CISMOD 92, Bangalore, India
Pages 244-260

An Entity-Relationship Model Extended To Describe Historical Information

Peter M^{c} Brien
Imperial College
Dept. of Computing
180 Queen's Gate
London SW7 2BZ
U.K.

pjm@doc.ic.ac.uk

Anne Helga Seltveit
Div. of Computer Science
University of Trondheim
N-7034 Trondheim-Nth
Norway

ahs@idt.unit.no

Benkt Wangler
SISU
PO Box 1250
16428 Kista
Sweden

benkt@sisu.se

Abstract

The entity-relationship (ER) model has proved a successful conceptual capture and modelling tool for the relational data model. Much effort has recently been made to extend the relational data model to describe historical information, but as yet little corresponding development in ER modelling has been made. We describe in detail the various temporal behaviours that entities and relationships may exhibit, and apply the results to enhance a binary ER model with temporal semantics. The resultant ERT may be used to fully model historical relational database schemata.

Keywords: Semantic Data Models, Entity-Relationship Models, Temporal Databases, Historical Databases.

Introduction

The semantical data models of the past have focused on data representation without considering in depth the temporal and behavioural aspects of information modelling. A large proportion of research into temporal data models and query languages has been in the context of the relational data model, and only to a lesser extent has it covered the field of semantical data models. In this paper we explore the enhancements to a binary entity-relationship model necessary to fully model the temporal aspects of a historical relational database.

When adding the temporal dimension to models belonging to the ER family, we must consider

- enhancing the relational database model to incorporate temporal information, and
- augmenting of graphical and textual formalisms of the ER model with time concepts so that temporal requirements, integrity constraints and other rules involving time can be specified.

Adding the Temporal Dimension to the Relational Model

The topic of adding the temporal dimension to the relational model has received considerable attention in the database literature, a recent survey of which may be found in [McK91]. It has made clear that two types of temporal information must be recorded in addition to attributes/tuples to fully capture the evolution of an attribute/tuple over time [Sno85], these being

- the *event time*, which records as a series of time points $E = [e_1, e_2, ...]$ the time over which we know (or think) a piece of information holds in the *universe of discourse* (UoD), and

- the *transaction time*, which records as a series of time points $T = [t_1, t_2, ...]$ the time over which the information holds in the *information system* (IS).

For simplicity, we store a continuous series of time points as a pair of time points representing the *interval* of the series. For example, we may want record in our IS on the 3rd January that '*x* worked for company *y*' from 1st January to the 31st, and remove that information at the end of the year. Then for '*x* worked for company *y*' we can state that the interval E = [1st January,31st January] and the interval T = [3rd January,31st December].

A database which models only event time is termed a *historical database*, and one which models both event and transaction time is termed a *temporal database* [Sno85]. In practice, the event time is of most interest, since the modelling of the UoD is what is usually the objective of an IS. The transaction time is only of use in 'meta' rules which review the activities of the IS, which would need to know what information the system used at any point in its execution.

A large volume of work has been reported on the subject of historical relational databases, those with an algebraic basis being surveyed in [McK91]. Most of these algebras involve specifying a particular data model for the temporal information stored, but in [Tuz90] a more general *temporal structure* was introduced, which describes any discrete temporal model. In the temporal structure, a historical database $\mathbf{D_T}$ is considered as a set of relational databases $\{ \mathbf{D}_t \mid 0 \leq t \leq \tau \}$ where the subscript t denotes the time associated with a particular database. Each member of $\mathbf{D}$ has the same schema as $\mathbf{D_T}$, a restriction which may be avoided in practice by taking the schema of $\mathbf{D_T}$ to be the union of the differing schemata needed at various $\mathbf{D}_t$. By restricting t to some finite subset of the natural numbers $0 \ldots \tau$ we achieve a bounded model, and the usual ordering on the natural numbers achieves a linear model. We will refer to each instance of t as a *tick*. The temporal algebras in [Tuz90,Gab91] can query the temporal structure. These algebras can be linked to the temporal US-Logic, which has been widely studied, and has a high expressive power.

Modelling the Temporal Dimension in the ER Model

The temporal structure finds an obvious equivalent in the ER model, by having a series of ER models over the ticks, where the structure of the model is the same at each tick. The instances of each entity and attribute may vary over time, as well as the involvement of entities in relationships. The purpose of the *ERT* model described in this paper is to allow for the graphical representation of these variations. We shall also define the information that it is necessary to represent in any relational database schema which implements an ERT model.

To set the results presented in this paper in perspective, it is worth reviewing some earlier work in modelling temporal information with the ER model. Ferg [Fer85] proposed that relationships could be time-stamped with explicit start and end attributes to restrict their period of validity. So that the attributes of entities could also be modelled as valid for certain periods of time, it was also proposed that the attributes of an entity could be graphically modelled as distinct domains from the entity, with a relationship placed between them.

