A General Approach to Represent and Query Now-Relative Medical Data in Relational Databases

Publisher:

Springer International Publishing

Published version:

DOI:10.1007/978-3-319-19551-3_41

A general approach to represent and query now-relative medical data in relational databases

Luca Anselma[1], Luca Piovesan[1], Abdul Sattar[2], Bela Stantic[2], Paolo Terenziani[3]

[1]Department of Computer Science, University of Turin, Italy
{anselma|piovesan}@di.unito.it
[2] Institute for Integrated and Intelligent Systems, Griffith University, Queensland, Australia
{a.sattar|b.stantic}@griffith.edu.au
[3]DISIT, Institute of Computer Science, Università del Piemonte Orientale, Alessandria, Italy
paolo.terenziani@unipmn.it

Abstract. Now-related temporal data play an important role in the medical context. Current relational temporal database (TDB) approaches are limited since (i) they (implicitly) assume that the span of time occurring between the time when facts change in the world and the time when the changes are recorded in the database is exactly known, and (ii) do not explicitly provide an extended relational algebra to query now-related data. We propose an approach that, widely adopting AI symbolic manipulation techniques, overcomes the above limitations.

Keywords. Temporal relational databases, now-related data, temporal algebra.

1 Introduction

Most clinical data are naturally temporal. To be meaningfully interpreted, patients' symptoms, laboratory test results and, in general, all clinical data must be paired with the time in which they hold (called *valid time* henceforth). The research about temporal data has demonstrated that designing, querying and modifying time-varying *relational* tables requires a new set of techniques [1]. In the medical area, several TDB approaches have been devised. For instance, Chronus II [2] has provided an implementation of a subset of the "consensus" approach TSQL2 [3] (which is the basis of the recent SQL:2011 standard), and Das and Musen have focused on temporal indeterminacy [4]. In the recent years, we have extended such a basic core of results to face with the telic/atelic distinction [5], periodically repeated data [6], and proposal vetting [7], proposing the adoption of AI symbolic manipulation techniques in the TDB context. In this paper, we continue such a line of research, facing "now-related" data, to cope with data such as "*John is in the Intensive Care Unit (ICU henceforth) from January 10 to now*". We call valid-time "***now-related***" those facts (tuples) starting in the past and still *valid* until the current time, as in John's example. Such data are very frequent in the medical context, where specific attention has to be devoted to patient symptoms, treatments, measurements *holding at the current time*. Recent TDB approaches have identified four different ways to implement "now" in a standard relational database, using (i) NULL, (ii) the smallest timestamp (MIN approach), (iii) the largest timestamp (MAX approach), or (iv) degenerate zero-point intervals (POINT approach) (see, e.g., [8]). However, such approaches have three main limitations: (1) they propose models to store data, but (except (iv)) no algebra to query them, (2) implicitly or explicitly follow the semantics by Clifford et al. [9], in which "now-related" tuples are interpreted assuming that the span of time occurring between (the time of) a change of the world and (the time of) the database update (henceforth called ***latency***) is ***exactly known***, and (3) they cannot cope with bounds on the future persistence of now-related facts. In this paper, we propose a new TDB approach overcoming the above limitations.

2 Now-related facts and latency of updates

All TDB approaches are (explicitly or implicitly) based on Clifford et al.'s semantics [9]; most of them assume that all "current" data remain current unless they are modified, which corresponds to assuming latency equal to zero. This is a severe limitation, since *generalized relations* cannot be treated [10]. The effect of such an assumption can be shown on Example 2 below (asserted at time 14).

Example 2. John is hospitalized in ICU from January 10 to NOW.

On January 14 (the time of assertion), we are certain that John has been in ICU on January 10, 11, 12 and 13 (homogeneously with treatment of 'now' in transaction time in BCDM [11], we assume that NOW is excluded. Notice, however, that our approach is mostly independent of such a choice). Possibly, John may stay in ICU all January 14, and on the 15, and so on, but this is not certain. Then, let us look at the same information two days after, i.e., on January 16, supposing that no modification has been done in the TDB. Clearly, if latency were known, the fact that the TDB has not been changed would provide us an additional knowledge. For instance, if the latency is 0, we are certain that John was in ICU also on January 14 and 15; if latency is 1 (i.e., facts are recorded one day after they happen in the real world), we are only certain that John has been in ICU also on January 14. However, in the more general case in which latency is unknown (which is more realistic in several medical domains), the valid time of now-related tuples depends only on the time when the now-related fact is asserted (henceforth called *assertion time*), and it is independent of the value of NOW. Indeed, if latency is unknown, the fact that the TDB has not been changed until January 16 (or, in general, any time *t* greater than 14) does not provide any additional information. Maybe John has been dismissed on January 14, but this fact has not been recorded yet (e.g., due to a long-term strike of data entry clerks). In general, if no assumption can be made on when changes in the modeled world are recorded in the TDB the (intended) meaning of ***"the fact f holds from t_{start} to NOW"***, ***asserted at time t_a*** (i.e., NOW=t_a when the fact is asserted) is that *f* holds at each time unit from *start* to t_a (excluded), and it will end sometime in the future (i.e., some time after t_a). In other words, the semantics of NOW (with unknown latency) involves ***temporal indeterminacy in the future*** with respect to the ***assertion time*** t_a (notice that assertion time may differ from the time of insertion of the fact in the DB –i.e., from *transaction time*– and is related with Combi and Montanari's *availability time* [12]). In the following, we provide a relational model covering such an intuition.

