Capturing Temporal Constraints in Temporal ER Models

Carlo Combi[1], Sara Degani[1], and Christian S. Jensen[2]

[1] Department of Computer Science - University of Verona
Strada le Grazie 15, 37134 Verona, Italy
{carlo.combi,sara.degani}@univr.it
[2] Department of Computer Science - Aalborg University
Selma Lagerlöfs Vej 300, DK-9220 Aalborg Øst, Denmark
csj@cs.aau.dk

Abstract. A wide range of database applications manage information that varies over time. The conceptual modeling of databases is frequently based on one of the several versions of the ER model. As this model does not provide built-in means for capturing temporal aspects of data, the resulting diagrams are unnecessarily obscure and inadequate for documentation purposes. The TimeER model extends the ER model with suitable constructs for modeling time-varying information, easing the design process, and leading to easy-to-understand diagrams. In a temporal ER model, support for the specification of advanced temporal constraints would be desiderable, allowing the designer to specify, e.g., that the value of an attribute must not change over time. This paper extends the TimeER model by introducing the notation, and the associated semantics, for the specification of new temporal constraints.

Keywords: Conceptual modeling, database design, entity-relationship models, temporal databases, temporal data models, temporal constraints.

1 Introduction

A wide range of database applications manage information that varies over time: travel applications such as airline, train, and hotel reservations; record-keeping applications such as medical records; and financial applications like banking account management are some examples. Frequently, in the database design process for such applications, traditional data models are used; one of the several versions of the Entity-Relationship (ER) model is a common choice [1,3]. The ER model is easy to understand and use, and it allows one to define database schemata by means of easy-to-comprehend diagrams. Nevertheless, it does not explicitly support the management of time-varying information, and it is mainly left to the application designers and developers to discover and implement the temporal concepts meaningful for the application itself; this makes the design process more complex and leads to difficult-to-understand database diagrams. For this reason, a wide range of temporal extensions of the ER model have been developed by the research community over the years [6]. These extensions allow a more natural and elegant design of temporal databases, providing means for capturing the temporal aspects of the recorded data. Existing temporal ER extensions, however, do not consider some advanced temporal aspects in data modeling, such as, for example, specifying that an attribute cannot change over time, or that an entity identifier cannot be re-used for different entities in different times. In this paper, we extend the temporal data model TimeER [3,7], introducing new notations for the specification of advanced temporal constraints, thus enhancing the expressiveness and temporal support of the model. In particular, we propose some extensions to the notion of key constraint; we apply the concept of time-invariance to attributes and relationships; finally we introduce new temporal superclass/subclass relationship constraints. We are not aware of any other temporal ER model that supports the temporal constraints defined in this paper.

The paper is structured as follows. Section 2 introduces the TimeER model and a temporal relational model that we will use to define the semantics of the new temporal constraints; in Sect. 3, we describe a motivating example taken from a clinical scenario; in Sect. 4, we define new advanced temporal constraints for the TimeER model; Section 5 describes the semantics of the temporal key constraint defined in Sect. 4; finally, Sect. 6 offers concluding remarks and identifies directions for future research.

2 Background

In the following, we describe the main aspects of the TimeER model; furthermore, we present a temporal relational data model that will be used for the definition of the semantics of the new temporal constraints.

2.1 The TimeER Model

The Time Extended ER (TimeER) model [3,7] extends the EER model described by Elmasri and Navathe [1]. Existing ER constructs with their usual semantics are retained, and new notation providing implicit temporal support is added. More specifically, built-in temporal support is included for entities, relationships, and attributes. Four types of temporal aspects of information can be captured in a TimeER diagram, namely *valid time*, *transaction time*, *lifespan*, and *user-defined time* [8]. Table 1 indicates which aspects of time may be associated with each database concept. Note that TimeER offers support for both lifespan and valid time for relationships, as a relationship can be perceived as an entity that exists in its own right, or as a "complex attribute" of the involved entities. Temporal aspects are captured adding annotations to the modeling constructs: LS indicates lifespan support, VT indicates valid-time support, TT indicates transaction-time support, LT indicates lifespan and transaction time support, and finally BT indicates both valid and transaction time support. In Sect. 3, we describe an example that illustrates how temporal information is represented in a TimeER diagram.

Table 1. Association of aspects of time to TimeER database concepts

	Entity types	**Relationship types**	**Super/subclass relationships**	**Attributes**
Lifespan	Yes	Yes (entity view)	No	No
Valid time	No	Yes (attribute view)	No	Yes
Transaction time	Yes	Yes	No	Yes

2.2 The Surrogate-Based Relational Model

In the following, we describe the surrogate-based relational model. A mapping from TimeER modeling constructs to the surrogate-based relational model is defined in [4].