This approach was refined in the first version of the Tempora ERT model (entity-relationship with time) [The91], where a 'T' marking was placed on relationships when the time-stamp was required, all attributes of entities were represented as graphically distinct *value classes*, and the entities themselves were permitted to be time-stamped.

Elmasri and Wuu [Elm90a] when proposing a temporal extension to the EER [Elm89] model also incorporate the idea of life-span for entities and relationships, and in [Elm90b] this was extended by representing the various temporal roles of a *conceptual entity*. Every object is time-stamped, and a distinction is made between *conceptual objects* and *temporal objects*. The start-time of a conceptual object corresponds to the time point when the concept materializes in the real world. After that the object will stay in existence forever, with the motivation that concepts never die. By contrast, a temporal object has an existence period corresponding to the time it plays a specific role in the reality, e.g. the time period that a certain *person* (conceptual entity) plays the role of a *living-person* (temporal entity). A similar distinction is drawn between non-temporal and temporal relationships.

By contrast, in the original ERT model, only the entities and relationships that are time-varying and for which you want to record the history will receive a time-stamp, which is interpreted as representing their period of existence in the real world. Thus it may be realized that the ERT restricts its descriptions to historical information. By contrast the EER mixes notions of transaction time and event time in its notion of a conceptual object. Note that in the ERT non time-stamped entities are considered as lasting forever, and so roughly correspond to the conceptual entities of [Elm90b]. ERT time-stamped entities and relationships stay in the database for any period of time (the transaction time is not represented); it is only their validity in the UoD that ends, and hence they may last forever in the memory of the system.

The approach in this paper refines definitions in the original ERT model in two respects:

- the definitions of the temporal objects are given mathematically, by specifying what constraints are placed on the sets of event time points of an object when it has some temporal marking applied to it. This will allow for future work to make a more formal connection between the ERT and the temporal structure.

- the temporal markings are used to represent the temporal variation of objects with respect to each other, instead of simply modelling that some temporal variation occurs. In particular, we relate the period which relationship involvements can exist to the period for which the associated entities exist, and relate the period of entity subclasses exist to the period that the superclass entity exists. We identify that there are two distinct aspects to the temporal nature of a relationship involvement, which we shall call *temporal variation* and *historical perspective*.

The use of strict semantics for the temporal components, and the introduction of more specific types of temporal markings for relationships, leads to a more accurate reflection of the UoD. This in turn serves to give constraints on the values that time-stamps may take in the database. These constraints can be used both to derive which relations in the RDB must have time-stamps, and to act as runtime constraints on what value the time-stamps may take. We thus will produce relational database schemata with fewer time-stamps than those produced by the earlier ERT model, and thus simplify the maintenance of temporal information.

The paper is divided into four sections. Section 1 briefly introduces the binary ER model we used as a basis for the temporal extension. Section 2 discusses the manner in which data may both vary over time and be related between times, and then introduces constructs into the ER model to represent these variations. Section 3 describes the constraints and guidelines we place on the use of the temporal constructs, both during the development of the model, and during the run-time use of a database. Finally Section 4 sketches a mapping from the ERT model to a relational schema. A query language for the ERT model has already been described in [McB91].

The ERT Model forms part of the TEMPORA information system development framework, which includes conceptual modelling and software engineering tools, and a run-time platform based around the Sybase RDBMS and BIM-Prolog. An overview of the TEMPORA framework may be found in [Lou90].

1 An Extended ER Model

Since the temporal ER model presented in the next section was designed to describe the temporal characteristics of the Tempora ER model, we will briefly describe the features of that model in this section, and then give an example which will form the basis for the remainder of the discussion in this paper. The constructs of our ER Model are listed in Table 1.

Example 1 : ER Model for people working for companies

The situation described by Figure 1 is as follows, with class and relationship names in italicised typeface:

Each *company* can *employ* any number (0:N) of *persons*, but each *person* works for at most one (0:1) company, and *lives_in* exactly one (1:1) *town*. Each *town* must have at least one (1:N) inhabitant. We know that *persons* divide into two sub-classes of *male* and *female*, and that there are no *persons* which do not belong to one of these sub-classes (hence ●). Every person has up to four (0:4) grandparents (we wish to allow for some people not having their grandparents recorded in the database), and can have any number of grandchildren. The *town has* a *post_code* and *name*, but we will not store the *name* of the *town* (a rule will derive it from the *post_code*). Each *company* has one *name*, and one *address*, which contains details which we wish not to state at this time. Later we may fill in the details of *address* by giving an ER model such as that of Figure 2.

