On the Ontological Expressiveness of Temporal Extensions to the Entity-Relationship Model

Heidi Gregersen, Christian S. Jensen

August 7, 2002

TR-69

A TIMECENTER Technical Report

Title On the Ontological Expressiveness of Temporal Extensions to the Entity-Relationship Model

Author(s) Heidi Gregersen, Christian S. Jensen

Publication History Proceedings of the 1st International Workshop on Evolution and Change in Data Management, Paris, France, November 15 - 18, 1999, pp. 110-121.
August 2002, A TIMECENTER Technical Report.

Abstract

It is widely recognized that temporal aspects of database schemas are prevalent, but also difficult to capture using the ER model. The database research community's response has been to develop temporally enhanced ER models. However, these models have not been subjected to systematic evaluation. In contrast, the evaluation of modeling methodologies for information systems development is a very active area of research in information systems engineering community, where the need for systematic evaluations of modeling methodologies is well recognized.

Based on a framework from information systems engineering, this paper evaluates the ontological expressiveness of three different temporal enhancements to the ER model, the Entity-Relation-Time model, the TERC+ model, and the Time Extended ER model. Each of these temporal ER model extensions is well-documented, and together the models represent a substantial range of the design space for temporal ER extensions. The evaluation considers the uses of the models for both analysis and design, and the focus is on how well the models capture temporal aspects of reality as well as of relational database designs.

The database research community had over the last 2 decades developed temporally enhanced ER models. However, these models have not been subjected to systematic evaluation. In contrast, the evaluation of modeling methodologies for information systems development is a very active area of research in information systems engineering community, where the need for systematic evaluations of modeling methodologies is well recognized.

Based on a framework from information systems engineering, this paper evaluates the ontological expressiveness of three different temporal enhancements to the ER model, the Entity-Relation-Time model, the TERC+ model, and the Time Extended ER model. Each of these temporal ER model extensions is well-documented, and together the models represent a substantial range of the design space for temporal ER extensions. The evaluation considers the uses of the models for both analysis and design, and the focus is on how well the models capture temporal aspects of reality as well as of relational database designs.

1 Introduction

Both the research community and the companies that design databases have recognized that temporal aspects of database schemas are both prominent and difficult to capture using the ER model. Intuitive and easy-to-comprehend diagrams become obscure and cluttered when modeling fully the temporal aspects. In companies this problem is often solved either by totally ignoring the temporal aspects in the ER diagrams or supplementing the diagrams with phrases such as "full temporal support." The result is that the mapping of ER diagrams to relational tables must be performed by hand; and the ER diagrams do not document well the temporally extended relational database schemas used by the application programmers.

The research community's response to the shortcomings of the regular ER model for the modeling of temporal aspects has been to develop temporally enhanced ER models, and a number of models have been reported in the research literature [12, 5, 16, 4, 2, 21, 20, 15, 26]. These temporal ER models are developed in an attempt to provide modeling constructs that more naturally and elegantly permit the database designer to capture general temporal aspects of information, such as valid and transaction time. For a survey of the existing models, see [8]

Both the standard and temporally enhanced ER models may be used for different, but related purposes, namely for analysis—i.e., for modeling a part of reality—and for design—i.e., for describing the database schema of a computer system. The typical use seems to be one where the model is used primarily for design, with the design diagrams also serving as analysis diagrams, and where the constructed diagrams are mapped to a relational platform. In step with the increasing diffusion and use of relational platforms in industry, ER modeling is growing in popularity.

In the database research community, the models that are offered for conceptual database design are rarely evaluated systematically. In contrast, in the area of information systems engineering, the evaluation of modeling methodologies for information systems development is a very active area of research. Researchers within this area have recognized the need for systematic evaluations of modeling methodologies. A substantial number of evaluations are reported in the literature [1, 13, 18, 6, 10, 17, 22, 23, 24, 14, 25], and IFIP Working Group 8.1 is co-sponsoring an annual workshop, EMMSAD, devoted solely to this topic. The outcomes of the evaluations are useful for both designers of new methodologies and the users of the methodologies: The designers can use the evaluation criteria as design criteria when developing new methodologies, and the users can use the results to identify the modeling methodology most suitable for their specific purposes.

Weber and Wand have developed a framework for evaluating the ontological expressiveness of information systems development methodologies [22, 23, 24]. This framework includes a "representational," ontological model of the real world that covers both structural and behavioral aspects. The framework has been used to evaluate the notations, and their semantics, of three models for information systems development, namely Data Flow Diagrams [22], the ER model [22, 23], and the NIAM methodology [25]. This paper reuses Weber and Wand's approach, but evaluates three different temporal extensions to the ER model. Since a temporal ER model, like the ER model, only models structural aspects, and since our focus is specifically on the temporal aspects, we do not use the representational model of Weber and Wand, but use new representational models.

Specifically, the objective of this paper is to evaluate the ontological expressiveness of the temporal notational constructs of three selected temporal ER models: the Entity-Relation-Time (ERT) model [21], the TERC+ model [26], and the Time Extended ER model (TIMEER) [7]. Each of these temporal ER model extensions is well-documented, and together the models represent a substantial range of the designs space for temporal extensions. The evaluation will consider the uses of the models for both analysis and design, and it is evaluated how well the models capture temporal aspects of reality as well as of a database design. This necessitates the use of two different representational models, one for analysis and one for design.

The three models to be evaluated were chosen based on their recency and quality. One was published in 1991, and the latter two are among the most recent published models and may be considered second-generation models. As such, the designers of these models may be expected to have benefited from the knowledge and insights accumulated in earlier proposals. Indeed, we find the chosen models to be some of the most promising proposals, and we feel that they represent well the state-of-the-art in the area.

In the literature, we have found four surveys and comparisons of methodologies for the analysis and design of information systems, which are to some extent related to this papers evaluations. Brandt [1] surveys and evaluates thirteen methodologies for system's specification. The methodologies are evaluated using a taxonomy with eight parameters: origin (where the methodology is developed) and experience, the development steps of the methodology, the description of the methodology, product control factors, representation means (notation, etc.), documentation, user-orientation, and supporting tools and prospects of the methodology. Kung [13] studies three conceptual models with a time perspective. The analysis is based on the following five general features of a conceptual model: its understandability, its expressiveness, its data processing independence, its check-ability, and its changeability. Floyd has [6] evaluated and compared three different system's development methodologies. The methods are evaluated with respect to six criteria, of which some are further subdivided. The criteria include the methodology's application area, perspective, and guidelines; the theory on which the methodology is based; and the coherence of the methodology's guidelines and its coverage of development tasks. Jayaratna [11] has developed a framework, termed NIMSAD, for understanding and evaluating methodologies. This framework distinguishes between four essential aspects, namely the problem situation (the methodology's context), the intended problem solver (the user of the methodology), the problem-solving process (the methodology), and the evaluation of the three former elements. This evaluation method is the only one that evaluates the methodology's users, which is highly relevant, e.g., in order to establish whether the users need further education in using a specific methodology. The evaluations above focus on modeling properties, the usage of the models and the user-friendliness of the models evaluated. Only one evaluation considers expressiveness as a criterion [13], but not with respect to the models' abilities to express temporal aspects; in contrast, the evaluation in this paper focuses entirely on expressiveness in relation to temporal aspects.