Domains of Attributes. The surrogate-based relational model supports the lexical domains of the standard relational model: $\mathcal{D}_D = \{D_1, \ldots, D_n\}$. Further, it supports a domain of surrogates, termed the *E-domain*, and three time domains, D_{LS} (lifespan domain), D_{VT} (valid time domain) and D_{TT} (transaction time domain).

Surrogates are system-generated unique internal identifiers; their values cannot be modified by the users of the model. Attributes defined over the E-domain are called *E-attributes*, and attributes defined over the time domains are termed *time attributes.* As a convention, the names of E-attributes end with the character ø; the names of time attributes are $\mathrm{LS_s}$, $\mathrm{LS_e}$, $\mathrm{VT_s}$, $\mathrm{VT_e}$, $\mathrm{TT_s}$, $\mathrm{TT_e}$, where s and e indicates start and end of the considered temporal dimension, respectively.

Relations. The surrogate-based relational model has two types of relations, *E-relations* and *A-relations.* E-relations are used to represent TimeER entity types and also relationship types that are considered to exist in their own right. An *E-relation* has a single *E-attribute* and a number of time attributes, depending on the time support specified for the corresponding entity type. *A-relations* represent entity or relationship attributes. An A-relation references, through a surrogate, the E-relation corresponding to the entity type the represented attributes belong to. If the represented attribute is temporal, the A-relation has also a number of time attributes.

Keys and Constraints. In traditional relational models, a primary key normally serves two roles: it is a lexical identifier, and it models existence. In the surrogate-based data model, lexical identification and existence are separated: E-relations only have a primary key, as the term "primary key" is used to exclusively model existence; A-relations have a unique identifier termed "key" [4]. At any point in time, null-values are not allowed in E-relations (Entity Integrity Constraint), and all surrogates referenced from an A-relation must exist in the corresponding E-relation (Referential Integrity Constraint). Pairs of time attributes are used to record the starting and the ending chronons of time intervals. The semantics of temporal aspects is enforced through the definition of a number of constraints on the database. For example, the valid time of any tuple of an A-relation must be included in the lifespan of the referenced tuple of the corresponding E-relation. These constraints are not enforced automatically by the data model, but must be enforced by explicit specifications, e.g., using assertions.

3 Motivating Example

We proceed to briefly introduce a motivating example taken from the clinical context. The TimeER diagram in Fig. 1 models a clinical database that stores information about patients, their admission into the hospital, drugs, and physicians. A patient is identified by an SSN (Social Security Number) and is characterized by name, address, and birth place; it must be considered that the SSN of a patient could change over time if, e.g., the patient changes name for some reasons. Information about changes of patients' addresses are recorded, too. The hospital records the patients' hospitalization history, and when this information is current in the database. Each hospital admission is characterized by a code and by the admission reason. Three different kinds of hospital admission are possible: emergency admission, regular admission, and day-hospital admission. For each emergency admission, the database stores information about the assigned bed number and the emergency level, and about possible changes of bed; for each regular admission, information about the assigned bed number (and possible changes to it) and the reservation number are recorded. Moreover, for each day-hospital admission the database stores information about the reservation number.

The hospital mantains data about drugs and about each single drug package. Each drug is identified by its National Drug Code (NDC) and is characterized by a name and by the drug class (prescription drug or over-the-counter drug); the NDC of a drug can neither vary over time, nor be re-assigned to a different drug. For each drug package, the tracking code is recorded; the tracking code of a drug package cannot vary over time; moreover, it cannot be re-assigned to other drug packages before six years from its assignement. Information about patient allergies to drugs are stored too; it must be considered that once a drug allergy is recognized, it cannot disappear.

Physicians make diagnoses on patients; each physician is identified by a code and is characterized by a name. The hospital keeps track of diagnosis histories, and also of when this information is recorded in the database. Physicians can be either hospital physicians or general practitioners; the basic information of a physician needs to remain in the database even in the case the physician decides to resign.