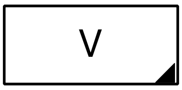
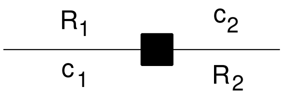
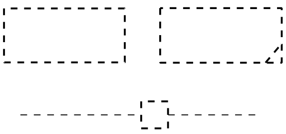
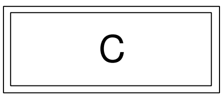
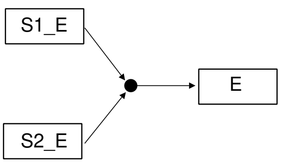
<table>
<tr><th>ER Model Construct</th><th>Description</th></tr>
<tr><td>
</td><td>An entity class E, which denotes a set of objects (concrete or abstract) in the UoD which share the same type of attributes and relationships with other objects.</td></tr>
<tr><td>
</td><td>A value class V, which is a set of lexical objects only of interest when associated with some entity as that entity's attribute.</td></tr>
<tr><td>
</td><td>A relationship R_1, which denotes a set of associations or involvements between an entity E_1, and a value class V_2 or entity class E_2. The cardinality c_1=min:max denotes that each instance of E_1 must have at least min and at most max involvements in R_1. The inverse relationship is described by R_2 and c_2, and involves the same pairs of entities as does R_1.</td></tr>
<tr><td></td><td>Dashed outlines indicate that the objects are derived rather than stored in the database. Rules to derive such objects can be written in any suitable language, a specialized language ERL [Mc B91] is used by the TEMPORA framework. No other changes to the semantics of the diagram are implied.</td></tr>
<tr><td>
</td><td>A complex entity C is an abstraction of an ERT sub-schema, where the only relationships with classes outside the sub-schema are those with entity class C.</td></tr>
<tr><td>
</td><td>Disjoint sub-classes (or specializations) S_1_E and S_2_E of an entity class E may be represented by a relationship with a circular join. If the circle is solid then the sub-classes are total, if open then the sub-classes are partial. (Sub-classes which are not disjoint may be represented by separate arrow lines pointing into E).</td></tr>
</table>

Table 1 : Constructs of the ER Model

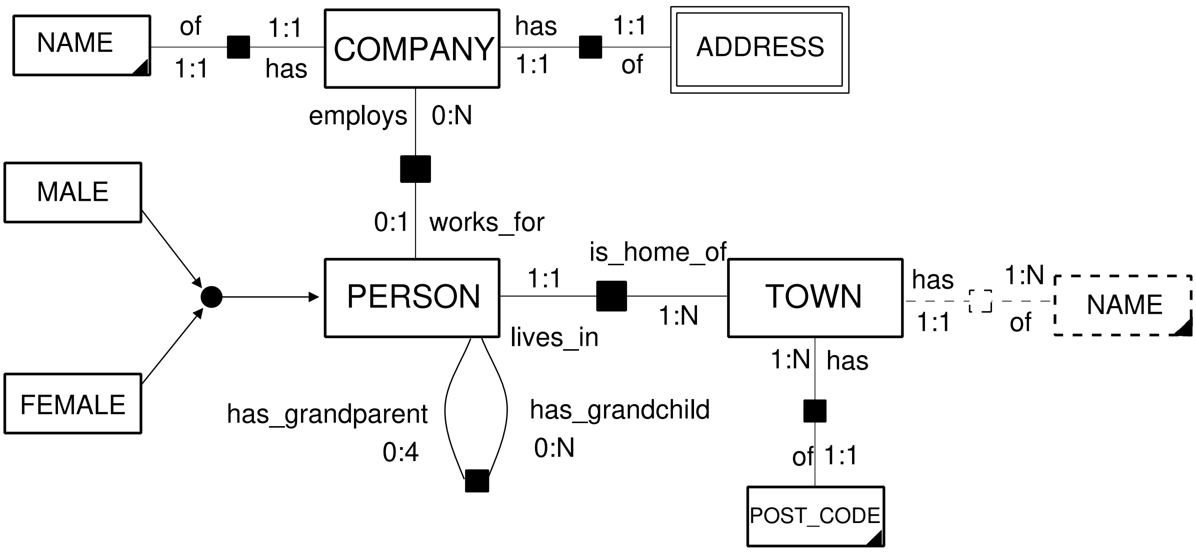


Figure 1 : People working for companies ER model

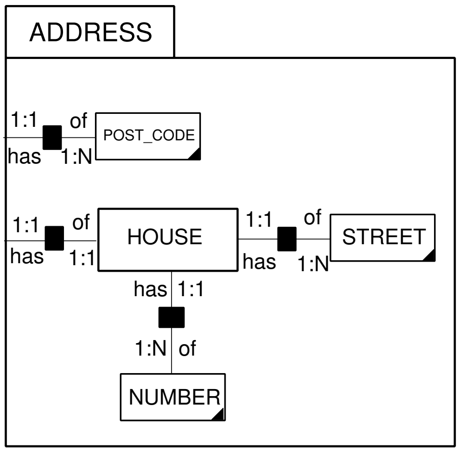


Figure 2 : Details of the *address* complex object

2 Temporal Characteristics of the ER Model

When we record the evolution of the ER model over time, we have a series of models, each of which is associated with a particular tick. In the same way that the temporal structure restricts the relational schema to be the same at each tick, so we will restrict the ER model will be the same at each tick. In the temporal structure, tuples of relations vary over time, in the ER model it is the instances of classes and relationship involvement that may vary in the following ways:

- *Entity Class* instances may exist at certain ticks, and not at others. If this is so, we may say that the entity undergoes *temporal variation*, and denote this on the ER model by adding a T descriptor on to the entity class box.

