Inferred Validity of Transaction-Time Data

Cristina De Castro **Maria Rita Scalas**

C.I.O.C.-C.N.R. and Dipartimento di Elettronica, Informatica e Sistemistica

Università di Bologna Viale Risorgimento 2, I-40136 Bologna, Italy

Tel. + 39 (51) 644.{3542,3544} Fax: +39 (51) 644.3540

E-mail: {cdecastro,mrscalas}@deis.unibo.it

Abstract:
Temporal databases can support at least two kinds of independent time dimensions: *transaction-time*, which tells when an event is recorded in a database, and *valid-time*, which tells when an event occurs, occurred or is expected to occur in the real world. According to the semantics of transaction-time, when data are retrieved from a transaction-time relation, the only temporal information that can be associated to such data is the time when they were stored, updated or deleted. No conjecture on their validity in the real world is straightforward. Nevertheless, when transaction-time data are used, a validity is implicitly assigned to them. In this paper we propose three possible interpretations of the validity of transaction-time data. Such proposals can be useful in a temporal heterogeneous environment, where relations of different temporal format must interoperate: as a matter of fact, the proposed solutions allow a conversion of temporal data to the bitemporal format, and thus they provide a common format for the execution of the operations. All the three proposed solutions assume that data can be considered valid at least since they were stored. Such semantics is intrinsic of transaction-time DBs, which are accordingly named *historical*, thus we can say that, when data validity is not specified, the knowledge actually stored in the database is taken into account. The second assumption is that any conjecture on the effective beginning of the validity of data preceding their insertion time would create false information and must thus be avoided. These criteria fix the beginning of the inferred validity. The end of validity can be defined in three distinct ways, depending on how much it can span along the valid-time axis. A comparison among the three types of inference is provided at the end, on the basis of the possible use of such data, because the inferred validity can be used for the translation of data both to the valid-time or the bitemporal format. The differences among the three inferences look remarkable.

Keywords: Transaction-Time, Valid-Time, Inferred Valid-Time Extension

1. Introduction and Notation

In current bibliography on temporal databases there is a common agreement for the support of at least two kinds of time dimensions: *transaction-time*, which tells when an event is recorded in a database, and *valid-time*, which tells when an event occurs, occurred or is expected to occur in the real world [Soo91,TSC+93]. Further proposals include the temporal dimension *event-time* [KC93], which allows the distinction between retroactive and proactive updates, impossible with transaction and valid-time only. In this paper we are concerned with the possible deductions that can be made when only transaction-time is supported and thus, for the beginning, we do not take into account event time. If we consider transaction- and valid-time, according to the temporal dimensions they support, temporal databases can be classified as *monotemporal* (transaction- or valid-time) , *bitemporal* or *snapshot* [JCE+94]. Transaction-time DBs record all the versions of data inserted, deleted or updated in successive transactions (current and non current versions). The temporal information of a transaction-time relation concerns the time when data are recorded, updated or deleted from the database. In this sense, transaction-time is the *database time*. Valid-time DBs maintain the most recently inserted versions of data, each relative to a distinct valid-time interval (current versions only). The temporal information of a valid-time relation concerns the time when events actually happen in the real world. In this sense, valid-time is the *time of the miniworld* to represent. Bitemporal DBs support both transaction and valid-time and thus maintain all the valid-time versions recorded in successive transactions (current and non current versions). Snapshot DBs do not support time: they maintain only the most recently inserted (current) version.

When data are retrieved from a database where their validity is not represented, the user implicitly assigns them a validity interval. For instance, when you get the latest version of the telephone book, you can use the telephone numbers and thus you consider them valid. This process is, of course, risky, but reflects commonly used deductions made on every type of available data. In this paper we consider a transaction-time database and propose three distinct ways for inferring data validity from their represented transaction-time. Even if transaction- and valid-time are independent (orthogonal) dimensions, the above considerations justify this type of inference.