Conceptual models for database design have also been evaluated and compared. Schrefl et al. [18] develop a set of criteria for comparing conceptual, or "semantic," data models and evaluate seven conceptual data models with respect to these criteria. The criteria focus on which modeling constructs, a semantic data model should offer. Hull and King [10] discuss issues of conceptual data modeling and survey sixteen conceptual data models. The models are subdivided in five categories: prominent models (the most well known at the time), other highly structured models, binary models, and relational extensions. The focus of the survey is on the philosophical bases of the models and on their structural modeling constructs, and it is also considered whether or not the models have constructs that describe dynamic aspects. Peckham and Maryanski [17] describe generic properties of conceptual data models and survey a representative selection of models. They also offer guidelines for comparing conceptual data models; here, the focus is on whether or not a model offers modeling constructs for describing structural and dynamic aspects of a database. Leander et al. [14] compare the modeling capabilities of the ER model and the NIAM methodology. Specifically, they present and compare the modeling constructs of the two models; they evaluate the modeling capabilities based on the mappings of ER and NIAM diagrams into the relational model; and they relate the two models to design objectives, including expressiveness, declarativeness, simplicity, readability, minimality, and the existence of a formal basis. The evaluations of the conceptual models for database design all focus on non-temporal properties and on the use of the models for design. This papers evaluation focuses exclusively on the modeling of temporal aspects, and in addition considers the use of the models for analysis.

Temporally extended ER models have been surveyed and evaluated with respect to a set of evaluation criteria by the author in [8, 9]. The focus of these evaluations are entirely on model properties, and criteria based on real-world and relational ontologies are do not considered.

In summary, the focus of previous, related evaluations ranges from determining the environments in which methodologies where developed, over the usages of the methodologies, to the user-friendliness of the methodologies. Some studies examine the expressiveness of conceptual data models [13, 14]. However, previous work does not consider evaluation parameters that concern how well the models capture temporal aspects, which is the topic of this paper. We evaluate three different temporally extended ER models' abilities to describing the temporal aspects of reality and of relational database schema capturing temporal aspects. That is, this papers evaluation focuses entirely on how well temporally extended ER models capture the temporal aspects of reality and of a database design.

The paper is structured as follows. Section 2 characterizes temporally extended ER models. Section 3 states the objectives of the evaluations, and Section 4 presents the evaluation framework. In Sections 5 and 6, the three temporal ER extensions are evaluated. Finally, Section 7 summarizes the findings and outlines directions for future research.

2 Characteristics of a Temporal ER Model

This section offers an abstract characterization of the implications of supporting lifespan and valid and transaction time are for the ER model. Specifically, the interrelation between these three temporal aspects and the ER model's modeling constructs as well as the temporal aspects themselves is explored.

We use the term "fact" to denote any statement that can be assigned a truth value, i.e., true or false. In the ER model, unlike in the relational model, a database is not structured as a collection of facts, but rather as a set of entities and relationships with attributes, with the database facts being implicit.

The temporal aspects of information that have received the most attention in the temporal database community are *valid time*, *transaction time*, and *lifespan* (also termed “existence time”). Valid time applies to facts, lifespan applies to “things” with independent existence, and transaction time applies to “anything” that may be recorded in a database. All are general—rather than application specific—aspects of information. As such, these aspects are prime candidates for being built into a temporal ER model.

A data model should make it possible to conveniently and concisely capture all information about reality that is meaningful to capture and is relevant for the application at hand. Since any entity has existence and thus a lifespan, it should be possible for database designers to conveniently indicate that lifespans should be captured for entities. This is desirable because lifespans are important in many applications and because entities may exist beyond the times when their attributes have (non-null) values—it is thus not possible to infer the lifespans of entities from the valid times of the attribute values associated with the entities.

Because facts have *valid time* and attributes and relationship types are the modeling constructs that capture facts at the conceptual level, a temporal ER model should offer built-in support for registering valid time for attributes and relationship types. Built-in support for valid time is desirable because it is fundamentally important in a large class of applications to know at what times the facts recorded in the database are true.

An inherent constraint applies to valid time and lifespans. Specifically, at any time during the database’s evolution, the valid time of any attribute value of any entity must be a subset of the lifespan of the entity.

Next, anything, not just facts, that may be stored in a database has a *transaction time*, which may or may not be captured in the database. With transaction time being captured, past states of the database are retained, which is essential in applications with accountability or trace-ability requirements, of which there are many. The need for recording transaction time is thus widespread.

Above, we discussed which temporal aspects are applicable to which ER modeling constructs. Having introduced the temporal aspects, the next step is to consider them as modeling constructs in their own right and then consider which temporal aspects are applicable to them.

The three temporal aspects are akin to attributes; their values are associated with and describe other “things” and do not have independent existence. For example, lifespan may be considered a special kind of attribute, since a lifespan captures when a “thing,” modeled as an entity, exists in the modeled reality.

We proceed to consider in turn what temporal aspects may be meaningfully associated with lifespan and valid and transaction time. Lifespan cannot be associated with a lifespan, since a lifespan value is a kind of attribute value that characterizes an entity and does not have independent existence. In contrast, valid time may meaningfully be associated with a lifespan, but the valid-time value of a lifespan value would be equal to the lifespan value, so this would not provide any new information and is thus redundant. Transaction time may be associated with a lifespan, just as transaction time may be associated with any other attribute. This is actually what happens when an entity type supports both lifespan and transaction time.

Considering now valid time instead of lifespan, we observe that lifespan does not apply, for the same reason as above. Valid time can be associated with valid time, but as for lifespan above, this would not provide any new information. Transaction time can be associated with valid time, and this occurs when an attribute or relationship type is specified as bitemporal.

Now considering transaction time, again lifespan does not apply. Valid time can be associated with transaction time, but the valid-time value of a transaction-time value will be equal to the transaction-time value, so this does not add any new information. The transaction-time value of a transaction-time will also not provide any new information.

Table 1 summarizes which temporal aspects should be supported by which modeling construct. In the table, “N.A.” indicates “not applicable,” “X” means “applicable,” and “(X)” means “applicable, but redundant.”

The modeling constructs that remain to be considered are the integrity constraints that can be specified in an ER diagram. Specifically, it is possible to specify the attributes of an entity type that constitute the

	Lifespan	Valid Time	Transaction Time
Entity Types	X	N.A.	X
Attributes	N.A.	X	X
Relationship Types	N.A.	X	X
Lifespan	N.A.	(X)	X
Valid time	N.A.	(X)	X
Transaction Time	N.A.	(X)	(X)

Table 1: Application of Temporal Aspects to Modeling Constructs

identifier of the entity type. These attributes are then key attributes and a *key constraint* applies to the entity type. For each entity type participating in a relationship type, a participation constraint has to be specified. For ER models that have specialization/generalization constructs, constraints on how the entities of the superclass participate in the subclasses must be specified. In a temporal ER model, it would be natural to have both temporal versions of these, which apply over all of time, and snapshot reducible [19] versions, which apply in isolation at each single point in time. Table 2 describe these constraints in turn.

	Temporal	Snapshot Reducible
Key Constraint	For all of time, an entity is uniquely identified by the value(s) of the key attribute(s).	At each individual point in time, the value(s) of the key attribute(s) uniquely identifies an entity.
Relationship participation constraint	The participation constraint must hold for all of time.	At each individual point in time, the participation constraint must hold.
Disjointness generalization constraint (disjoint/overlapping)	For all of time, the specified disjointness constraint for the specialization must hold, e.g., if specified as disjoint then, once an entity becomes member of a subclass, the entity cannot become a member of any other subclass of the superclass.	At each individual point in time, the specified disjointness constraint must hold, e.g., if a disjoint constraint is specified then an entity cannot at any point in time be a member of more than one subclass, but it may be a member of more than one subclass of the superclass at different points in time.
Completeness generalization constraint (partial/total participation)	For all of time, the completeness constraint must hold, e.g., if partial is specified then an entity of the superclass need not ever be a member of any subclass.	At each individual point in time, the completeness constraint must hold.