- A *Relationship* involvement may exist for a subset of the number of the ticks for which both the entities it associates exist, and only associate entities which exist at the same tick.

If this is so, we may say that the relationship undergoes *temporal variation* with respect to the entities it associates, and denote this on the ER model by adding a T descriptor on the relationship line. As for standard relationships, the T-marked relationship R_1 from E_1 to E_2 is symmetric, in that the inverse relationship R_2 (from E_2 to E_1) holds for the same pairs of entities, and at the same ticks, as R_1.

- A *Relationship* involvement may exist at a certain ticks between an entity E_1 which exists at those same ticks, and an entity E_2 which exists at *other* ticks. If this is so, we may say that the relationship has *historical perspective*, and denote this on the ER model by adding a H descriptor on the relationship line. Note that such relationships are asymmetric, since at any tick it is only necessary for E_1 to exist, and so the inverse relationship (from E_2 to E_1) may not hold at the same ticks.

- *Value Class* instances only have meaning when associated with some entity. Thus there is no benefit in recording and modelling the temporal behaviour of value instances themselves, since the only way they may be accessed (relationships) can already recorded.

Before giving more detailed descriptions of the notions introduced above, we must define a notation for describing the ticks when an object (i.e. entity or relationship instance) holds. We may represent the period over which a time varying object x holds as a set $I_x = \{t_a, t_b \dots t_z\}$ where $t_a, t_b \dots t_z$ are the ticks at which it holds. Since the series of ticks concerned is usually continuous, we will call it an *interval*. (What we have actually defined is a set of intervals, but it is convenient to simply refer to this as an interval).

The definition of I_x may be given the shorthand notation of $[t_a, t_z]$ for the case where the series if ticks is continuous. Our definition of an interval allows for the use of the usual set operators for finding interval union, inclusion, *etc.* In our discrete bounded structure the possible ticks of an interval are limited to a finite set $\aleph = \{0 \dots \tau\}$, and so for all objects x the interval I_x will satisfy $I_x \subseteq \aleph$.

In the following definitions, when we speak of the interval I_E of entity class E, we are referring to the intervals of the instances of the entity class, and the definitions given for the properties of the intervals must hold for all instances of the entity class or relationship concerned. Thus making the statement $I_{\mathsf{E}} \subseteq \aleph$ implies the statement that $\forall e \,.\, e \in \mathsf{E} \rightarrow I_e \subseteq \aleph$.

2.1 Entity Temporal Variation

Entities which are not sub-classes of another Entity

The *temporal variation* in entity existence describes the property of an entity only existing at certain times in our UoD. This will be indicated by the use of a T-mark in accordance with Definitions 1 and 2.

Definition 1 Marking an entity class E with T, when E is not a specialization of another entity class, indicates that $I_E \subseteq \aleph$.

Definition 2 An entity class E which is not marked by T, and which is not a specialization of another entity class, has $I_E = \aleph$.

We use the T-mark on an entity class when we find that entity instances possess such temporal variation, and it is of significance to the description of rules and queries about the UoD that it should be recorded. For instance, for the people working for companies ER model (Figure 1), we might have *company* and *person* entities changing over time, and together with rules which use the knowledge about past and future entities (for example, rules which find the *persons* who worked for a *company* in the past). We would thus indicate that the temporal variation of the entities is of significance by marking them with a T-mark in our ERT version of the model in Figure 6. The *town* entity class may be regarded as fixed with respect to the time period of the UoD (the interval $\aleph$) and can be said to always valid for that period. Thus we have no T-mark on the entity class in Figure 6. Note that since all value classes V are not marked, so we always have $I_V = \aleph$.

Figure 3 shows the instances of entities and their relationship involvement of the entities in figure of Figure 1 over the period of ticks {1,2,3}. Since we allow temporal variation of the *company* entity class, it is possible for the class to have instances $\{a_1,a_2,a_3\}$ at ticks 1 and 2, and just instance $\{a_3\}$ at tick 3. This variation is modelled by placing a T-mark on the entity class in Figure 6. The *town* entity class has no temporal variation, so can only keep the same instances $\{c_1,c_2\}$ for all ticks, and hence the class has no T-mark in Figure 6.