3 1NF Relational Data Model

In this section, we propose a compact ***1NF representation*** for now-related tuples, considering valid time, coping with unknown latency and with now-bounded tuples.

Definition: pn-tuple and pn-relation. Given a schema $(A_1, \ldots, A_n)$ where each A_i represents a non-temporal attribute on the domain D_i, a pn-relation r^{pn} is an instance of the schema $(A_1, \ldots, A_n | VTs, VTa, VTe)$ defined over the domain $D_1 \times \ldots \times D_n \times T^C \times T^C \times T^C$, where T^C is the domain of *chronons* [3]. For each instance of the schema, $VTs \leq VTa \leq VTe$. Each tuple $x = (a_1, \ldots, a_n | v_s, v_a, v_e) \in r^{pn}$ is termed a pn-tuple ("pn" stands for possibly now-related).

Intuitively speaking, and considering a valid-time now-related tuple, VTs represents the starting time of valid time, VTa the assertion time, and VTe the future bound for 'now' (the value c_{max} –the maximum possible time in T^C– is used in case no bound has to be modelled, see first row of Table 1). Example 3, at the granularity of days, can be expressed in our model as shown by the second row of Table 1. Notice that, in our representation, time intervals are closed to the left and open to the right.

Example 3. A patient that reaches the emergency department (henceforth ER) can be kept under observation for a maximum period of two days. Afterwards, (s)he must be moved to another ward. Tom was hospitalized in the ER on April 4. Such fact was asserted on day April 5.

Intuitively speaking, the second row of Table 1 represents the fact that we are certain that Tom is in the ER on April 4, and will possibly stay there one more day (until April 6, excluded). In our model, tuples that are not now-related can still be represented, using the convention introduced in Property 1.

Table 1. Pn-relation representation of Examples 2, 3 and 4.

Patient	**Ward**	**VTs**	**VTa**	**VTe**
John	ICU	Jan 10	Jan 14	c_{max}
Tom	ER	Apr 4	Apr 5	Apr 6
Bill	Cardiac Surgery	Aug 16	Aug 31	Aug 31

Property 1: consistent extension (wrt TSQL2). Any not now-related tuple can be easily represented as a special case of the above representation, in which $VTa = VTe$.

Thus, pn-relations can include heterogeneous types of tuples, in the sense that any of them, independently of the others, may be now-related or not. For instance, Table 1 represents both the now-related facts of Examples 2 and 3 (first and second rows) and the not now-related fact that Bill has been hospitalized in the Cardiac Surgery ward from August 16 to 30 (third row).

4 Relational Algebra

Our representation models temporal indeterminacy in a compact and implicit way. The interval $[VTs, VTa)$ is the span of time in which the fact certainly occurs, while $[VTa, VTe)$ is the span of future time in which the fact might hold (resembling Das and Musen's *intervals of uncertainty* [4]). Thus, for instance, the first row of Table 1 is a *compact representation* of the fact that John is in ICU from 10 to 13, *or* from 10 to 14, *or* … *or* from 10 to c_{max}. We provide a temporal extension to Codd's relational algebra operators in such a way that our operators *directly operate on our implicit 1NF representation*, but are ***consistent*** with such an underlying *semantics*. Additionally, our temporal relational operators are ***reducible*** to TSQL2 ones (which, in turn, are reducible to standard Codd's operators). Reducibility grants for the interoperability of our approach to TSQL2-based ones, and with standard DBMS. For the sake of brevity, only Cartesian product is reported. We follow the TSQL2 notation; x[X] represents the value of attribute X in the tuple x.