Table 2: Temporal and Snapshot Reducible Integrity Constraints

Finally, valid time and transaction time could also be applied to the key constraint itself. For example, the meaning of assigning a valid time to a key constraint is that the constraint will apply during the specified time value only. The association between an attribute and an entity type could also be assigned a valid time and a transaction time, with the meaning of the former being that the entity type will have that particular attribute only during the specified time value. These assignments of temporal aspects specify how the database schema changes over time and thus concern schema versioning. We are not aware of any temporal

ER model that supports schema versioning, and although very interesting, we will consider this to be outside the scope of the thesis.

3 Evaluation Objectives

This section first describes the context of the evaluation, by outlining several uses and aspects of a conceptual model that may be subjected to evaluation. It then proceeds to state the objective of the evaluation, by positioning it within this context.

3.1 Types of Evaluation

As already suggested, we distinguish between two uses of a conceptual model, namely the use of a model for *analysis* and the use for *design*. When a model is used for analysis, it is used for modeling a small part of reality. When used for design, the aim is to model the underlying implementation, which is usually the relational database model. One method to evaluate a temporal ER model is to examine separately how well it performs with respect to analysis and design of temporal databases.

Three different approaches to evaluating a conceptual model can be taken. First, the evaluation can be done by examining the designs, or diagrams, that result from using the conceptual model, i.e., when evaluating a temporal ER model, the *diagrams* that result from using the model will be the target of the evaluation. Second, the *notation*, i.e., the "building blocks" of the conceptual model can be evaluated with respect to analysis or design by examining, e.g., which notational constructs the model offers for modeling specific aspects of either reality or the underlying implementation. Third, the *methods and guidelines* that describe how to use the model during the different stages of the activities can be evaluated.

The difference between evaluating the resulting diagrams versus the notation of a temporal ER model may be explained as follows. A temporal ER model is a graphical model, that is, the notation of the model is graphical symbols, including rectangles, diamond, and lines. Each symbol has a specific interpretation (semantics) that gives the meaning of the symbol. In contrast, a diagram is a connected collection of symbols. To exemplify the difference see Figure 4 (page 15). This figure presents a TIMEER diagram describing a database. A TIMEER notational construct is, for example, the rectangle that is used to represent entity sets. Evaluating a conceptual model by considering specific diagrams produced using the model versus evaluating it by considering each of its modeling constructs yields different evaluations.

It is meaningful to combine evaluations based on the the three aspects with the use of a conceptual model for both of the activities of analysis and design, yielding the following six different evaluation possibilities, which we will discuss in turn next.

$$\begin{matrix}\text{Analysis}\\ \text{Design}\end{matrix} \times \begin{matrix}\text{Diagrams}\\ \text{Notation}\\ \text{Methods and Guidelines}\end{matrix}$$

Analysis

One can evaluate a diagram with respect to analysis by observing how well the diagram describes the modeled reality. This is done by comparing the diagram with the requirements specification. The notation of the model is irrelevant to the evaluation of a diagram, in the sense that the focus is entirely on how easy it is to recognize the modeled reality based on a specific diagram. Another way to evaluate a diagram is to compare it with a diagram obtained using another conceptual data model. This will require that a metric be available for the comparison.

The notation of a conceptual model can be evaluated with respect to analysis by examining whether or not the modeling constructs offered by the model can describe the relevant reality with the desired accuracy. In order to do this, we have to examine what the reality consists of and examine whether or not the model offers modeling constructs that can describe the reality.

The methods and guidelines for how to use the model during the analysis phase, if provided, can be evaluated with respect to how well they support the users of the model in modeling reality.

Design

ER diagrams are often used as documentation for the schema of the underlying database. For example, if the underlying database is relational, the ER diagram should document the relational schema. This means that it should be possible for database designers to recognize the underlying database structure by examining the diagram.

When evaluating the notation of a model with respect to design, we have to examine which modeling constructs the underlying data model offers and and then to determine what modeling construct the conceptual model offers for describing these underlying-model constructs. We will assume that the underlying data model is the relational model.

Finally, the methods and guidelines for how to use the model in the design phase can be evaluated with respect to how well they help and support the construction of a model of the underlying database.

3.2 Choice of Evaluation Objectives

We have chosen not to base our evaluation on specific temporal ER diagrams, for several reasons. First, it is not clear who should create the diagrams. It is almost impossible to ensure that the corresponding diagrams for several different temporal ER models are created under similar conditions, e.g., are created by persons equally experienced in using the models and in reading a requirements specification. Solving this problem by having the same persons create all diagrams introduces new problems: the sequence in which the diagrams are created is likely to matter so that later diagrams are influenced by knowledge obtained during the creation of earlier diagrams. Second, designing a metric for comparing diagrams is problematic. The decision of what is to be measured, and the importance of this is subjective. Third, the evaluation of how well a ER diagram documents the underlying database is also quite subjective. Different database designers may have different answers to whether or not a digram documents the underlying database. We have also chosen not to evaluate the methods and guidelines that the designers of the temporal ER models may have given, for the simple reason that no temporal ER model is equipped with such methods and guidelines.

Rather, we have chosen to evaluate the notations of three temporal ER models with respect to both analysis and design. The evaluation will be done within a framework originally developed for evaluating the notations of methodologies for information systems development. We will develop two ontologies, one for analysis and one for design, to be used for evaluating the temporal ER model.

4 An Ontologically-based Evaluation Framework

This section presents the evaluation framework and the ontologies to be used when evaluating the temporal ER models. First, we define the overall framework in which the proposed evaluation will occur. Second, we develop the ontology that will be used when evaluating the models with respect to analysis, that is, for describing the temporal aspects of the modeled reality. Third, we develop the ontology that will be used when evaluating the models with respect to design.

4.1 Overall Framework

Wand and Weber have developed a framework for evaluating the ontological expressiveness of the notations of methodologies for analysis and design of information systems (ISAD) [24] and this framework has been used to evaluate NIAM [25].

One part of the framework is a representational model of the real world. This model consists of all the real-world constructs, called ontological constructs, that an ISAD notation must be able to describe in order to model the real world. Since our evaluation will focus on temporally extended ER models' capabilities in modeling temporal aspects of the reality and a database design, we will develop our own representational models.

The evaluation of whether or not a notation is ontologically expressive is based on the notion of mathematical mappings. The focus is on two sets: the set of ontological constructs and the set of notational constructs that can be obtained from the description of the model to be evaluated. Two mappings exist between these, the representation mapping, which maps ontological constructs to corresponding notational constructs, and the interpretation mapping, which maps notational constructs to corresponding ontological constructs, see Figure 1.