Entities which are sub-classes

When entities are subclasses of other entities, we view their temporal properties with respect to the interval of the superclass entity, and not $\aleph$.

Definition 3 An entity class S_E which is not marked by T, and S_E is a specialization of another entity class E, indicates that $I_{S\_E} = I_E$.

Definition 4 An entity class S_E which is marked by T, and S_E is a specialization of another entity class E, indicates that $I_{S\_E} \subseteq I_E$.

If we made the *male* and *female* subclasses of *person* T-marked, then we would be indicating that instances of *persons* are viewed and recorded as sometimes being *male* and sometimes *female*. In general, one may use specialization to model the various states that an entity may assume. For example, an *order* entity, may during its life-time may move through the states of being a *new_order*, a *back_order* and a *delivered_order*. It is obvious that it cannot assume more than one state at a time, and hence these states should be modelled as disjoint subclasses with T-marks.

Changing unmarked Entities

Note that we may still update entities which are not T-marked, but in so doing we loose all information about the past and future. For example, adding instance c_3 to the *town* entity class would lead to the c_3 instance being added to the entity class at all ticks. Even though we do not have T-marked subclasses in this example, it is possible for a *person* to change from *male* to *female* subclasses. However the recorded information will indicate that any particular instance of *person* has belonged to just one subclass for its entire existence.

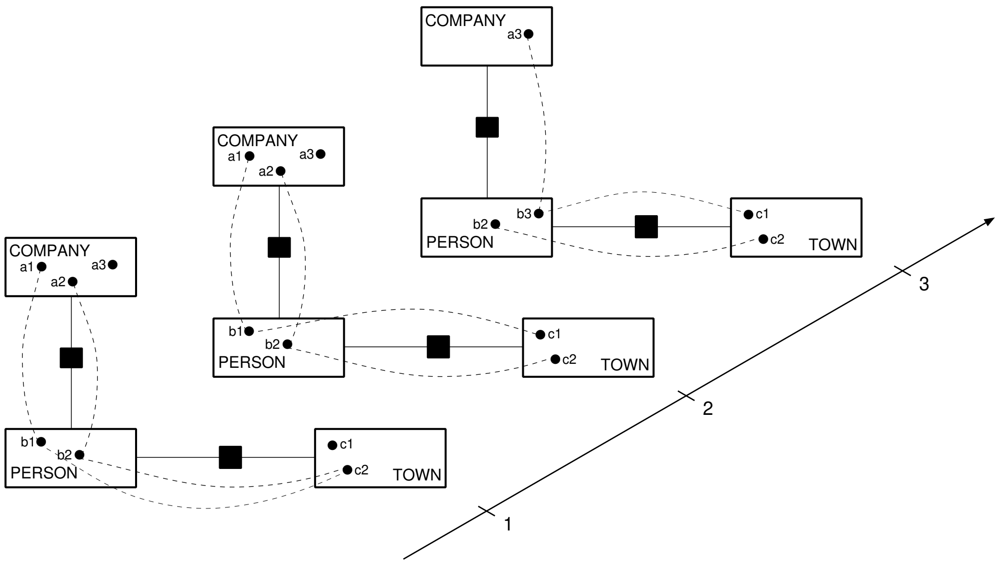


Figure 3 : Temporal variation of entity and relationship involvement

2.2 Relationship Temporal Variation

The *temporal variation* of a relationship describes the changes of relationship involvement that an entity may undergo over its period of existence.

Definition 5 If I_{E1} and I_{E2} are the intervals for which entities of classes E_1 and E_2 exist, and I_R is the interval over which the instances of E_1 and E_2 are involved in T-marked relationship R then $I_R \subseteq I_{E1} \cap I_{E2}$.

Definition 6 If I_{E1} and I_{E2} are the intervals for which entities of classes E_1 and E_2 exist, and I_R is the interval over which the instances of E_1 and E_2 are involved in an unmarked relationship R, then $I_R = I_{E1} \cap I_{E2}$.

Definitions 5 and 6 would also apply if we were to replace one (but not both) of the entity classes by a value class. It is only meaningful to talk about a relationship involving one or two entities about which one knows something, a property normally referred to as *referential integrity*. In this case we have said that both entities must hold at the ticks that we say the relationship involvement holds, by restricting the interval of a relationship to be a subset of the union of the entities it involves. In the next subsection we will deal with the case when just one of the entities holds at the ticks that the relationship holds.