Not only the proposed methods give criteria for making inferences on the validity of data, but also they allow the conversion of data to the bitemporal format. This use is necessary in a temporal heterogeneous environment, where the interoperability is required of relations of different temporal format [DGS94,DGS95].

The temporal representation adopted in this paper is the *Bitemporal Conceptual Data Model* (BCDM in [JCE+94]), where time-stamps are represented by *Temporal Elements*. In the following, we provide a concise description of the BCDM formalism and of the adopted notations.

Time is represented by means of *temporal elements*, which consist of sets of *chronons*. As in [JCE+94] *a chronon is a non-decomposable time interval of some fixed, minimal duration*. A duration *p* represents the chosen granularity of time.

A (bi)temporal chronon is an ordered pair of unitary-length chronons, one relative to transaction-time, the other to valid-time: for instance (t_i, t_j) is a bitemporal chronon, where t_i denotes transaction-time, and t_j denotes valid-time. Bitemporal elements are sets of bitemporal chronons of the type $\tau_b = \{(t_i, t_j), ... , (t_l, t_m)\}$. In general, the symbol τ_X denotes a transaction-time (τ_t), valid-time (τ_v) or bitemporal (τ_b) element.

A monotemporal element can always be represented by the union of disjoint component subsets, each represented by its endpoints (IN, OUT for transaction-time, FROM, TO for valid-time) and containing contiguous chronons only: every chronon t_j, such that $t_{mi} \le t_j \le t_{ni}$, belongs to the component subset $\{t_{mi} .. t_{ni}\}$. For instance, the general representation of a transaction-time temporal element is $\tau_t = \cup_i \tau_{ti} = \cup_i \{t_{mi} .. t_{ni}\}$.

The paper is organized as follows: in section 2 we present the three inferences, discuss the criteria on which they are based and provide some examples. In section 3 we carry on the discussion and focus the attention on the different results obtained when the data produced by each inference are selected at distinct transaction-time instants.

2. Three Inferences on the validity of transaction-time data

In this section we present the three different inferences, named *Square Inference*, *Stripe Inference* and *L-Shaped Inference* respectively, and discuss the criteria on which each of them is based. A comparison among the three is carried on. The common assumption concerns the beginning of validity of data: transaction-time (historical) data can be considered valid from the instant they were recorded. Probably such validity precedes this instant, but, in the absence of further information, it would be absolutely unsafe and arbitrary to make further conjectures. The difference among the three solutions is in the extent data validity is allowed to span along the valid-time axis.

The following notation is adopted: if the non-temporal attributes are denoted by r and $\oplus$ denotes the operation of tuple concatenation, a version of an object along the temporal dimension X can be expressed as $r_X = r \oplus (\tau_X)$.

- Transaction-time semi-axis: $\{T_0 .. T_\infty\}$
- Valid-time semi-axis: $\{t_0 .. t_\infty\}$
- Current transaction-time: T_{now}
- If r_t is a transaction-time record, r_b' , r_b'' and r_b''' will denote the results in the bitemporal format obtained by using the first, second and third inference made on the validity of r_t.

The transaction-time relation T-Employee in Tab.1 will be used in the examples. For the sake of simplicity, in the examples the transaction-time temporal elements have a single component. The granularity of time chosen for the examples is one year, thus, for instance, {90 .. 92} starts at the beginning of 1990 and finishes at the end of 1992.

2.1. Square Inference

The criterion on which the Square Inference is based is that when data are retrieved from a transaction-time relation they can be considered valid *no less and no more than in their transaction-time interval*. This type of inference is represented in Fig.1.