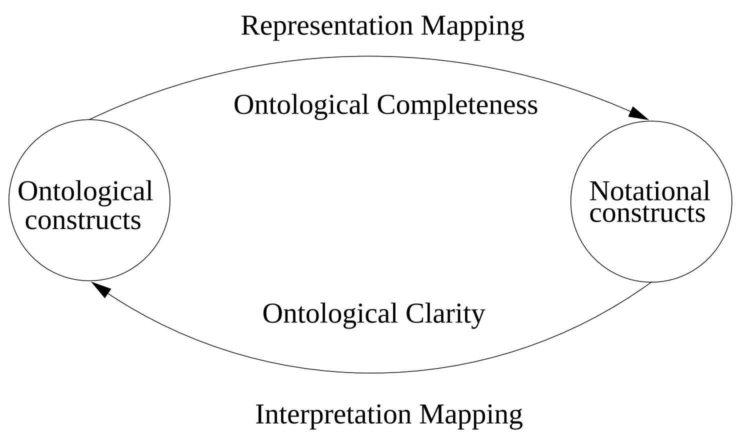

Figure 1: Sets and Mappings in the Evaluation Framework [24]

Informally, the notation of a model is ontologically complete if the notation can represent the same information as the representational model (the ontological constructs); otherwise, it is ontologically incomplete and suffers from construct deficit [24]. More formally, a notation is ontologically complete if the representational mapping from the ontological constructs to the notational constructs is total. The definitions presented in the following are slightly revised versions of the definitions presented by Wand and Weber [24].

Definition 4.1 Ontological completeness: Let f be a mapping from the set of ontological constructs O_c to the set of constructs M_c of a temporal ER model, M. Then M is *ontologically complete* if f is a total mapping. Otherwise, ontological incompleteness, or construct deficit, exists.

Ontological clarity concerns the interpretation mapping from the notational constructs to the ontological constructs. Three situations can obscure the clarity of a notation. First, if one notational construct can be mapped into more than one ontological construct, this implies construct overload. Second, if more than one notational construct can be used to model the same ontological construct, the notation suffers from construct redundancy. Third, if there are notational constructs that do not represent any ontological constructs, the notation suffers from construct excess.

Definition 4.2 Ontological clarity: Let g be a mapping from the set of notational constructs M_c of a temporal ER model, M, to the set of ontological constructs O_c. Then M has *construct overload* if g is a one-to-many mapping, has *construct redundancy* if g is a many-to-one mapping, and has *construct excess* if g is a partial mapping.

4.2 Real-World Ontology

In this section, we define the real-world ontology; in doing so, we will focus on ontological constructs that relate to temporal aspects.

The modeled reality consists of objects. An object is a thing that has independent existence and can be separated from other objects; hence, a data model used for modeling the reality should provide means of conveniently modeling the existence and unique identification of objects. The time during which an object exists in the reality, we call the *existence time* of the object.

An object is characterized by its properties. At any given point in time, each property of an object has a value. The values of some properties remain unchanged over time while others vary, that is, at different points in time, the values of a property for an object may be different. The time during which it is true (in the modeled reality) that a property has a specific value is called the *valid time* of that particular value.

Objects may be interrelated via relations that may or may not vary over time. Different types of relations exists. Some relations are associative in nature, e.g., an employee of some company is associated with some department within the company. Other relations are more generali-zing/specializing in nature, e.g., the staff at a university might be specialized into administrative, academic, and technical staff.

Explicit *constraints* on how an object may participate in the different types of relations may be expressed. For example, a company might have the policy that an employee should at any time be associated with one department, and must during her employment be associated with at least one, but no more than four different departments. The constraints that must hold for all points in time, we will call *temporal*, while we will term the constraints that must hold at each point in time in isolation *snapshot* constraints.

To summarize, the ontological constructs that are needed to describe the temporal aspects of the modeled reality include existence times of objects, valid time of attributes and relationships, and temporal and snapshot relation constraints. Note that transaction time is not a part of the real-world ontology, since transaction time captures when database objects are stored in the database, which is an aspect separate from the modeled reality.

4.3 Relational Ontology

When a conceptual model is used for database design, the predominant target model is the relational model. This means that what is to be described is the underlying relational database schema. Since we examine temporal ER models, we will expect the target model to be the relational model or a temporal extension of it.

In a relational database collections of tuples are stored in relations which form the database. The tuples in a relation are identified by a primary key.

Tuples that are stored in the same relation have the same set of attributes, and each attribute is defined over some domain. A relation may capture the valid time of its tuples by using *time attributes* defined over an appropriate *time domain*. The time during which a tuple is current in the database is called the *transaction time* of the tuple, and a relation may also time attributes that capture this aspect. Some relations may contain attributes which are foreign keys and thus represent references to other relations in the database. A set of integrity constraints are defined over the relations, i.e., entity integrity constraints and referential integrity constraints [3]. *User-defined constraints* can also be a defined over the relations, e.g., how many tuples in one relation are allowed to have references to the same tuple in another relation, both at each point in time (snapshot) and for all points in time (temporal).

Thus, the ontological constructs needed to describe the temporal aspects of a relational database schema include time domains, lifespan attributes, valid-time attributes, transaction-time attributes, and snapshot and temporal user-defined constraints.

5 Model Evaluation—Analysis

This section examines the ontological completeness and clarity with respect to the real-world ontology developed in Section 4.2 of three temporally extended ER models. The notations of the three models are presented by diagrams modeling the company database described in Example 5.1. Because we will present (almost) all the notational constructs of each model, the diagrams may contain notational construct that are used entirely for design and will only become relevant in the next section.

Example 5.1 This text describes requirements for a company database. Each department has a number, a name, some locations, and is responsible for a number of projects. The company keeps track of when a department is inserted and deleted. It also keep track of the various locations of a department. A department keeps track of the profits it makes on its projects. Because the company would like to be able to make statistics on its profits, each department must record the history of its profits over periods of time.

In addition to an ID and a budget, each project has a manager and some employees who work for the project. The company registers the history of the budget of a project. Each project is associated with a department that is responsible for the project. Each employee belongs to a single department throughout his or her employment. For each employee, the company registers the ID, name, date of birth, and salary. The company also records the history of employments. The departments would like to keep records of the different employees' salary histories. For reasons of accountability, it is important to be able to trace previous records of both profits and salaries.

Employees work on one project at a time, but employees may be reassigned to other projects, e.g., because a project may require employees with special skills. Therefore, it is important to keep track of who works for which project at a given time and what times employees are expected to finish working on their current project. Some employees are trainees, some are project managers. Once a manager is assigned to a project, the manager will manage the project until it is completed or otherwise terminated. □

5.1 The ERT Model

The first model to be examined is the Entity-Relation-Time (ERT) model [21].

In the description of ERT that follows, we will use the term class instead of the term type (to be consistent with reference [21]). The model offers support for lifespans of entities and the valid time of binary relationships. Figure 2 is an ERT diagram modeling the database of Example 5.1. A rectangle represents an entity class. An entity class expanded with a "timebox" containing the symbol T is specified as temporal, which indicates that the lifespans of the entities represented by the entity class are to be captured. A value class is represented by a rectangle with a black triangle placed in the bottom-right corner. Value classes represent properties, and only entity classes can be related to value classes. Entity and value classes can be specified as complex by using a double rectangle.