The TimeER model allows one to represent several temporal aspects of the above example. First of all, changes of the address of a patient are captured through the VT annotation for the attribute *address* of the entity *Patient*. Changes of bed of a patient are modeled similarly. Moreover, the TimeER model

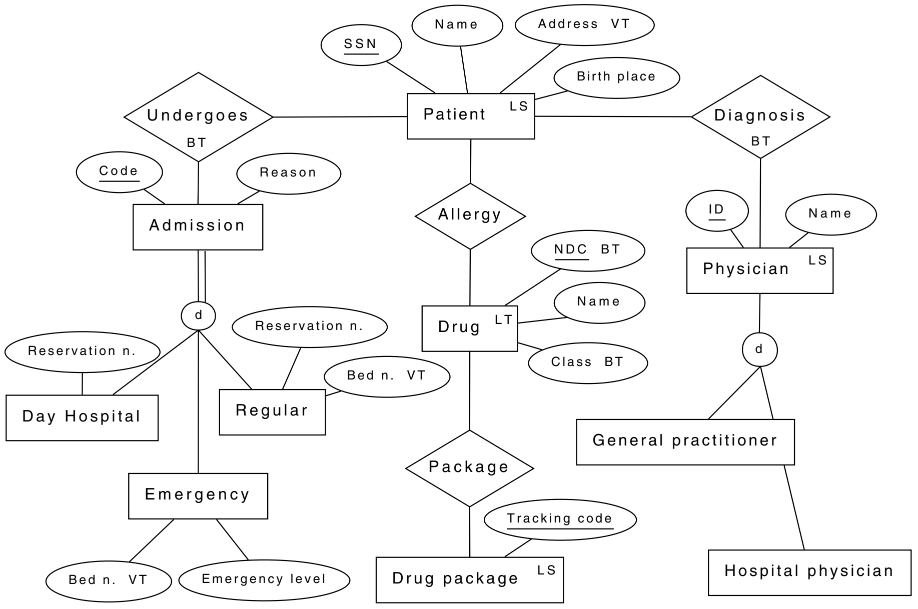

Fig. 1. A TimeER diagram modeling a clinical database

allows one to keep track of the hospitalization history of a patient, and of when that information is available in the database, through the BT annotation for the relationship *Undergoes*. Similarly, the *Diagnosis* relationship is annotated with BT to record patient diagnosis history and the time during which the information is available in the database. The existence-times of entities *Patient*, *Physician*, and *Drug package* are recorded by the LS annotation for the respective entities; moreover, for the entity *Drug* existence-time and time of occurence in the database are stored by the LT annotation.

Nevertheless, we can observe that some of the database requirements cannot be properly expressed by TimeER constructs. First of all, TimeER keys have a snapshot-reducible semantics [7], as it is ensured that any key at any point in time uniquely identifies an entity; this notion, however, is not sufficient to constrain a key value to be time-invariant, as is required for the *Tracking code* attribute of the entity *Drug package*; moreover, it does not allow one to express the fact that a key value cannot be assigned to two different entities at two different points in time, as is required for the *NDC* attribute of the entity *Drug*. The notion of time-invariance cannot be expressed for regular attributes either; this would be desiderable for the attribute *Birth place* of the entity *Patient*, as the birth place of a person does not change over time; moreover, TimeER does not provide means to express the fact that a drug allergy of a patient cannot disappear over time. Finally, TimeER superclass/subclass relationship constraints do not allow one to express, for example, that the kind of admission of a patient cannot change over time.

4 Introducing New Temporalities in TimeER

In this section we introduce the main contribution of the paper, namely the definition of advanced temporal constraints for TimeER diagrams. First, we consider the key constraint, and we define different versions of it, considering its temporal aspects. Then we apply the notion of time invariance to attributes and relationships. Finally, we define new temporal constraints over superclass/subclass relationships. Figure 2 shows how the diagram in Fig. 1 can be modified to also capture the new constraints.

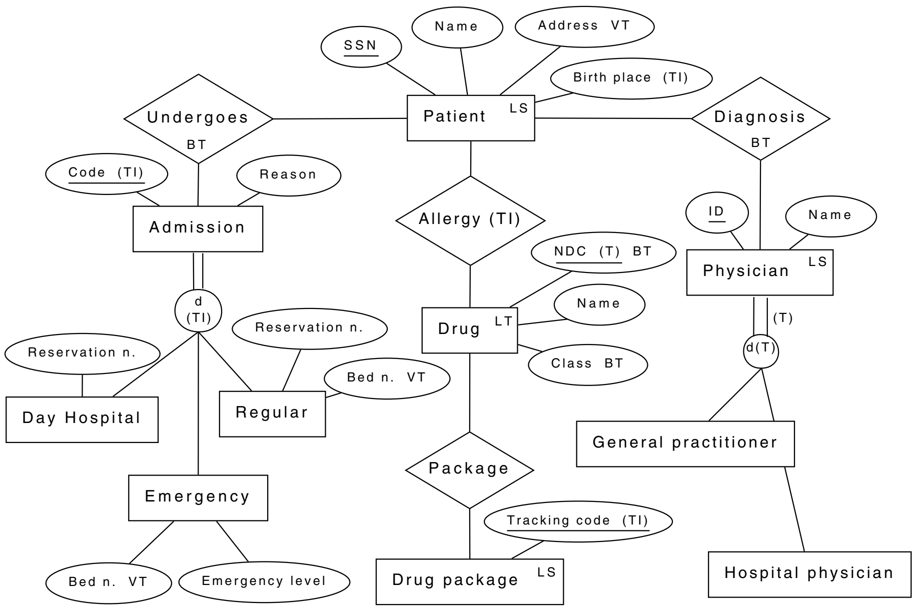

Fig. 2. The TimeER diagram modeling the clinical database with the new constraints