The Square-inferred valid-time pertinence equals the transaction-time temporal element of each record: the transaction-time record $r \oplus (\tau_t)$ is thus transformed to the bitemporal format as follows:

$$r \oplus (\tau_t) \rightarrow r \oplus (\tau_t) \oplus (\tau_v) \qquad \text{where the inferred validity is:} \qquad \tau_v \equiv^{def} \tau_t$$

If τ_t is the union of disjoint intervals $\tau_t = \cup_i \tau_{ti}$, the above definition must be applied to each component transaction-time temporal element τ_{ti}:

$$r \oplus \{\cup_i \tau_{ti}\} \rightarrow r \oplus (\cup_i \tau_{ti}) \oplus (\cup_i \tau_{vi}) \qquad \text{where} \qquad \tau_{vi} \equiv^{def} \tau_{ti}$$

2.2. Stripe Inference

The criterion on which the Stripe-Inference is based is that when data are retrieved from a transaction-time relation they can be considered valid since they were stored and *indefinitely valid in their transaction-time interval*. The Stripe inference reconstructs the bitemporal pertinence of each record *taking into account that every record was current before it was updated*. Therefore, when a record had not been archived yet, it could be considered undefinitely valid. This type of inference is represented in Fig.2.

The Stripe-inferred valid-time pertinence spans the whole valid-time axis starting from the minimum chronon of the transaction-time temporal element of each record: the transaction-time record $r\oplus(\tau_t)$ is thus transformed as follows:

$$r \oplus (\tau_t) \rightarrow r \oplus (\tau_t) \oplus (\tau_v) \qquad \text{where} \qquad \tau_v \equiv^{def} \{\min \{\tau_t\} .. t_\infty\}$$

Again, if $\tau_t = \cup_i \tau_{ti}$, the above definition must be applied to each τ_{ti}:

$$r \oplus (\cup_i \tau_{ti}) \rightarrow r \oplus (\cup_i \tau_{ti}) \oplus (\cup_i \tau_{vi}) \qquad \text{where} \qquad \tau_{vi} \equiv^{def} \{\min \{\tau_{ti}\} .. t_\infty\}$$

2.3. L-Shaped inference

A transaction-time tuple $r \oplus (\tau_t)$ is said to be *current* if $T_{now} \in \tau_t$ (in this case $OUT = T_\infty$); it is said to be *archived* if $T_{now} \geq \min\{\tau_t\}+1$ (in this case $OUT < T_\infty$). A current tuple $r \oplus (\{IN .. T_\infty\})$ (e.g. r_3 in Fig.3) can be considered valid since it

NAME	JOB	SALARY	τ_t
Ann	Engineer	2800	{85 .. 90}
Ann	Manager	3000	{91 .. T_∞}
John	Engineer	1500	{90 .. 92}
John	Engineer	2000	{93 .. T_∞}

Table 1: transaction-time relation T-Employee

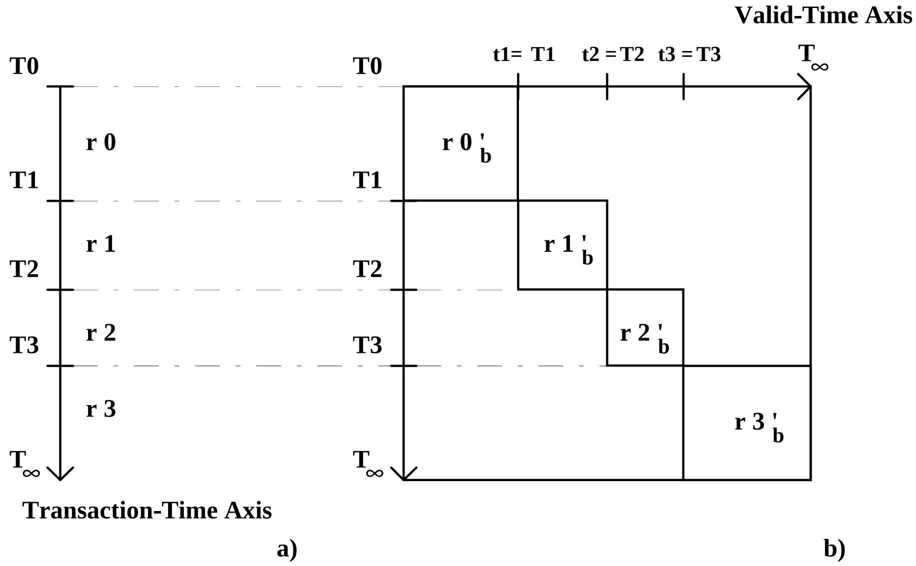

Figure 1: (a) transaction-time pertinence; (b) corresponding Square-inferred validity