User-defined relationship classes are denoted by small filled rectangles, and only binary relationships are available. These can be specified as temporal by expanding the filled rectangle with a timebox. For each entity (or value) class participating in an user-defined relationship class, an involvement role and a cardinality constraint must be specified. An ISA relationship class is denoted by a circle with arrow(s)

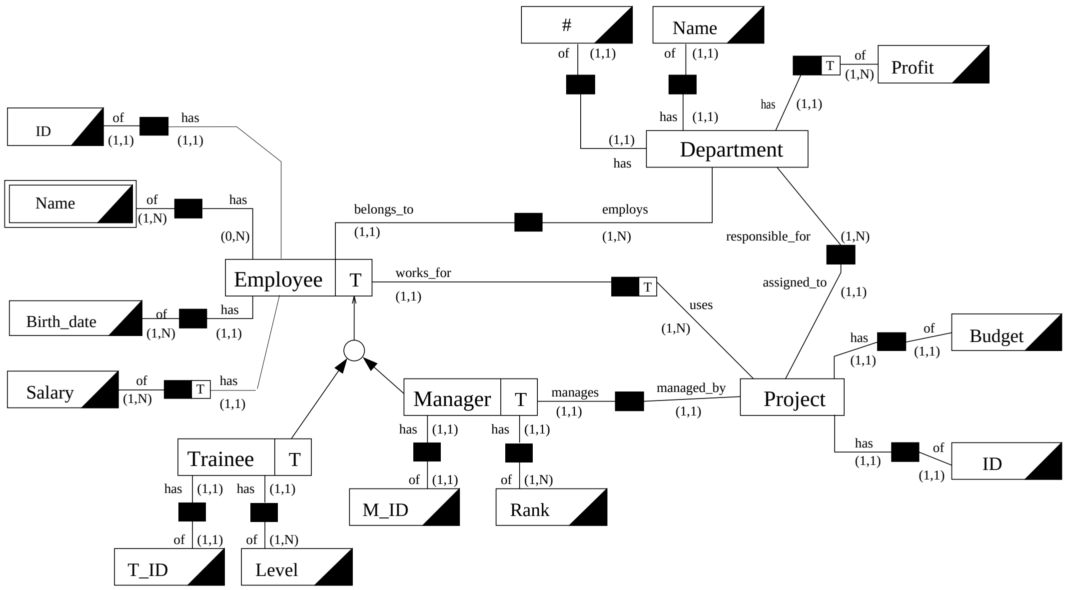

Figure 2: An ERT Diagram Describing a Company Database

flowing from the subclass(es) to the circle and an arrow flowing from the circle to the superclass. If the circle is non-filled, then a participation constraint of the ISA relationship class is specified as partial, and a filled circle indicates total participation. If more than one arrow is pointing into the circle, then a participation constraint of the ISA relationship class is specified as disjoint. A part_of/component_of relationship class models a relationship between a complex entity and one of its components. The notation is as for user-defined relationship classes, but these relationships are not visible in top-level diagrams.

Ontological Completeness

To examine the ontological completeness of ERT, the ontological constructs are mapped to ERT constructs. If an ontological construct cannot be mapped to a notational construct, then an instance of construct deficit exists. Table 3 presents the result of the evaluation.

The ERT model suffers from one, possible two, cases of construct deficit. First, it is not possible to specify temporal constraints. Second, it is not entirely clear if the cardinality constraint, generalization completeness constraint, and the generalization disjointness constraint of the ERT model specify snapshot constraints since the semantics of these constraints are not explicitly defined. The descriptions of these constraints in in Table 3 represent our interpretations.

Ontological Clarity

Ontological clarity concerns the interpretation mapping that goes from notational constructs to ontological constructs. The results for ERT are given in Table 4.

The ERT model does not suffer from construct redundancy. It is not possible to determine whether or not the model suffers from construct excess because the semantics of the constraints that can be expressed for user-defined relationship classes and ISA relationship classes are not explicitly defined. There is one case of construct overload: The timebox is used for modeling both lifespan and valid time.

Ontological Construct	ERT Representation
Existence times of objects	Represented by expanding the entity class symbol with a timebox containing the symbol T.
Valid time of properties	Represented by expanding the relationship class connecting the value class to the entity class with a timebox.
Valid time of relationships	Represented by expanding the user-defined relationship class with a timebox.
Temporal relation constraints	Not represented.
Snapshot relation constraints	Snapshot cardinality constraints may be represented by the minimum and maximum numbers of times that an entity or value can participate in the relationship in parentheses close to the line connecting the entity and relationship class. Snapshot generalization completeness constraints may be expressed by a non-filled circle (partial participation) or a filled circle (total participation). Snapshot disjointness generalization constraints may be expressed when an entity class has more than one subclass. If more than one arrow is pointing into the same circle, the relationship class is disjoint; relationships among subclasses with separate circles are overlapping.

Table 3: Evaluating ERT With Respect to Ontological Completeness

ERT Construct	Ontological Construct
Timebox	Models the valid time of properties and relationships and the existence time of objects.
Cardinality constraint	Might model a snapshot cardinality constraint, although this is unclear from the model's description.
Superclass/subclass completeness constraint	Might model a snapshot generalization completeness constraint, although unclear from the model's description.
Superclass/subclass disjointness constraint	Might model a snapshot generalization disjointness constraint, although unclear from the model's description.

Table 4: Examining ERT With Respect to Ontological Clarity

5.2 TERC+: A temporal Conceptual Model

This section evaluates the TERC+ model [26], which supports lifespan of entity types and valid time of relationship and attribute types. Figure 3 presents the TERC+ diagram modeling the database described in Example 5.1.

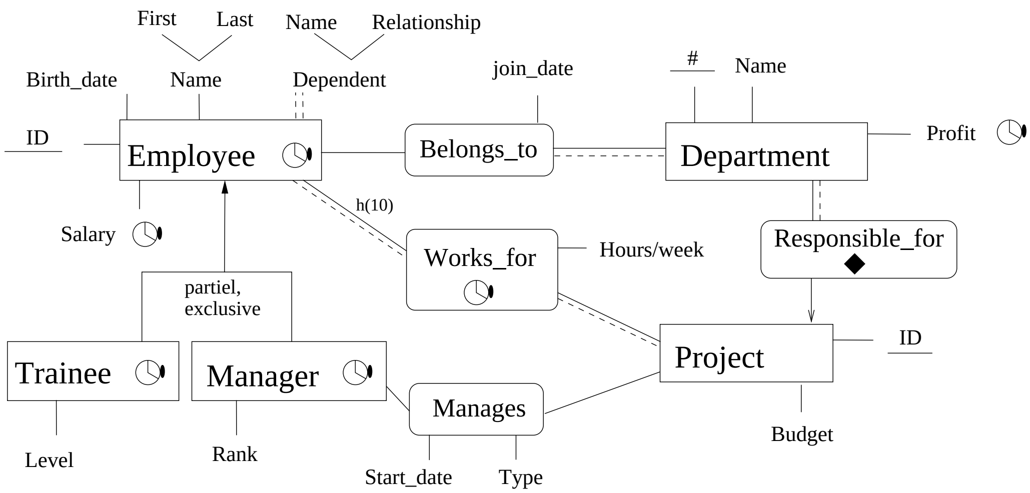

Figure 3: A TERC+ Diagram Describing a Company Database

An entity type is represented by a rectangle, and a relationship type is denoted by a rectangle with rounded corners. Entity types and relationship types can be annotated with a clock symbol to indicate that their life cycles are to be captured. Attributes of entity types and relationship types represent properties. Attributes are represented by plain text linked to entity or relationship types. They may be simple or composite. In addition, attributes can be either single-valued (a full line link) or multi-valued (a dashed line and a full line link). Attributes can be annotated with a clock symbol to indicate that valid time is to be captured. Key attributes are underlined.

