughly speaking: an event which behaves as described by points (i)-(iii) in section 3: it has *no downward* and *upward hereditary* properties and it is *countable*) is modeled through a DB data model and query language which are based on the *point-based semantics* [23]. In order to deal with *telic* events (which respect the particular intervals, even if adjacent), a new data model and query language are needed, based on *interval-based semantics*[2].

$PHLEBO^T$

P_code	Drug	VT
John	*Y*	{[10:00-10:50],[10:51,11:30]}
John	Z	{[17:05-17:34]}
Mary	*Z*	{[10:40-10:55],[10:56-11:34]}
Ann	*Z*	{[10:53-11:32]}

Fig. 3. Relation $PHLEBO^T$

[2] This point, initially risen by [2], is now generally accepted within the linguistic and the AI communities (see, e.g., [8]).

Definition. *Interval-based semantics for relational 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 occurs. In this semantics time intervals are atomic primitive entities, in the sense that they cannot be decomposed.

As an example, the relation in Figure 3 shows a telic relation PHLEBO^{T} (i.e., a relation based on interval semantics) modelling our example.

Notice that the difference between PHLEBO^{A} and PHLEBO^{T} is not one of *syntax* (in the latter, time intervals are explicitly represented), but rather one of semantics. If an interval-based semantics is adopted, each interval is interpreted as an atomic (indivisible) one (see Figure 4). Thus, e.g., the tuple <John, *Y* ¦ {[10:00-10:50], [10:51-11:30]} does *not* imply that John had a administration at 10:05.

[10:00-10:50]	→	<John, *Y*>
[10:51-11:30]	→	<John, *Y*>
……		

Fig. 4. Interval-based semantics of the relation PHLEBO^{T} in Figure 3

On the other hand, *atelic* events (e.g., *"patient Y having very high temperature"*; roughly speaking: events for which *downward* and *upward inheritance* hold, and which are not *countable*) are correctly coped with by "standard" point-based-semantics relational approaches. Moreover, in most medical applications, also standard *atemporal* relations (i.e., relations where time has not to be coped with) are very useful. Thus, our extended temporal model consists of relations of three sorts: "standard" *atemporal* relations, *atelic* relations (with the "usual" *point-based semantics*) and *telic* relations (with an *interval-based semantics*).

5 Query Language

The preceding sections focused on extensions to a temporal model to add support for both telic and atelic tables. We now show how these concepts can be added to an SQL-based temporal query language. As we'll see, only a few new constructs are needed. The specifics (such as using TSQL2) 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.

The first change is to support the definition of telic tables (the default is designated as atelic). This can be done with an "`AS TELIC`" clause in the TSQL2 `CREATE TABLE` statement.

For telic queries, we prepend the keyword "TELIC". For example, the three queries Q1, Q2, and Q3 could all be correctly written as TELIC SELECT … Q1 would then return the empty relation, as no single i.v. period was longer than an hour. Q2 would return two infusions for John, one starting at 10:51 and one starting at 17:05, as well as one i.v. for Mary (starting at 10:56) and one for Ann, starting at 10:53. The third query would return a count of 3.

Furthermore, in the queries, *coercion* functions are useful in order to convent tables of the different sorts.

(Q4) Who had one (complete) i.v., while John was having an Y i.v.?

As shown in Section 3, i.v. should be regarded as telic facts. However, when stating *"while John was having an i.v. of Y"* we look inside the fact, *coercing* it into an atelic one. Thus, this query involves two different ways of looking at the tuples in relation PHLEBO^T. First, John's i.v. infusions must be interpreted as atelic facts, since we are not looking for i.v. infusions that occurred during *one* of John's infusions, but, more generally, *while John was having an i.v.* (i.e., we are interested in i.v. infusions occurred during [10:00-11:30]). On the other hand, the i.v. infusions we are asking for must be interpreted as telic facts, since we look for *each complete occurrence* of them which is fully contained in [10:00-11:30]. For example, we want Ann in our output, since she had an i.v. from 10:40 to 10:55, regardless of the fact that she also had another i.v. from 10:56 to 11:34. We thus need more flexibility: although each base relation must be declared as telic or atelic, we need coercion functions (**TELIC** and **ATELIC**) to allow switch from one interpretation to the other at query time.

```
TELIC SELECT P2.P_CODE
FROM PHLEBO^A (ATELIC PERIOD) AS P,
     PHLEBO^A (PERIOD) AS P2
WHERE P.P_CODE='John' AND P.Drug='Y'
      AND P CONTAINS P2
```

6 Conclusions

In this paper, we have argued that current database approaches have some limitations, so that an important class of temporal medical data (i.e., *telic* data) cannot be properly represented, and we have proposed a new three-sorted model and a query language that overcome such limitations. While the data model has been already presented in [26], where we also proposed an extended three-sorted temporal algebra coping with both telic and atelic relations, and with coercion functions, , in this paper we considered the impact of the telic/atelic distinction on medical data, and extended the TSQL2 query language to cope with it.

Before concluding, we think that it is worth remarking that, although in this paper we showed the impact of neglecting the telic/atelic distinction on a specific medical example, problems such as the ones discussed in section 3 arise whenever *value-equivalent* tuples (i.e., tuples which are equal in their data part) concerning telic data have temporal extents that meet or intersect in time. This phenomena can occur in *primitive* relations, such as PHLEBO^A in figure 1, but also, and more frequently, in *derived* relations. For example, projection of a relation on a subset of its attributes (e.g., projecting the PHLEBO^A relation over the *Drug* attribute) usually generates several value-equivalent tuples. Also, switching from a finer to a coarser *temporal granularitiy* in the validity time (e.g., from minutes to hours, or days; consider, e.g., [3]) can originate temporal intersections that where not present in the primitive data.