NAME	JOB	SALARY	$\tau_b = \tau_t \times \tau_v$
Ann	Engineer	2800	{85 .. 90} × {85 .. 90}
Ann	Manager	3000	{91 .. T_∞} × {91 .. t_∞}
John	Engineer	1500	{90 .. 92} × {90 .. 92}
John	Engineer	2000	{93 .. T_∞} × {93 .. t_∞}

Table 2: Square-inferred validity of T-Employee

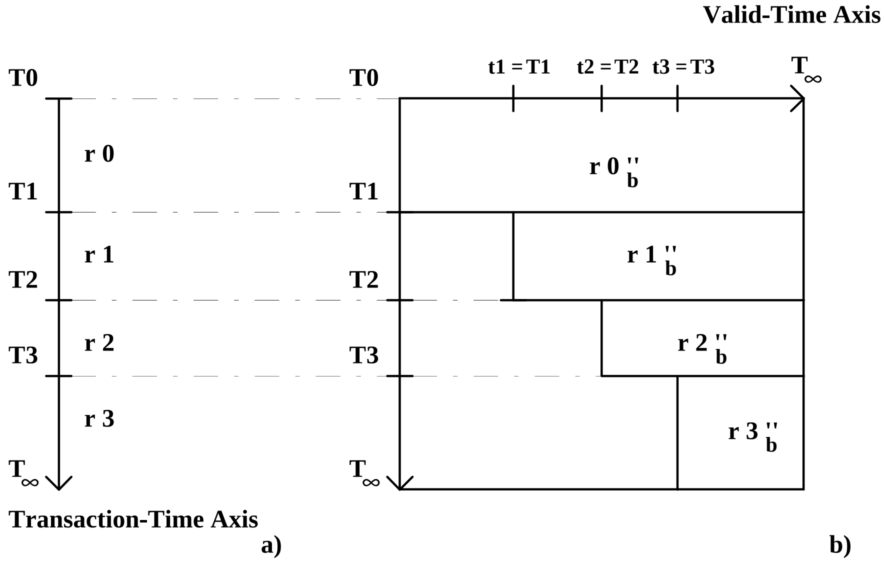

Figure 2: (a) transaction-time pertinence; (b) corresponding Stripe-inferred validity

NAME	JOB	SALARY	$\tau_t \times \tau_v$
Ann	**Engineer**	**2800**	**$\{85 .. 90\} \times \{85 .. t_\infty\}$**
Ann	**Manager**	**3000**	**$\{91 .. T_\infty\} \times \{91 .. t_\infty\}$**
John	**Engineer**	**1500**	**$\{90 .. 92\} \times \{90 .. t_\infty\}$**
John	**Engineer**	**2000**	**$\{93 .. T_\infty\} \times \{93 .. t_\infty\}$**