Cardinality constraints are expressed using the lines connecting the entity types to the relationship types. Four cardinality constraints can be specified: (0,1) is represented by a single dashed line; (1,1) is represented by single full line; (0,n) is represented by a double dashed line; and (1,n) is represented by a dashed and a full line in combination. A historical cardinality constraint, h(*max*), can be expressed for relationship types. Superclass/subclass relationships are denoted by arrows flowing from the subclasses to the superclasses. A part_of/component_of relationship type is denoted as a relationship type annotated with a filled diamond and an arrow pointing to the component.

Ontological Completeness

The evaluation of the ontological completeness of the TERC+ model is presented in Table 5.

The TERC+ model suffers from one case of construct deficit since it is does not provide notation for specifying temporal constraints. The TERC+ construct that specifies historical participation constraints, (h(*max*)), does not specify a temporal constraint because it only limits the maximum number of relations that one object can participate in during its lifespan. It is not possible to state a minimum number.

Ontological Clarity

The ontological clarity of the TERC+ model is described in Table 6.

Ontological Construct	TERC+ Representation
Existence time of objects	Entity types can be annotated with a clock symbol to indicate that the existence times of the entities represented by the entity type are to be captured.
Valid time of properties	Attributes can be annotated with a clock symbol to indicate that the valid time of the attribute is to be captured.
Valid time of relationships	Relationship types can be annotated with a clock symbol to indicate that the valid times of the relations represented by the relationship type are to be captured.
Temporal relation constraints	Not represented.
Snapshot relation constraints	Snapshot cardinality constraints are expressed using the lines connecting the involved entity types to the relationship types. Snapshot reducible disjointness generalization constraint can be expressed by annotating the n-tailed arrows pointing from the subclass(es) to the superclass with words such as “exclusive.” Snapshot reducible generalization completeness constraints are represented by annotating the n-tailed arrows pointing from the subclass(es) to the superclass with words such as “total.”

Table 5: Evaluating TERC+ With Respect to Ontological Completeness

TERC+ Constructs	Ontological Construct
Clock symbol	Models valid time of properties and relationships and existence time of objects.
Cardinality constraint	Model a snapshot cardinality constraint.
Historical cardinality constraint	No corresponding ontological constructs.
Total generalization constraint	Models a snapshot generalization completeness constraint.
Exclusive generalization constraint	Models a snapshot generalization disjointness constraint.

Table 6: Examining TERC+ With Respect to Ontological Clarity

The TERC+ model does not have redundant modeling constructs, but has one case of constructs excess because the temporal cardinality constraint, (h(*max*)), cannot be mapped to any ontological construct. At first, it seems like it could be mapped to a temporal associative relation participation constraint, but the constraint does not specify any minimum, and no default value for the minimum is implied. There is one case of construct overload since the clock symbol models both existence and valid time.

5.3 The TIMEER Model

The last model to be evaluated is the Time Extended ER (TIMEER) model. The model offers support for lifespans of entity types, valid time for attributes and relationship types, and transaction time for entity types, relationship types, and attributes. Figure 4 presents a TIMEER diagram corresponding to the database in Example 5.1.

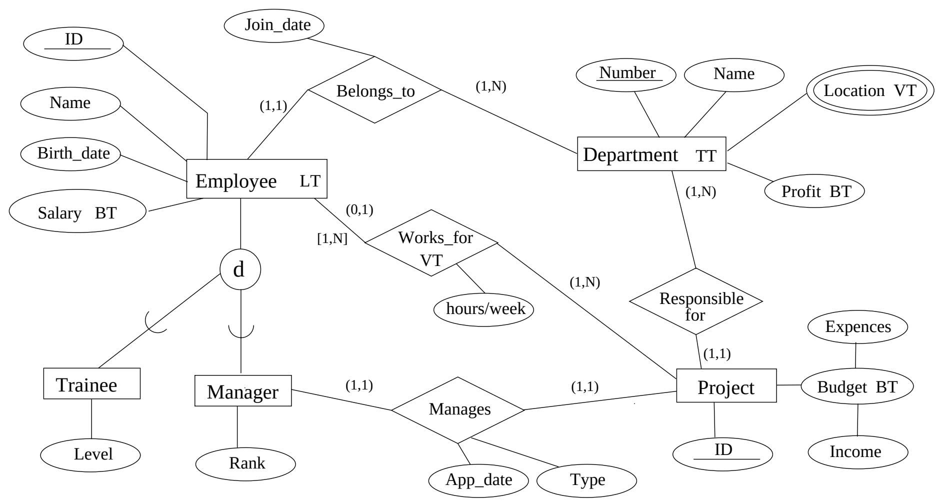

Figure 4: A TIMEER Diagram Describing a Company Database

Entity types are represented by rectangles and can be annotated with an LS, to indicate that the lifespans of the entities represented by the entity type are to be captured. Attributes of entity types (or relationship types) are represented by ellipses. Attributes can be atomic or composite, and single or multi-valued (double ellipse). Key attributes are underlined. An attribute can be annotated with a VT, meaning that valid time of the attribute is to be captured.

A relationship type is represented by a diamond and is annotated with a VT if the valid time of the relationship type is to be captured. For each entity type participating in a relationship type, a snapshot participation constraint has to be specified. This constraint is represented by placing, in parentheses, the minimum and maximum number of relations an entity of the involved entity type can participate in close to the line connecting the involved entity type and relationship type. Lifespan participation constraints are represented by placing the minimum and maximum number of an entity's participation in the relationship in square brackets ([min, max]) close to the line connecting the involved entity types and the relationship type. The meaning of the lifespan participation constraint is that for all of time, any entity of the entity type must participate in at least "min" and at most "max" relations of the relationship type. Superclass/subclass relationships are denoted by a circle with lines connecting the subclasses and the superclass. A double line connecting the superclass and the circle indicates a total superclass/subclass relationship, and a single line indicates a partial relationship. By annotating the circle with an **o** or a **d** the superclass/subclass relationship is specified as overlapping or disjoint, respectively.

Ontological Completeness

In this section we examine whether or not TIMEER is ontologically complete. From Table 7, it can be seen that the TIMEER model does not suffer from construct deficit.

Ontological Construct	TIMEER Representation
Existence time of objects	Entity types can be annotated with an LS to indicate the existence time is to be captured.
Valid time of properties	Attributes are annotated with a VT if valid time is to be captured.
Valid time of relationships	Relationship types are annotated with a VT if valid time is to be captured.
Temporal relation constraints	Represented by Lifespan participation constraints. Temporal generalization completeness constraints are represented by a double line connecting the superclass and the circle in the superclass/subclass relationship type.
Snapshot relation constraints	Represented by snapshot participation constraints. Snapshot disjointness generalization constraints are expressed by annotating the circle in the superclass/subclass relationship type with an **o** (overlapping) or a **d** (disjoint). Snapshot generalization completeness constraints are represented as the temporal specialization completeness constraint.

Table 7: Evaluating TIMEER With Respect to Ontological Completeness

Ontological Clarity

The result of examining the ontological clarity of TIMEER is presented in Table 8.

TIMEER Construct	Ontological Construct
LS	Models that the existence time of objects represented by an entity type is to be captured.
VT	Indicates that the valid time of properties, represented by annotated attribute types and relationship types, are to be captured.
TT	Does not have a corresponding ontological construct.
LT	Does not have a corresponding ontological construct.
BT	Does not have a corresponding ontological construct.
Snapshot participation constraint	Models a snapshot relation participation constraint of associative relations.
Lifespan participation constraint	Models a temporal participation constraint.
Superclass/subclass completeness constraint	Models a temporal generalization constraint and a snapshot generalization constraint.
Superclass/subclass disjointness constraint	Models a snapshot generalization constraint.