4.1 Key Constraints

As the traditional ER model, the TimeER model allows one to indicate that a set of attributes represents the key of an entity type. TimeER keys have a snapshot-reducible semantics. In a temporal ER model, however, it may be desiderable to have the possibility of specifying different kinds of key constraints, considering the relationship between key attributes and time. In the following, we first describe the notion of snapshot-reducible key as defined in the TimeER model. Therefore, we define and compare two advanced notions of key constraint, namely the *time-invariant key constraint* and the *temporal key constraint.*

Definition 1 (Snapshot-Reducible Key). *An entity snapshot-reducible key (or simply entity key) is a group of the entity's attributes that has to satisfy the following constraint: at any point in time, the mapping from the entity set to the corresponding set composed of groups of values for the key attributes is one-to-one.*

Definition 1 states that a snapshot-reducible key, at any point in time, uniquely identifies an entity. The concept of snapshot-reducible key is defined in terms of conventional keys and snapshot reducibility: snapshot reducibility ensures, for example, that at any point in the valid time domain, a single-valued attribute, for which valid time is captured, has only one value for an entity; combining this with the conventional key constraint, we have that any key attribute at any point in time uniquely identifies an entity.

An example scenario describing the application of the snapshot-reducible key constraint is represented by the entity *Patient* in Fig. 2. The Social Security Number of a patient can be updated over time; for example, a foreigner moving to Italy could change his surname on the basis of the Italian surname attribution rules, which are different from those of the patient's original country; as a consequence, the Social Security Number, too, has to be modified. Then, the *SSN* key attribute has a snapshot-reducible semantics, as its value for a specific entity has to be unique at each single point in time, but it may vary over time.

For a snapshot-reducible key, it is ensured that at any point in time, each entity has a different value for the key attribute; however, for two different points in time, the same key value could identify two different entities, or an entity could be identified by two different values of the key. Considering this aspect, we introduce in the TimeER model the possibility of specifying two more restrictive kinds of key. We call the first one *time-invariant key*, and it is defined as follows.

Definition 2 (Time-Invariant Key). *An entity time-invariant key is a group of the entity's attributes with the following properties: it is a snapshot-reducible key, and the values of the key attributes of an entity do not change over time in the valid-time domain.*

The second point in Definition 2 states that an entity is identified by the same time-invariant key value for all times in the valid-time domain: if we fix a single point in the transaction-time domain (if required) and then consider two different points in the valid-time domain, the same entity cannot have two different values for the key attribute. In other words, two entities identified by different time-invariant key values are different entities.

An example of application of time-invariant key is represented by the *Tracking code* key attribute of the entity *Drug package* in Fig. 2. The tracking code is a unique number associated with each drug package that cannot be repeated for at least six years. This means that two drug packages identified by different tracking codes are different packages, but the same tracking code could identify two different drug packages at two different valid-time instants. It follows that the *Tracking code* key has a time-invariant semantics.

If an entity key is defined as time-invariant, it cannot have two different values for the key attributes at two different valid-time points, but the key value can be reassigned to different entities over time. We introduce a third kind of key constraint in TimeER that is even more restrictive than the time-invariant key, as it also prevents the reassignement of key values over time. It is defined as follows.

Definition 3 (Temporal Key). *An entity temporal key is a group of the entity's attributes with the following properties: it is a time-invariant key, and the values of the key attributes of an entity cannot be reassigned to a different entity in the valid-time domain.*

The first and the second point of Definition 3 imply that two entities identified by different time-invariant key values are different entities; the third point implies that two entities identified by the same time-invariant key value are the same entity. The *NDC* attribute of the entity *Drug* in Fig. 2 is an example of a temporal key. Indeed, the National Drug Code is a number unique to every drug type that can neither vary over time for a specific drug type, nor be reassigned to a different drug.

The graphical notation for the three kinds of key is the following. To indicate that an entity attribute is a snapshot-reducible key, the key attribute name is underlined. For a time-invariant key, the label (TI) is placed to the right of the attribute name, and it is underlined together with the attribute name; for a temporal key, a (T) is used in the same way.