Table 3: Stripe-inferred validity of T-Employee

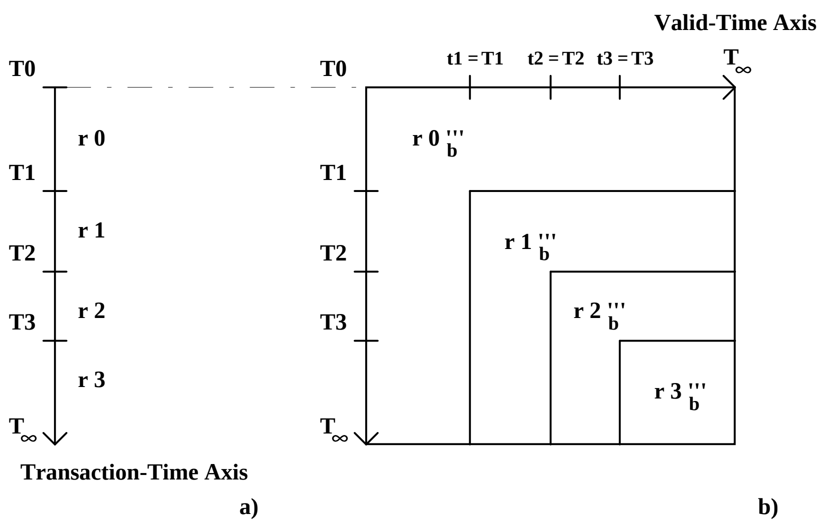

Figure 3: (a) transaction-time pertinence; (b) corresponding L-shaped inferred validity