Definition (Cartesian product). Given two pn-relations r^{pn} and s^{pn} defined on the schemas R: (A_1, …, A_n | *VTs, VTa, VTe*) and S: (B_1, …, B_m | *VTs, VTa, VTe*) respectively (where A_1, …, A_n and B_1, …, B_m represent the non-temporal attributes), the Cartesian product $r^{pn} \times^{pn} s^{pn}$ has schema (A_1, …, A_n , B_1, …, B_m | *VTs, VTa, VTe*) and is defined as follows:

$$
\begin{aligned}
r^{pn} \times^{pn} s^{pn} = \{ & x \setminus \exists\, x1 \in r^{pn} \wedge \exists\, x2 \in s^{pn} \wedge \\
& x[A_1, \ldots, A_n] = x1[A_1, \ldots, A_n] \wedge x[B_1, \ldots, B_m] = x2[B_1, \ldots, B_m] \wedge \\
& x[VTs] = max(x1[VTs], x2[VTs]) \wedge \\
& x[VTa] = max(min(x1[VTa], x2[VTa]), x[VTs]) \wedge \\
& x[VTe] = min(x1[VTe], x2[VTe]) \wedge x[VTs] < x[VTe]\}.
\end{aligned}
$$

Cartesian product manages the non-temporal attributes $A_1, \ldots, A_n, B_1, \ldots, B_m$ in a standard (i.e., Codd's) way and it evaluates the intersection of the "certain" (i.e., [x1[Vts],x1[VTa]) ∩ [x2[Vts],x2[VTa])) and "possible" times of the paired tuples.

5 Conclusions and future work

Now-related temporal data play an important role in the medical context, where specific attention is devoted to patient symptoms, treatments, measurements holding at the current time. Current relational approaches to now-related data assume that the "*latency*" of updates is known and do not explicitly provide an extended relational algebra to query them (except the POINT approach [8]). We propose an approach that, adopting AI techniques, overcomes such limitations, and we analyze its properties (reducibility). In our future work, we aim at extending our approach to cope also with *transaction time*, considering also cases in which the *latency* of updates is *known*. Also, in this paper, we considered only the temporal indeterminacy derived from now-related tuples, and we want to extend it to deal with more general cases (as, e.g., in [9], where, notably, temporal indeterminacy is ***not*** used to cover the semantics of "now"). Finally, a major extension would consist in the addition of suitable AI-based mechanisms to deal with persistence. Indeed, in this paper, we suppose that no additional knowledge is available with respect to the facts in the TDB, so that each of them can persist from the assertion time to a future bound (or, if it is missing, forever). This resembles McCarthy's inertia principle [13], which was extended by McDermott [14] considering the typical lifetime of facts. Recent AI approaches have investigated a knowledge-based analysis of persistence. In particular, considering the medical domain, Shahar has studied persistence in the general context of data interpolation, considering both forward and backward persistence, and stressing the fact that it depends on the concepts, concepts' values, and even context [15].

Acknowledgments

The work was partially supported by Compagnia di San Paolo in the Ginseng project.

References

1. Das AK, Musen MA (1997) A foundational model of time for heterogeneous clinical databases. Proc AMIA Annu Fall Symp 106–110
2. O'Connor MJ, Tu SW, Musen MA (2002) The Chronus II temporal database mediator. Proc AMIA Symp 567–571
3. Snodgrass RT (1995) The TSQL2 temporal query language. Kluwer
4. Das AK, Musen MA (1994) A temporal query system for protocol-directed decision support. Methods Inf Med 33:358–370
5. Terenziani P, Snodgrass RT, Bottrighi A, Torchio M, Molino G (2007) Extending temporal databases to deal with telic/atelic medical data. Artif Intell Med 39:113–126
6. Stantic B, Terenziani P, Governatori G, Bottrighi A, Sattar A (2012) An implicit approach to deal with periodically repeated medical data. Artif Intell Med 55:149–162
7. Anselma L, Bottrighi A, Montani S, Terenziani P (2013) Managing proposals and evaluations of updates to medical knowledge: theory and applications. J Biomed Inform 46:363–376
8. Stantic B, Sattar A, Terenziani P (2008) The POINT approach to represent now in bitemporal databases. Journal of Intelligent Information Systems 32:297–323
9. Clifford J, Isakowitz T, Dyreson C, Jensen CS, Snodgrass RT (1997) On the Semantics of "Now" in Databases. ACM Transactions on Database Systems 22:171–214
10. Jensen CS, Snodgrass R (1994) Temporal specialization and generalization. IEEE Transactions on Knowledge and Data Engineering 6:954–974
11. Jensen CS, Snodgrass RT (1996) Semantics of Time-Varying Information. INFORMATION SYSTEMS 21:311–352
12. Combi C, Montanari A (2001) Data Models with Multiple Temporal Dimensions: Completing the Picture. In: Dittrich KR, Geppert A, Norrie MC (eds) Advanced Information Systems Engineering, 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001, Proceedings. Springer, pp 187–202
13. McCarthy J (1986) Applications of Circumscription to Formalizing Common-Sense Knowledge. Artif Intell 28:89–116
14. McDermott DV (1982) A Temporal Logic for Reasoning About Processes and Plans. Cognitive Science 6:101–155
15. Shahar Y (1999) Knowledge-based temporal interpolation. J Exp Theor Artif Intell 11:123–144