For all the three kinds of keys it is possible to specify, besides valid time, transaction time, too. Specifying the valid time for a time-invariant or a temporal key does not serve the purpose of keeping track of changes of the key value in the valid-time domain; however, even though a temporal key value cannot vary over time, specifying its valid time can be useful, as it enables the capture of the starting and the ending instants of validity of the value. Moreover, the specification of transaction time for time-invariant key attributes allows one to view previously current database states.

4.2 Time-Invariant Attributes

Similarly to how we applied the notion of time-invariance to entity keys, we can apply this notion to simple attributes.

Definition 4 (Time-Invariant Attribute). *A time-invariant attribute is an attribute whose value does not change over time in the valid-time domain.*

This means that, (possibly) given a fixed point in the transaction-time domain, a time-invariant attribute of a given entity cannot have two different values for two different points in the valid-time domain. Time-invariant attributes model entity and relationship properties that do not vary over time. A simple example is the attribute *Birth place* of the entity *Patient* in Fig. 2. As the birth place of a person does not change over time, the attribute is specified as time-invariant.

As we can see in the figure, the graphical notation for time-invariant attributes is the label (TI) placed to the right of the attribute name. Valid and/or transaction time can be specified also for time-invariant attributes.

4.3 Time-Invariant Relationships

The notion of time-invariance can be applied to entity relationships, too.

Definition 5 (Time-Invariant Relationship). *A relationship between two entities is time-invariant if, once it has been established, it holds as long as both involved entities exist in the mini-world.*

Definition 5 implies that a time-invariant relationship can start at any point during the existence of the involved entities, but that, after the starting instant, it has to hold for all the time during which the involved entities exist in the modeled reality.

In Fig. 2, the entity *Patient* is related to the entity *Drug* by means of the relationship *Allergy*. It can be observed that once a drug allergy is recognized, it cannot disappear; therefore, each instance of the relationship *Allergy* holds as long as the involved *Patient* and *Drug* instances exist in the modeled reality. It follows that the relationship *Allergy* is time-invariant.

A time-invariant relationship R is represented by placing the label (TI) in the right corner of the diamond representing R. Temporal support can be specified also for time-invariant relationships.

4.4 Superclass/Subclass Participation Constraints

The TimeER model allows one to specify snapshot totality and disjointness constraints over superclass/subclass relationships, which state that the traditional totality and disjointness constraints, respectively, must hold at each single point in time. In many situations, however, the notions of snapshot totality and disjointness constraints are not adequate to express the actual semantics of the superclass/subclass relationship. We therefore define the advanced notions of *temporal totality constraint*, *temporal disjointness constraint*, and *time-invariant superclass/subclass relationship.*

Definition 6 (Temporally-Total Superclass/Subclass Relationship). *Let E and $E_1, \ldots, E_n$ be* TimeER *entities such that E is a superclass and $E_1, \ldots, E_n$ are subclasses of E. If the superclass/subclass relationship is temporally total, then each member of the superclass is a member of at least one of the subclasses for at least one time instant in its lifespan.*

A temporally total superclass/subclass relationship is represented by placing the label (T) near the double line that represents the total participation constraint.

A situation, in which the temporal totality constraint is necessary to express the actual semantics of a superclass/subclass relationship, is represented by the superclass *Physician* and its subclasses in Fig. 2. The modeled database keeps track of hospital physicians and general practitioners; suppose that the hospital inserts data about a physician through occurrences of either entity *General practitioner* or entity *Hospital physician.* If a physician resigns, for example to start working privately, the basic information about the entity still needs to remain in the database; therefore, from the instant of the physician's resignation, the entity becomes an instance of the superclass only. It follows that each physician recorded in the database must be a general practitioner or a hospital physician for at least one time instant in its lifespan. This condition can be expressed by means of the temporal totality constraint.

Definition 7 (Temporally-Disjoint Superclass/Subclass Relationship). *Let E and $E_1, \ldots, E_n$ be* TimeER *entities such that E is a superclass and $E_1, \ldots, E_n$ are subclasses of E. If the superclass/subclass relationship is temporally disjoint then an instance e of E is a member of at most one of the subclasses for all times in its lifespan.*

A temporally disjoint superclass/subclass relationship is represented by placing the label (T) in the circle containing the specification of the disjointness constraint.