Table 8: Examining the TIMEER With Respect to Ontological Clarity

The model does not suffer from construct redundancy. There are three cases of construct excess. The three excessive constructs are TT, LT, and BT, which indicate transaction time support, lifespan and transaction time support, and valid and transaction time support, respectively. There is one case of construct overload: the total specialization constraint. This overload occurs because if the constraint holds over all points in time then it also holds at each single point in time.

5.4 Summary

All three models capture existence time of objects and valid time of attributes and relationships. TERC+ and TIMEER are capable of expressing snapshot constraints, and ERT might also support this. Only TIMEER supports the expression of temporal constraints. All three models are thus well suited for analysis, but if temporal constraints are to be expressed, the TIMEER model should be chosen.

6 Model Evaluation—Design

Having evaluated the models with respect to analysis, we proceed to evaluate their ontological completeness and clarity with respect to design, thus using the relational ontology developed in Section 4.3.

6.1 The ERT Model

Ontological Completeness

As can be seen from Table 9, ERT suffers from three, possibly four, cases of construct deficit. First, the model has no notation for specifying time domains. Second, the model does not support transaction time. Third, no notation is offered for specifying temporal constraints. Fourth, since the semantics of the construct offered for specifying cardinality constraints is unclear, we cannot determine with certainty if the cardinality constraints is a snapshot constraint.

Ontological Construct	ERT Representation
Time domains	Not represented.
Lifespan attributes	Represented by the timebox extension of entity classes.
Valid time attributes	Represented by the timebox extension of user-defined relationship classes.
Transaction-time attributes	Not represented.
Temporal user-defined constraints	Not represented.
Snapshot user-defined constraints	Represented by the placing the min-max constraint near the entity type (or value class) participating in the relationship class.

Table 9: Evaluating ERT With Respect to Ontological Completeness

Ontological Clarity

This section examines ERT with respect to ontological clarity. The result is presented in Table 10.

The model does not suffer from construct redundancy. It is unclear if ERT suffers from construct excess due to the unclear semantics of the constructs offered for specifying constraints. The model suffers from one cases of construct overload: the timebox models both lifespan for entity types and valid time for attributes.

ERT Construct	Ontological Construct
Timebox	Models lifespan attributes and valid-time attributes.
Cardinality constraint	Might model a snapshot user-defined cardinality constraint.
Superclass/subclass completeness constraint	Might model a snapshot user-defined generalization completeness constraint.
Superclass/subclass disjointness constraint	Might model a snapshot user-defined constraint.

Table 10: Examining ERT With Respect to Ontological Clarity

6.2 The TERC+ Model

Ontological Completeness

In this section we examine whether or not TERC+ is ontologically complete with respect to the relational ontology. From Table 11, it can be seen that the model suffers from three cases of construct deficit. First, there is no notation available for specifying time domains. Second, there is no notation offered for specifying the transaction time of attributes. Third, temporal user-defined constraints cannot be specified.

Ontological Construct	TERC+ Representation
Time domains	Not represented.
Lifespan attributes	Represented by annotated entity types with a clock symbol.
Valid-time attributes	Represented by attributes and relationship types annotated with a clock symbol.
Transaction-time attributes	Not represented.
Temporal user-defined constraints	Not represented.
Snapshot user-defined constraints	Represented by the constraint (min, max) that has to be specified for each entity type participating in a relationship type.

Table 11: Evaluating TERC+ With Respect to Ontological Completeness

Ontological Clarity

The result of examining the ontological clarity of TERC+ is presented in Table 12.

TERC+ Construct	Ontological Construct
Clock symbol	Model valid-time or lifespan time attributes.
Cardinality constraint	Models a snapshot user-defined cardinality constraint.
Historical cardinality constraint	No corresponding ontological constructs.
Total generalization constraint	Models a snapshot user-defined generalization completeness constraints.
Exclusive generalization constraint	Models a snapshot user-defined generalization disjointness constraints.

Table 12: Examining TERC+ With Respect to Ontological Clarity

The model does not suffer from construct redundancy, but suffers from construct overload since the clock symbol models both valid-time and lifespan attributes. The model suffers from construct excess since the historical cardinality constraint cannot be mapped to any ontological construct.

6.3 The TIMEER Model

Ontological Completeness

Moving on to the TIMEER model, Table 13 shows that it is ontologically complete with respect to the relational ontology.

Ontological Construct	TIMEER Representation
Time domains	The domain of a time attribute is indicated by the annotation used. If the annotation is an LS then the time domain is the lifespan time domain.
Lifespan attributes	Entity types annotated with an LS (LT) model the presence of lifespan attributes.
Valid-time attributes	The annotations VT and BT indicate the presence of valid-time attributes.
Transaction-time attributes	The annotations TT, LT, and BT indicate the presence of valid-time attributes.
Temporal user-defined constraints	These are represented by the lifespan participation constraint [min, max] for relationship types.
Snapshot user-defined constraints	These are represented by the snapshot participation constraint (min, max) for relationship types.
	Represented by the superclass/subclass completeness constraint.
	Represented by the superclass/subclass disjointness constraint.

Table 13: Evaluating TIMEER with Respect to Ontological Completeness

Ontological Clarity

In this section we examine the ontological clarity of the TIMEER model with respect to the relational ontology. It follows from Table 14 that the model does not suffer construct redundancy. However, it suffers from one case of construct overload since the superclass/subclass completeness constraint models both temporal and snapshot user-defined generalization completeness constraints. The model does not suffer from construct excess.

TIMEER Construct	Ontological Construct
LS	Models lifespan attributes.
VT	Models valid-time attributes.
TT	Models transaction-time attributes.
LT	Models lifespan and transaction-time attributes.
BT	Models valid-time and transaction-time attributes.
Snapshot participation constraints	Model snapshot user-defined constraints.
Lifespan participation constraints	Model temporal user-defined constraints.
Superclass/subclass completeness constraints.	Model temporal and snapshot user-defined generalization completeness constraints.
Superclass/subclass disjointness constraints.	Model snapshot user-defined generalization disjointness constraints.

Table 14: Examining the TIMEER With Respect to Ontological Clarity

6.4 Summary

All three models offer support for capturing lifespan and valid-time attributes and snapshot constraints (although the semantics of ERT's constraints are unclear). Only TIMEER is able to model transaction-time attributes and temporal constraints. Since transaction-time support is frequently necessary in applications, ERT and TERC+ fall somewhat short in supporting the modeling of a database design.

7 Summary and Research Directions

At the outset, the paper characterizes temporal ER models by describing the interrelation between the modeling constructs offered by the ER model and the temporal aspects, lifespan, valid time, and transaction time, which are candidates for being given built-in support in the ER model. This leads to a cross-tabulation indicating which modeling constructs may be associated with which temporal aspects, thus offering a basis for obtaining maximal and meaningful temporal support.

This is followed by a presentation of an overall framework for examining whether or not a temporal ER model is ontologically expressive. The framework includes two ontologies to be used as so-called representational models. The real-world ontology describes real-world constructs necessary to model the temporal aspects of reality, and the relational ontology describes constructs necessary to capture the temporal aspects of a relational database schema corresponding to a temporal ER diagram. The framework is used for evaluating three temporal extensions of the ER model, namely the ERT, the TERC+, and the TimeER model, with respect to their use for analysis as well as design.