As regards future work, we envision the possibility of extending also the conceptual level (e.g., the *entity-relationship* model) to properly cope with *telic* (and *atelic*) facts. Moreover, we want to implement our approach and apply in GLARE (GuideLine Acquisition, Representation and Execution), a manager of clinical guidelines which strictly interacts with different databases [24],[25].

Finally, we plan to investigate the impact of considering other semantic features of temporal data (such as, e.g., the ones addressed in [19] or in [3]).

References

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

[2] Bennet, M., Partee, B.: Tense and Discourse Location In Situation Semantics. Indiana University Linguistics Club, Bloomington (1978)

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

[4] Böhlen, M. H. , Snodgrass, R. T., Soo, M.: Coalescing in Temporal Databases. Proc. of the International Conference on Very Large Databases (1996) 180–191

[5] Böhlen, M. H., Busatto, R., Jensen C. S.: Point- Versus Interval-Based Data Models. Proc. of the IEEE Int'l Conference on Data Engineering (1998) 192–200

[6] Chomicki, J.: Temporal Query Languages: A Survey. Proc. of the International Conference on Temporal Logic, Springer-Verlag (LNAI 827) (1994) 506–534

[7] Combi, C.: Modeling Temporal Aspects of Visual and Textual Objects in Multimedia Databases. Proc. TIME'00 (2000) 59-68

[8] Dowty, D.: The effects of the aspectual class on the temporal structure of discourse. Tense and Aspect in Discourse, Linguistics and Phylosophy 9(1) (1986) 37-61

[9] Gadia, S. K.: A homogeneous relational model and query languages for temporal databases. ACM Transactions on Database Systems 13(4) (1988) 418-448

[10] Jensen, C. S., Snodgrass, R. T.: Semantics of Time-varying information. Information Systems 21(4) (1996) 311-352

[11] Jensen, C. S., Snodgrass, R. T.: Temporal Data Management. IEEE Transactions on Knowledge and Data Engineering 11(1) (1999) 36-44

[12] Keravnou, E.T.: Special issue: Temporal Reasoning in Medicine. Artificial Intelligence 8(3) (1996)

[13] Keravnou, E.T.: A Multidimensional and Multigranular Model of Time for Medical Knowledge-Based Systems. J. Intell. Inf. Syst. 13(1-2) (1999)73-120

[14] Nguyen, J. H., Shahar, Y., Tu, S. W., Das, A. K., Musen, M. A.: Integration of Temporal Reasoning and Temporal-Data Maintenance into a Reusable Database Mediator to Answer Abstract, Time-Oriented Queries: The Tzolkin System. J. Intell. Inf. Syst. 13(1-2) (1999) 121-145

[15] O'Connor, M. J., Tu, S. W., Musen, M. A.: The Chronus II Temporal Database Mediator. JAMIA, Symposium Supplement, AMIA Fall Symposium, Isaak S. Kohane ed. (2002) 567-571

[16] Özsoyoglu, G., Snodgrass, R. T.: Temporal and real time databases: a survey, IEEE Transaction On Data and Knowledge Engineering 7(4) (1995) 513-532

[17] Roddick, J. F., Patrick, J. D.: Temporal semantics in information systems: a survey. Information Systems 17(3) (1992) 249-267

[18] Shahar, Y., Musen, M.: Knowledge-based temporal abstraction in clinical domains. Artificial Intelligence in Medicine 8(3) (1996) 267-298

[19] Shoham, Y.: Temporal Logics in AI: Semantical and Ontological Considerations, Artificial Intelligence 33 (1987) 89–104

[20] Snodgrass, R. T., Ahn, I.: Temporal databases. IEEE Computer 19(9), 35-42, (1986)

[21] I. Ahn, G. Ariav, D. Batory, J. Clifford, C. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Kafer, N. Kline, K. Kulkarni, C. Leung, N. Loretzos, J. Roddick, A. Segev, M. Soo, S.M. Sripada: in R. T. Snodgrass (ed.) The Temporal Query Language TSQL2, Kluwer Academic, (1995)

[22] Snodgrass, R. T., Böhlen, M. H., Jensen, C. S., Steiner, A.: Transitioning temporal support in TSQL2 to SQL3, in O. Etzion, S. Jajodia, and S. Sripada (eds.): Temporal Databases: Research and Practice, LNCS 1399, Springer Verlag, 150-194, 1998.

[23] Terenziani. P.: Is point-based semantics always adequate for temporal databases?. Proc. TIME 2000, Cape Breton,Canada IEEE Press (2000) 191-199

[24] Terenziani, P., Molino, G., Torchio, M.: A Modular Approach for Representing and Executing Clinical Guidelines. Artificial Intelligence in Medicine 23 (2001) 249-276

[25] Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G., Correndo, G.: A context-adaptable approach to clinical guidelines. Proc. MEDINFO'04, M. Fieschi et al. (eds), Amsterdam, IOS Press (2004) 169-173

[26] Terenziani, P., Snodgrass, R. T.: Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases: A Treatment of the Telic/Atelic Distinction. IEEE Transactions on Knowledge and Data Engineering 16(5) (2004) 540-551

[27] D. Toman: "Point-based temporal extensions of SQL and their efficient implementation", in O. Etzion, S. Jajodia, and S. Sripada (eds.): Temporal Databases: Research and Practice, LNCS 1399, Springer Verlag (1998) 211-237

[28] Y. Wu, S. Jajodia, and X. S. Wang: “Temporal Database Bibliography Update,” 338-366, in O. Etzion, S. Jajodia, and S. Sripada (eds.): Temporal Databases: research and practice, Lecture Notes in Computer Science, Vol. 1399, Springer Verlag (1998).