An example of temporal variation in relationship involvement is shown in Figure 3. A *person* may move from *town* to *town*, whilst both the *person* and *town* remain in existence, and thus the *lives_in* relationship is subject to temporal variation. This can be seen in that at tick 1 we have {*b1 lives_in c2*, *b2 lives_in c2*} and at tick 2 {*b1 lives_in c1*, *b2 lives_in c2*}; the *b1* entity has changed its association from *c2* to *c1*. We represent the recording of this movement in the ERT model of Figure 6 by placing a T-mark on the relationship line. If the relationship had not been T-marked, we could still change the involvement of *b1*, but there would be no record of the previous value, and it would appear as if *b1* was associated with *c1* for the entire period that those two entities exist.

2.3 Relationship Historical Perspective

The *historical perspective* of a relationship describes if the two entities' involvement-intervals are distinct periods of time. The semantics of the H-mark are given in Definition 7.

Definition 7 If we have instances of entity classes E_1 and E_2 related by R, then if the instance of E_1 is involved with R over period I_{E1}, E_2 involved for I_{E2}, and relationship R is H-marked then we allow $I_{E1} \neq I_{E2}$.

For example, we may say that one *person* has a grandparent who is another *person*, but these two entities need not exist together for any one tick if the grandparent died before the grandchild was born. Figure 4 illustrates this situation, where we wish to express that b_1 was the grandparent of b_3, but since the two involved entities exist at different times, we must use a relationship which spans from an ER model at some ticks (1 and 2 in the example) to other ticks (3 in the example).

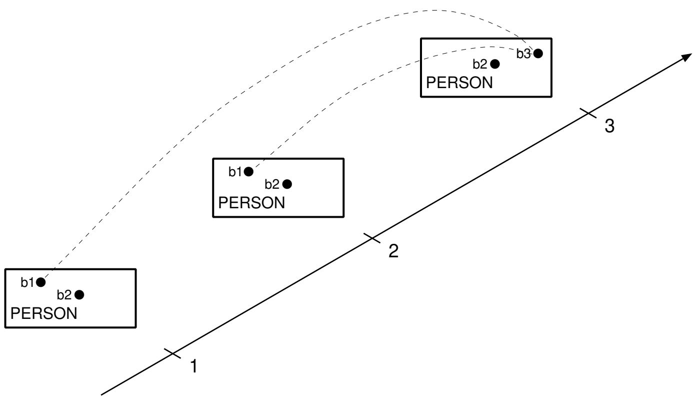


Figure 4 : Historical perspective of the 'has_grandparent' relationship

The historical perspective of a relationship will have *temporal direction*, as described in Definition 8. We can see from the definition that the unmarked relationship is merely a *current* historical perspective relationship. For example, we may say that the *has_grandchild* is a *current* and *future* relationship, since the grandchild can only exist at ticks during or after the ticks when the grandparent existed.

Definition 8 An H relationship of E_1 R E_2 is described as:
• *past* if E_2 holds at ticks before the ticks which E_1 holds at.
• *current* if E_2 holds at the same ticks for which E_1 holds.
• *future* if E_2 holds at ticks after the ticks which E_1 holds at.
• some Boolean combination of the above three items

Using Derivation Rules to Model* H *Relationships

We may describe the characteristics of H-marked relationship E_1 R E_2 by using derived entities. All we need do is construct a derived entity class with all the past, current or future instances of E_2 according to the temporal direction given to R by Definition 8, and use an unmarked relationship to connect the two. Figure 5 illustrates this solution for the *has_grandchild* and *has_grandparent* relationships of Figure 1. The instances of the *past_person* entity at a particular tick are derived from all the past and current *persons* relative to that tick, and the *future_persons* from all the current and future *persons*. Note that the asymmetric nature of an H-marked relationship is now made apparent by the need for two quite distinct unmarked relationships and derived entity classes to model it.

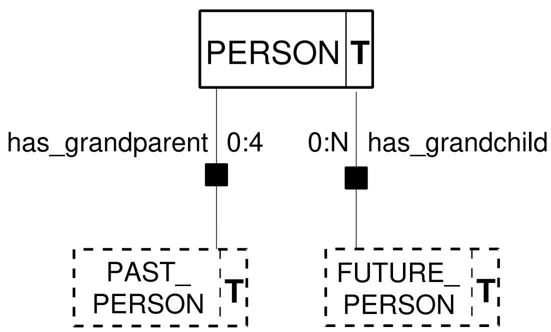


Figure 5 : 'has_grandparent' relationship modelled using derived entities

2.4 Relationships with Temporal Variation and Historical Perspective

Having introduced the two aspects of the temporal nature of relationship involvement, it is a natural question to ask how do the two combine? The effect of saying that a historical perspective has temporal variation is to say that one of entities involved does not have the perspective for its entire existence. In the grandparent example, we might want the say that the grandparent is only related to the grandchild from the time that the grandchild begins to exist. This we shall model using a TH-mark, the semantics of which are given in Definition 9.

Definition 9 If we have instances of entity classes E_1 and E_2 related by R, and if an instance of E_1 is involved with R over period $I_{E1R} \subseteq I_{E1}$, E_2 involved with the same relationship instance for I_{E2}, and relationship R is TH-marked, then we allow $I_{E1R} \neq I_{E2}$.