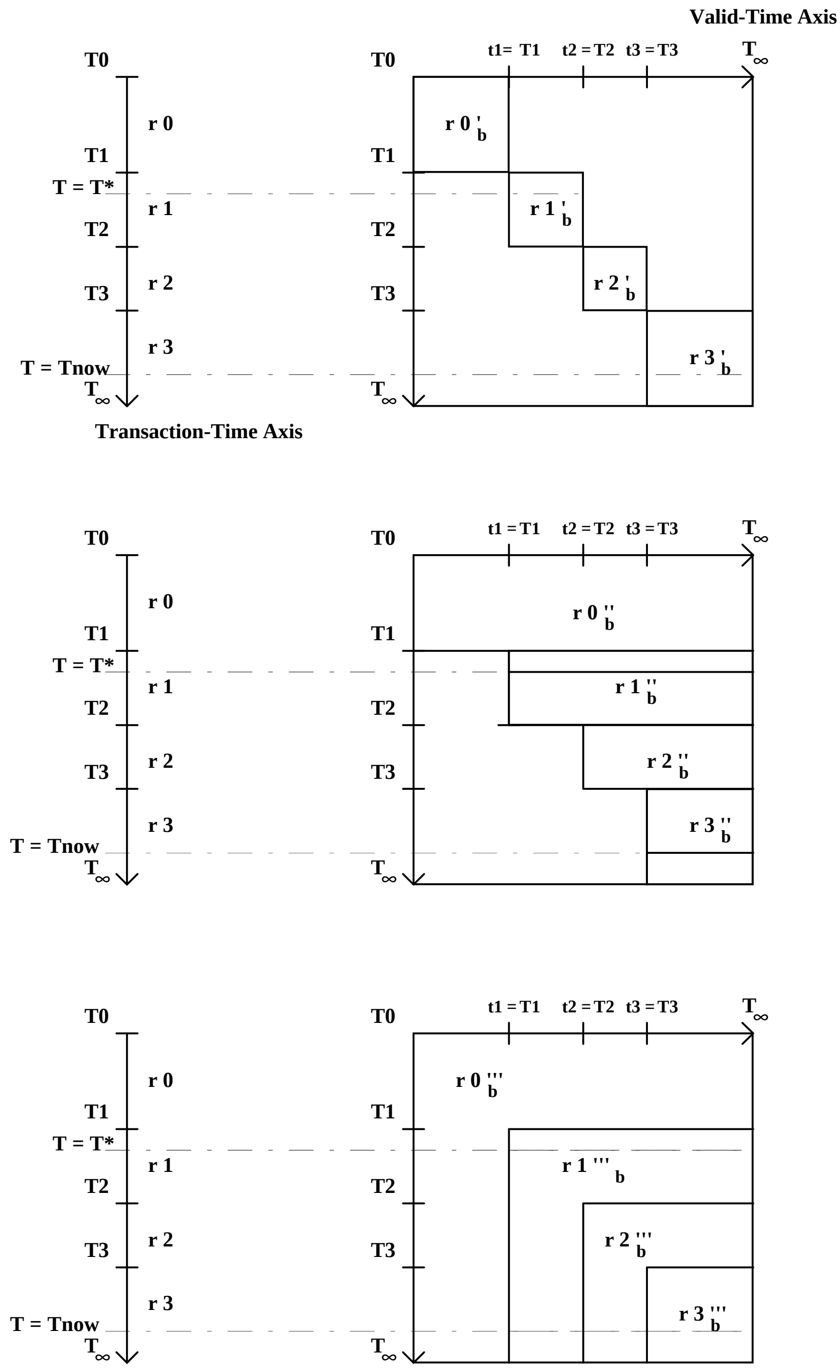

Figure 4: projection along the valid-time axis of the inferred validity at $T = T^* < T_\infty$ and at $T = T_{now}$

was stored and indefinitely valid, since it cannot be forecasted if an update transaction would ever occur and archive such tuple. *An archived tuple* r $\oplus$ ({IN .. OUT}) *can be considered as created by a transaction with effect in* [IN .. T_∞) $\times$ [IN .. T_∞) *and modified by a successive transaction with effect in* [OUT .. T_∞) $\times$ [OUT .. T_∞). The update transaction cuts the initial time pertinence of the tuple to an ``L-shaped'' region (e.g. r_{0b}''', r_{1b}''', r_{2b}''' in Fig.3 and their corresponding ones). This inference criterion was proposed in [DGS93] and fits the bitemporal view of a *diagonal user* (for the concept of *user* see [BG93]).

As in the Stripe-inferred case, the L-shaped inferred valid-time pertinence spans the whole valid-time axis starting from the minimum chronon of the transaction-time temporal element of each record; The difference is that this second criterion takes into account that, before a record was updated *it were unknown if it would have been updated and when*. As a consequence, before the update, not only could the valid-time interval be interpreted as covering the whole valid-time axis from the time IN, but the same hold for the transaction-time pertinence of the record itself (OUT = T_∞ before the update). The transaction-time record r $\oplus$ (τ_t) is transformed as follows:

$r \oplus (\tau_t) \rightarrow r \oplus (\tau_b)$,

where $\tau_b \equiv^{def} \{\min\{\tau_t\} \,..\, T_\infty\} \times \{\min\{\tau_t\} \,..\, t_\infty\}$ - $\{\max\{\tau_t\}+1 \,..\, T_\infty\} \times \{\max\{\tau_t\}+1 \,..\, t_\infty\}$

When $r \oplus (\tau_t)$ is current, $\{\max\{\tau_t\}+1 \,..\, T_\infty\} \times \{\max\{\tau_t\}+1 \,..\, t_\infty\} = \emptyset$

Again, if $\tau_t = \cup_i \tau_{ti}$, the above definition must be applied to each τ_{ti}:

$r \oplus (\cup_i \tau_{ti}) \rightarrow r \oplus (\cup_i \tau_{bi})$

where $\tau_{bi} \equiv^{def} \{\min\{\tau_{ti}\} \,..\, T_\infty\} \times \{\min\{\tau_{ti}\} \,..\, t_\infty\}$ - $\{\max\{\tau_{ti}\}+1 \,..\, T_\infty\} \times \{\max\{\tau_{ti}\}+1 \,..\, t_\infty\}$

3. Further Discussion and Conclusions

In this paper we have considered how data validity can be inferred in transaction-time databases. We proposed three distinct solutions: the Square Inference, the Stripe Inference and the L-Shaped Inference. The Square Inference reflects only the knowledge of the database: data are valid within their transaction-time interval. The Stripe-Inference takes into account that archived data were current before they were updated and, before the update, data could be considered indefinitely valid. A further deduction is made in the L-shaped solution: a portion of data is considered as still current.