As an example, consider the superclass/subclass relationship given by the entity *Physician* and its subclasses in Fig. 2. Suppose that the considered hospital does not allows a hospital physician to become a general practitioner, and vice–versa. In this case, the superclass/subclass relationship is temporally disjoint, as for all its lifespan, an instance of *Physician* can be a member of at most one of the two subclasses *General practitioner* and *Hospital physician.*

Definition 8 (Time-Invariant Superclass/Subclass Relationship). *A superclass/subclass relationship is time-invariant if each member of the superclass that belongs to one or more subclasses is a member of those subclasses for all of its lifespan.*

From the definition, it follows that the existence time of each instance of the subclasses is equal to the existence time of the corresponding instance of the superclass.

An example of time-invariant relationship is shown in Fig. 2, by the superclass *Admission* and its subclasses. A patient admission can only be an emergency admission, a regular admission, or a day hospital admission at a time, and the kind of admission cannot change over time. Therefore, an instance of the entity *Admission* is an instance of one of its subclasses *Emergency admission*, *Regular admission*, or *Day hospital* for all its lifespan; it follows that the superclass/subclass relationship is time-invariant. A time-invariant superclass/subclass relationship is represented by placing the label (TI) in the circle representing the superclass/subclass relationships.

5 Semantics

The semantics of the TimeER constraints defined in Sect. 4 can be expressed by means of their mapping to the surrogate-based relational model, presented in Sect. 2.2. In the following, we give the semantics for the temporal key constraint.

5.1 Semantics of the Temporal Key Constraint

In order to define the semantics of the temporal key constraint through its mapping to the target relational model, we first recall how entity types and their attributes are mapped to relations of the surrogate-based relational model [4]. For a temporal entity type, an E-relation is created as the union of the E-attribute and the time attributes corresponding to the temporal support specified for the entity type. Moreover, for each temporal attribute, an A-relation is created as the union of the E-attribute, the attribute itself, and the associated time attributes.

Some constraints apply to the relations created by the mapping [4]. First of all, it must be enforced that the information recorded by the A-relation is snapshot reducible: for A-relations recording valid time only, this means that no two tuples of the A-relation containing the same E-attribute can have overlapping valid-time intervals; similar constraints apply for A-relations recording transaction time only or both valid time and transaction time.

A temporal constraint must hold to ensure that attributes of temporal entities cannot be associated with time intervals for which the entities do not exist or are not registered in the database. For example, if the tuples in an A-relation representing temporal attributes record valid-time only, then the valid-time intervals have to be included in the lifespan interval recorded by the tuple of the E-relation with the same value of the E-attribute. Similar constraints apply for all the combinations of temporal support for the A-relation and the E-relation.

As an example, Table 2 represents the result of the mapping of the entity *Drug* in Fig. 2 and of its attributes. The attributes that are overlined in the relations indicate the primary keys of the relations, while attributes that are underlined constitute keys of the relations. The term *primary key* exclusively indicates existence; consequently, only E-relations have primary keys. The unique identifier of an A-relation is simply termed a *key*. In the example, the primary key of the E-relation *Drug* includes the LS_s timestamp attribute; the reason is that the surrogate-based relational model allows an entity to reborn in the database. It is worth noting that (possibly non temporally continuous) histories of entities and their attribues can be suitably derived through joins between the E-relation and the A-relations representing the considered entity. The attributes that the user may have specified as a key for an entity type in the diagram are indicated by the symbol "u.k." in a relation. Foreign keys of relations are indicated by the symbol "f.k." following the attribute names.

A further constraint must be enforced to ensure that, for each instant of validity of the value of an entity attribute, a corresponding value exists in the A-relation representing the user-defined key. Constraint 1 ensures this in the case where the user-defined key is a temporal key.

Table 2. The result of the mapping of the entity *Drug* in Figure 2

Drug

$\overline{\text{drugø}}$	$\overline{\text{LS}_\text{s}}$	LS_e

Drug_NDC

$\underline{\text{drugø}}$ f.k.	NDC u.k.	$\underline{\text{VT}_\text{s}}$	$\underline{\text{VT}_\text{e}}$	$\underline{\text{TT}_\text{s}}$	TT_e

Drug_name

$\underline{\text{drugø}}$ f.k.	name

Drug_class

$\underline{\text{drugø}}$ f.k.	class	$\underline{\text{VT}_\text{s}}$	$\underline{\text{VT}_\text{e}}$	$\underline{\text{TT}_\text{s}}$	TT_e

Constraint 1. *Let E be a* TimeER *entity with a temporal key for which valid time only is captured. Let R be the A-relation storing the temporal key of E, and let r_i be a tuple variable over R. Let S be an A-relation representing an attribute of E, and let s_i be a tuple variable over S. Let Rø and Sø be the foreign key of R and S, respectively, referring to the surrogate attribute of the E-relation representing E. Then:*

$$\forall s_i \in S\ \exists r_i \in R(s_i.S\text{ø}= r_i.R\text{ø} \wedge [s_i.VT_s, s_i.VT_e] \subseteq [r_i.VT_s, r_i.VT_e])$$

Constraint 1 can be straightforwardly defined for the cases in which transaction time also is captured for the A-relation recording the entity attribute and/or for the A-relation recording the temporal key.