Although it might be possible a situation where a THT marking could arise, its occurrence would be sufficiently rare not to justify introducing it into the model. Since it would require two time-stamps to be stored, would in any case need two relations in an RDB schema to represent it, and thus is more naturally modelled by a pair of TH relationships.

2.5 The ERT Model

We name the model which includes Definitions 1 to 9 the *ERT* model (Entity-Relationship with Time). To illustrate the description of the model, Example 2 describes the ERT version of the ER model in Example 1, extended with the temporal semantics of the information modelled.

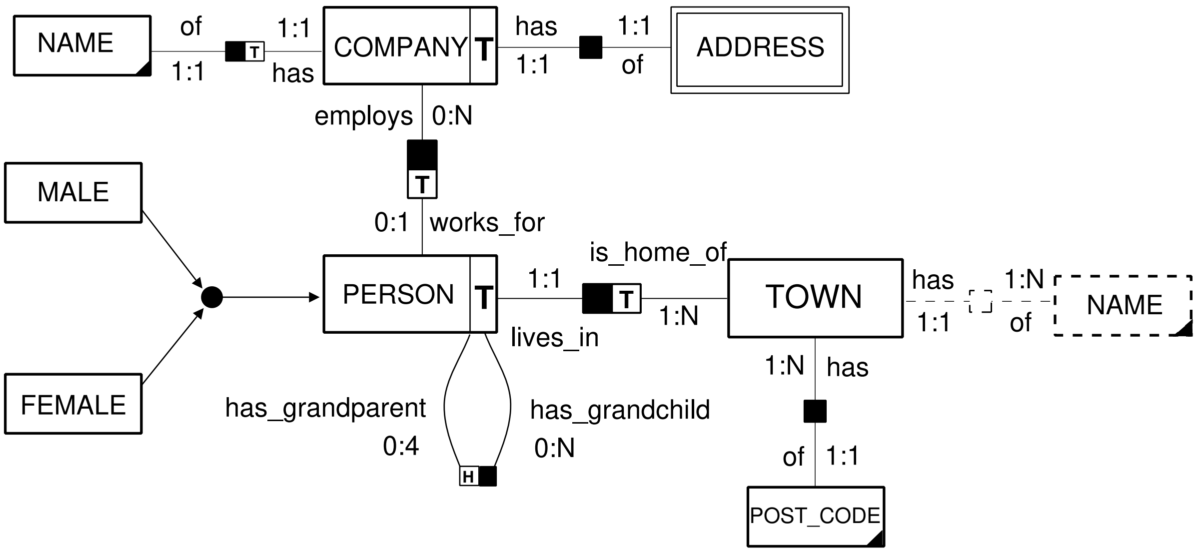


Figure 6 : People working for companies ERT model

Example 2 : ERT Version of the companies ER Model

We consider an instance of *company* to be a non-permanent entity (T on entity class box), which changes its name (T on relationship line), but not its address (unmarked relationship line). Whilst the *company* exists, it may employ a *person*, which is also a non-permanent entity. Neither the *company* nor *person* need be involved for their entire existence in the employs relationship (T on relationship line). The *person* class may be subdivided into *male* and *female*, but will belong to only one for its period of existence (subclasses have no T). Each *person* may be related to another *person* who is their grandparent or grandchild, and who exists at a different time (H on relationship line). We may describe the relationship *has_grandchild* as *current and future*, and the relationship has_grandparent as *past and current*. The *town* entity instances are permanent features of our UoD (no T on entity box), and keep the same *post_code* and *name* throughout their existence.

3 Constraints and Guidelines on the use of the ERT Model

Using Definitions 1-9, we may derive a number of constraint rules and guidelines for the ERT model. We list here only rules which are of significance to the temporal aspects of the model. The rules may be divided into the following categories:

- *Constraint rules* (prefixed **C**) which describe checks that can be made in the run-time database implementing the ERT model. As with any constraint violations, if an update operation is attempted which violates any of the rules, then either the violating operation should be rejected, or the operation should somehow be propagated to the dependent facts so that the integrity of the database is maintained.

- *Meta-schema rules* (prefixed **M**) which can be included in the tool used to construct the ERT model to stop semantically invalid models being constructed.

- *Guidelines* (prefixed **G**) which help the person attempting to model the UoD with an ERT in the choice of which model constructs to use. They are not rigid rules, but state what is unlikely to be a correct model.

3.1 Sub-Class Membership

C1 If the entity class E has a subclass S_E, then $I_{\text{S\_E}} \subseteq I_{\text{E}}$.