As far as the use of the inferred data is concerned, a distinction must be made between their use in the bitemporal or in the valid-time format. It can be noticed that the Square Inference can be used not only for the bitemporal view of data, but also for the valid-time view, since the Square-inferred valid-time intervals do not overlap; on the contrary, the other two solutions can be used only in the bitemporal format, since the Stripe- or L-shaped- inferred valid-time intervals overlap on the valid-time axis.

Furthermore, if we consider the bitemporal representations in Figs.1, 2, 3 and project (see Fig.4) the valid-time data inferred in each solution at $T = T^* < T_\infty$ and at $T = T_{now}$, we find out another important difference. At the generic time $T = T^* < T_\infty$, the Square Inference and the Stripe Inference return only the data whose original transaction-time pertinence contains T*, i.e. they return the *information which was available in the considered transaction-time interval*; the L-shaped inference returns all the data whose transaction-time pertinence *contains or precedes* T*, i.e. they return *all the information which was available at time T*, independently of the transaction-time pertinence of such data*.

If the projection is performed at the current time T_{now}, the Square inference and the Stripe inference return only the current data, whereas the L-shaped inference returns all the original data, both current and archived. These considerations can guide the use of the data produced in each type of inference. The Square method is most in harmony with the semantics of both transaction- and valid-time, thus it allows to use the inferred data for the translation to both the bitemporal- and the valid-time format in quite a safe way, just being aware that the validity were not explicitly defined. Also the Stripe method returns, for each transaction-time instant, only the data which were current at that time. In this sense, it is in harmony with the semantics of transaction-time. On the other hand, the inferred data can not be used with no transaction-time reference, because they overlap along valid-time. The L-shaped inference, if used with no transaction-time reference, is the most risky of the three. Its use can be justified by the fact that transaction-time can only grow, thus no retroactive insertion is possible. As a consequence, the original transaction-time data are, in each interval, the only one version which was ever recorded and thus, in this sense, *the most recently inserted one*.

References:

[BG93] Bhargava G., Gadia S.K., ``Relational Database Systems with Zero Information Loss", IEEE Trans. on Knowledge and Data Engineering, Vol. 5, No. 1, Feb. 1993.

[DGS94] De Castro C., Grandi F., Scalas M.R.: ``Semantic Interoperability of Multitemporal Relational Databases", in Entity-Relationship Approach - ER '93, Lecture Notes in Computer Science, Vol. 823, Springer-Verlag, 1994.

[DGS94] De Castro C., Grandi F., Scalas M.R.: ``Meaning of Relational Operations in Temporal Environment", accepted for presentation at the Basque International workshop on Information Technology (BIWIT 95), to be held in San Sebastian (Spain), July 1995.

[JCE+94] Jensen C., Clifford J., Elmasri R., Gadia S.K., Hayes P., Jajodia S. (editors), Dyreson C., Grandi F., Kafer W., Kline N., Lorentzos N., Mitsopoulos Y., Montanari A., Nonen D., Peressi E., Pernici B., Roddick J.F., Sarda N.L., Scalas M.R., Segev A., Snodgrass R., Soo M.D., Tansel A., Tiberio P., Wiederhold G.: ``A Consensus Glossary of Temporal Database Concepts", SIGMOD RECORD, Vol. 23, No. 1, March 1994

[KC93] Kim S.K., Chakravarthy S.: ``Modeling Time: Adequacy of Three Distinct Time Concepts for Temporal Databases", Proc. of 12th International Conference on Entity-Relationship Approach, Arlington, December 1993, also in Lecture Notes in Computer Science, Springer-Verlag.

[Soo91] Soo M., ``Bibliography on Temporal Databases," ACM SIGMOD Record, Vol. 20, No. 1, Mar. 1991.

[TSC+93] Tansel A., Snodgrass R., Clifford J., Gadia V., Segev A. (eds), *Temporal Databases: Theory, Design and Implementation*, The Benjamin/Cummings Publishing Company, Redwood city, California, 1993.