We therefore define the mapping of the temporal key constraint as Constraint 2 and Constraint 3; these constraints apply to the relation representing the temporal key attribute. Constraint 2 applies in the case where valid time only is captured for the temporal key attribute, and Constraint 3 applies when both valid time and transaction time are captured.

Constraint 2. *Let E be a* TimeER *entity with a temporal key for which valid time only is captured. Let R be the A-relation storing the temporal key of E, and let r_i, r_j be tuple variables over R. Let X be the group of attributes of R that represents the temporal key of E. Let Rø be the foreign key of R referring to the surrogate attribute of the E-relation representing E. Then:*

$$\forall r_i, r_j \in R\ ((r_i.R\text{ø} = r_j.R\text{ø} \ \Leftrightarrow r_i.X = r_j.X) \wedge \\ ((r_i.X = r_j.X \wedge [r_i.VT_s, r_i.VT_e] \cap [r_j.VT_s, r_j.VT_e] \neq \emptyset) \Rightarrow r_i = r_j)).$$

Constraint 3. *Let E be a* TimeER *entity with a temporal key for which both valid and transaction time are captured. Let R be the A-relation storing the temporal key of E, and let r_i, r_j be tuple variables over R. Let X be the group of attributes of R that represents the temporal key of E. Let Rø be the foreign key of R referring to the surrogate attribute of the E-relation representing E. Then:*

$$\forall r_i, r_j \in R\ (((r_i.R\text{ø} = r_j.R\text{ø} \wedge \\ [r_i.TT_s, r_i.TT_e] \cap [r_j.TT_s, r_j.TT_e] \neq \emptyset) \Leftrightarrow r_i.X = r_j.X) \wedge \\ ((r_i.X = r_j.X \wedge [r_i.VT_s, r_i.VT_e] \cap [r_j.VT_s, r_j.VT_e] \neq \emptyset \wedge \\ [r_i.TT_s, r_i.TT_e] \cap [r_j.TT_s, r_j.TT_e] \neq \emptyset) \Rightarrow r_i = r_j))$$

As an example, Table 3 shows an instance of the relation *Drug_NDC* of Table 2 that satisfies Constraint 3. Indeed, for each single point in the transaction-time domain, the mapping from the set of the *NDC* attribute values to the *Drug* entity set is one-to-one. Table 4, on the contrary, shows an instance that violates Constraint 3, as the key value of the entity with surrogate-value ø1 varies over time; moreover, the key value 00002-7597-01 that, during the valid-time interval [1, 20], is assigned to the entity with surrogate value ø2, is the value for the key attribute of the entity identified by the surrogate value ø3 during the valid-time interval [21, NOW].

Table 3. An example of satisfaction of temporal key constraint

Drug_NDC

drugø f.k.	NDC u.k.	VT_s	VT_e	TT_s	TT_e
ø1	50242-0040-62	1	NOW	1	10
ø1	60575-4112-01	1	NOW	11	UC
ø2	00002-7597-01	1	NOW	1	UC

Table 4. An example of violation of temporal key constraint

Drug_NDC

drugø f.k.	NDC u.k.	VT_s	VT_e	TT_s	TT_e
ø1	50242-0040-62	1	10	1	UC
ø1	60575-4112-01	11	NOW	11	UC
ø2	00002-7597-01	1	20	1	UC
ø3	00002-7597-01	21	NOW	21	UC

The notion of temporal key constraint can be defined by means of suitable temporal functional dependencies derived from those proposed in the literature [9] for a bitemporal data model, and by introducing the *temporal natural join* operator.

Intuitively, if X and Y are sets of non-timestamp attributes of a relation schema S, a *temporal functional dependency* $X \xrightarrow{\mathrm{T}} Y$ exists on S if, considering an instance of S as a collection of snapshot relations, the corresponding conventional functional dependency $X \longrightarrow Y$ holds on each such snapshot in isolation. Moreover, a *strong temporal functional dependency* $X \xrightarrow{\mathrm{Str}} Y$ exists on S if, (possibly) fixed a transaction time instant, if the value of X does not vary in two different valid time instants, then the value of Y does not vary as well. Finally, a *strong temporal equivalence* $X \xleftrightarrow{\mathrm{Str}} Y$ exists on S if $X \xrightarrow{\mathrm{Str}} Y$ and $Y \xrightarrow{\mathrm{Str}} X$.