The overall result is that no model is ontologically expressive with respect to either ontology. In addition, the ERT and TERC+ models are ontologically incomplete and ontologically unclear with respect to both the real-world and the relational ontologies. None of these models offer built-in support for transaction time. The TIMEER model is ontologically complete, but also ontological unclear with respect to both ontologies. The model supports all the three temporal aspects considered. As a result, TIMEER makes it possible to model the temporal aspects, as covered by the ontologies, of both reality and a relational database schema. With the ERT and TERC+ models, it is possible to model all the temporal aspects of reality that are considered, but not those of a relational database schema. This leads to the conclusion that these two models do not fully support the conceptual design of databases managing time-varying data.

The research reported in this paper points to several promising directions for future research, relating to the design conceptual models and to the evaluation of these.

It is recommended that the ERT and TERC+ models be enhanced with support for transaction time. Next, it appears relevant to consider extending the ERT and TIMEER models with support for for modeling dynamic aspects of reality; TERC+ already offers some such support. In addition, new support for capturing database behavior might be introduced in all three models. Indeed, the idea of being able to specify, at the time of conceptual design, known changes to the database schema in the diagram documenting the database schema appears to be very appealing. That is, it should be studied how to extend these and other conceptual models with notational construct that conveniently capture the evolution of the database schema over time.

As another topic, the evaluation framework itself might be expanded. The outcome of an evaluation is quite sensitive to the ontologies employed, making it an interesting direction to expand the ontologies to capture better structural aspects not related to time, and perhaps also to capture dynamic aspects. More generally, it is felt that there is a need for more methods that systematically evaluate and compare extended ER models. Such methods are likely to prove useful to both the designers of new conceptual models and the users of conceptual models.

Acknowledgements

This work was supported in part by grants 9502695 and 9700780 from the Danish Technical Research Council and grant 9701406 from the Danish Natural Science Research Council, as well as grants from the Nykredit corporation and the Danish Centre for IT Research.

Acknowledgements

This work was supported in part by grants 9502695 and 9700780 from the Danish Technical Research Council and grant 9701406 from the Danish Natural Science Research Council, as well as grants from the Nykredit corporation and the Danish Centre for IT Research.

References

[1] I. Brandt. A Comparative Study of Information Systems Design Methodologies. In T.W. Olle, H.G. Sol, and C.J. tully, editors, *Information Systems Design Methdologies: A feature Analysis*, pages 9–36. Elsevier Science Publishers, 1983.

[2] R. Elmasri, I. El-Assal, and V. Kouramajian. Semantics of temporal data in an extended er model. In *9th International Conference on the Entity-Relationship Approach*, pages 239–254, Lausanne, Switzerland, October 1990.

[3] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, 2. edition, 1994.

[4] R. Elmasri, G. Wuu, and V. Kouramajian. A temporal model and query language for eer databases. In A. Tansel et al., editor, *Temporal Databases: Theory, Design, and Implementation*, chapter 9, pages 212–229. Benjamin/Cummings Publishers, Database Systems and Applications series, 1993.

[5] S. Ferg. Modeling the Time Dimension in an Entity-Relationship Diagram. In *4th International Conference on the Entity-Relationship Approach*, pages 280–286, Silver Spring, MD, 1985.

[6] C. Floyd. A Comparative Evaluation of Systems Development Methods. In T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart, editors, *Information Systems Design Methdologies: Improving the Practice*, pages 19–54. Elsevier Science Publishers, 1986.

[7] H. Gregersen and C. S. Jensen. Conceptual Modeling of Time-varying Information. Technical Report TR-35, TimeCenter, September 1998.

[8] H. Gregersen and C. S. Jensen. Temporal Entity-Relationship Models—a Survey. *IEEE Transactions on Knowledge an Data Engineering*, 11(3):464–497, May 1999.

[9] H. Gregersen, C. S. Jensen, and L. Mark. Evaluating Temporally Extended ER Models. In K. Siau, Y. Wand, and J. Parsons, editors, *Proceedings of the Second CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, page 12, June 1997.

[10] R Hull and R. King. Semantic Database Modeling: Survey, Applications, and Research Issues. *ACM Computing Surveys*, 19(3):201–260, September 1987.

[11] N. Jayaratna. *Understanding and Evaluating Methodologies: NIMSAD – A Systematic Framework*. Information syetems, Management and Strategy Series. McGraw-Hill, 1994. ISBN: 0077078829.

[12] M. R. Klopprogge and P. C. Lockeman. Modeling information preserving databases: Consequences of the concept of time. In *Proceedings from the Ninth International Conference on Very Large Data Bases*, pages 399–416, October 1983.

[13] C.H.. Kung. An Analysis of Three conceptual Models with Time Perspective. In T.W. Olle, H.G. Sol, and C.J. tully, editors, *Information Systems Design Methdologies: A feature Analysis*, pages 141–167. Elsevier Science Publishers, 1983.

[14] A.H.F. Laender and D.J. Flynn. A Semantic Comparison of the Modeling Cababilities of the ER and NIAM Models. In R.A. Elamsri, V. Kouramajian, and B. Thalheim, editors, *Entity-Relationship Approach — ER'93*, volume 823 of *Lecture Notes in Computer Science*, pages 242–256. Springer Verlag, 1993. ISBN: 0-387-58217-7.

[15] V. S. Lai, J-P. Kuilboer, and J. L. Guynes. Temporal Databases: Model Design and Commercialization Prospects. *DATA BASE*, 25(3):6–18, 1994.

[16] A. Narasimhalu. A Data Model for Object-Oriented Databases with Temporal Attributes and Relationships. Technical report, National University of Singapore, 1988.

[17] J. Peckham and F. Maryanski. Semantic Data Models. *ACM Computing Surveys*, 20(3):153–189, September 1988.

[18] M. Schrefl, A.M Tjoa, and R.R. Wagner. Comparison-Criteria for Semantic Data Models. In J.K. Aggarwal, editor, *Proceedings of the First International Conference on Data Engineering*, pages 120–125, 1984.

[19] R. T. Snodgrass. The Temporal Query Language TQuel. *ACM Transaction on Database Systems*, 12(2):247–298, June 1987.

[20] B. Tauzovich. Toward Temporal Extensions to the Entity-Relationship Model. In *The 10th International Conference on the Entity Relationship Approach*, pages 163–179, San Mateo, California, October 1991. E/R Institute.

[21] C. I. Theodoulidis, P. Loucopoulos, and B. Wangler. A Conceptual Modelling Formalism for Temporal Database Applications. *Information Systems*, 16(4):401–416, 1991.

[22] Y Wand and R Weber. An Ontological Evaluation od Sytems Analysis and Design Methods. In E. D. Falkenberg and P. Lindgreen, editors, *Information SystemsConcepts: An In-depth Analysis*, pages 79–107. North-Holland, Amsterdam, 1989.

[23] Y Wand and R Weber. Toward a Theory of The Deep Structure of Information Systems. In J. I. DeGross, M. Alavi, and H. Oppelland, editors, *International Conference on Information Systems*, pages 61–71, 1990.

[24] Y Wand and R Weber. On the Ontological Expressiveness of Information Systems analysis and Design Grammars. *Journal of Information Systems*, 3(4):217–237, 1993.

[25] R. Weber and Y. Zhang. An Analytical Evaluation of NIAM's Grammar for Conceptual Schema Diagrams. *Information Systems Journal*, 6(2):147–170, 1996.

[26] E. Zimanyi, C. Parent, S. Spaccapietra, and A. Pirotte. TERC+: A Temporal Conceptual Model. In *Proc. Int. Symp. on Digital Media Information Base*, November 1997.