Table 5a shows an instance of the *Drug_class* relation with the schema described in Table 2, and an instance of the *Drug_NDC* relation that satisfies the temporal key constraint; Table 6b shows the result of a temporal natural join over these two instances. A temporal natural join is a binary operator that generalizes the snapshot natural join to incorporate one or more time dimensions. Tuples in a temporal natural join are merged if their explicit join attribute values match, and they are temporally coincident in the given time dimensions. We can notice that the following temporal functional dependencies hold for the S relation: $Drug\o \overset{\text{Str}}{\longleftrightarrow} NDC$ and $NDC \overset{\text{T}}{\rightarrow}$ S, where S is the set of all the attributes of relation S.

Table 5.

Drug_NDC

drugø f.k.	NDC u.k.	VT_s	VT_e	TT_s	TT_e
ø1	50242-0040-62	1	NOW	1	10
ø1	60575-4112-01	1	NOW	11	UC
ø2	00002-7597-01	1	NOW	1	UC

Drug_class

drugø f.k.	class	VT_s	VT_e	TT_s	TT_e
ø1	Prescription	1	NOW	1	10
ø1	Over-the-counter	1	NOW	11	UC
ø2	Prescription	1	NOW	1	UC

(a) Satisfaction of temporal key constraint

$S = Drug\_NDC \bowtie^T Drug\_class$

drugø f.k.	NDC u.k	class	VT_s	VT_e	TT_s	TT_e
ø1	50242-0040-62	Prescription	1	NOW	1	10
ø1	60575-4112-01	Over-the-counter	1	NOW	11	UC
ø2	00002-7597-01	Prescription	1	NOW	1	UC

(b) The temporal natural join over the relations in Table 6a

The temporal key constraint can therefore be defined as follows.

Definition 9. *Let E be an entity, and let $A_1, \ldots, A_n$ be the A-relation schema that represent the attributes of E. Let $E\o$ be the surrogate attribute of E, and let X be the set of attributes representing the key of E. Let S be the relation schema derived from the temporal natural join of $A_1, \ldots, A_n$. Then X is termed* temporal key *if $E\o \overset{\text{Str}}{\longleftrightarrow} X$ and $X \overset{\text{T}}{\rightarrow} S$.*

6 Summary and Research Directions

In this paper we extended the expressiveness of the TimeER model [3,7] with new constructs for specifying advanced temporal constraints. More specifically, we focused on enabling the expression of different temporal semantics of attributes and relationships, for keys, and for superclass/subclass relationships.

Furthermore, we demonstrated how it is possible to define the semantics of the temporal constraints by means of a surrogate-based relational model.

As for future work, we will focus on the completeness of the proposed extension of TimeER with respect to the requirements of database designers. Moreover, we will evaluate our proposal with respect to real-world conceptual design tasks, through the use of a prototype implementing the described constraints.

References

1. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems, 2nd edn., Benjamin/Cummings (1994)
2. Gregersen, H.: TimeERplus: A Temporal EER Model Supporting Schema Changes. In: Jackson, M., Nelson, D., Stirk, S. (eds.) BNCOD 2005. LNCS, vol. 3567, pp. 41–59. Springer, Heidelberg (2005)
3. Gregersen, H.: The Formal Semantics of the TimeER model. In: 3rd Asia-Pacific Conference on Conceptual Modelling, vol. 53, pp. 35–44. Australian Computer Society, Hobart (2006)
4. Gregersen, H., Mark, L., Jensen, C.S.: Mapping Temporal ER Diagrams to Relational Schemas. Technical report TR–39, TimeCenter (1998)
5. Gregersen, H., Jensen, C.S.: On the Ontological Expressiveness of Temporal Extensions to the Entity-Relationship Model. In: 1st International Workshop on Evolution and Change in Data Management, pp. 110–121. Springer, Paris (1999)
6. Gregersen, H., Jensen., C.S.: Temporal Entity Relationship Models - a Survey. IEEE Trans. Knowl. Data Eng. 11, 464–497 (1999)
7. Gregersen, H., Jensen, C.S.: Conceptual Modeling of Time-Varying Information. In: 2nd International Conference on Computing, Communications and Control Technologies, Austin, pp. 248–255 (2004)
8. Jensen, C.S., Dyreson, C.E. (eds.): Dagstuhl Seminar 1997. LNCS, vol. 1399, pp. 367–405. Springer, Heidelberg (1998)
9. Jensen, C.S., Snodgrass, R.T.: Temporally Enhanced Database Design. In: Advances in Object-Oriented Data Modeling, pp. 163–193. MIT Press, Cambridge (2000)