If the superclass is not T-marked then this is automatically true since its instances are regarded as having an interval $\aleph$. From **C1** we can deduce the following rule:

C2 If a non T-marked entity class E has a T-marked subclass, then instances of E can not be erased from the database unless they are also erased as members of the T-marked subclass.

If **C2** did not hold, it would be possible for members of the subclass to exist at times when they were not members of the superclass.

C3 At any particular tick, an entity may not be a member of two disjoint subclasses, but may during its life-time change membership from one subclass to another if they are T-marked, provided that the intervals do not overlap. Thus for subclasses S_1_E ... S_n_E of entity class E we may say that for all pairs of instance intervals $I_{\text{S}i} \cap I_{\text{S}j} = \varnothing$ where $\{i,j \in 1 \ldots \text{n}, i \neq j\}$.

M1 It is incorrect to have exactly one of a total set of disjoint subclasses T-marked.

G1 Either all entities of a particular set of disjoint subclasses must be T-marked, or non T-marked.

If **M1** did not hold, then the single T-marked subclass could cause other total subclasses S_n_E to hold for the range $\varnothing \subset I_{\text{Sn}} \subset I_{\text{E}}$, which would contradict their unmarked status. **G1** extends the restriction, in that it is unlikely that we would want to model situations such as S_1_E holding for all ticks of E or S_2_E and S_3_E changing over the time of E.

C4 If the disjoint subclasses S_n_E to S_n_E of entity E are total, then the sum of the intervals over which an instance of subclasses hold must be the interval of the associated superclass instance, i.e. $I_{S1} \cup \ldots \cup I_{Sn} = I_E$.

3.2 Relationships

Relationships which have an involvement cardinality x:y and x≥1 are said to be *total*. This implies that:

C5 For every tick over which the entity exists, an instance of a total relationship R involving that E must exist, i.e. $(\cup I_R) = I_E$.

C6 If the invalidation of a relationship would mean that the involvement cardinality falls below the lower limit (x above), then either a new instance of the relationship has to be inserted or the entity involved should also be invalidated.

G2 The use of total cardinality constraints on T-marked relationships should be avoid. This is due to it being complex, and thus inefficient, to implement.

3.3 Complex Entities

M2 If a complex entity class is T-marked, then its component entity classes should also be T-marked.

As with sub-classes, without this rule a situation could arise in which we have a dependent component valid for an interval greater than that of the complex entity to which it belongs, and thus becomes 'separated' from the rest of the ERT model.

4 Mapping an ERT Model to an RDB Schema

The most common use of ERT models is in the modelling of relational database schemata. Producing an optimal relational schema, and defining all the ways in which various aspects of the ERT are mapped to it would require a detailed paper in its own right. What is presented here is a naive mapping from the ERT to a RDB schema, loosely based on the non-temporal ER to RDB mapping presented in [Ull88]. Apart from supporting the T and H marks, the ERT model does not require entities to have a set of attributes which form a key. We thus need to use surrogates to uniquely identify entities, where attribute `E#` of relation `E` is the surrogate for entity class E.

It was shown in [Gab91,McB92] that the temporal structure, and its associated temporal relational algebra (TRA) query language, can be encoded to a relational database and algebra, with two tick values being added to the relations of the temporal structure. The pairs of ticks in a tuple represent the maximal interval for which the tuple holds. We can produce such a relational schema from our ERT model using the mapping described in Table 2.

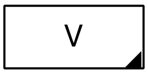
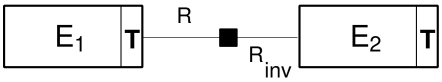
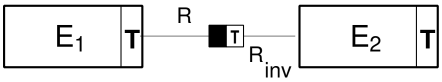
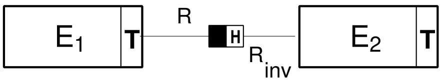
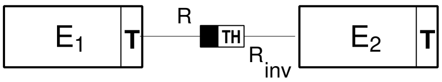
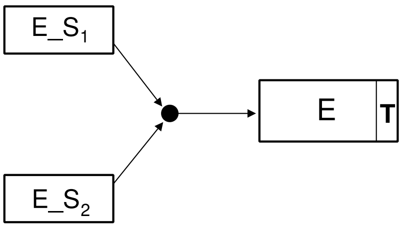
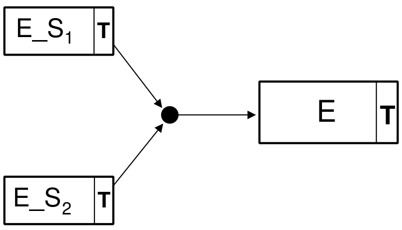
ERT Object	*RDB Schema*
	`E(E#)`
	`E(E#,Start_Time,End_Time)`
	`V(V)`
	`R(E1#,E2#)`
	`R(E1#,E2#,Start_Time,End_Time)`
	`R(E1#,E2#)`
	`R(E1#,E2#,Start_Time,End_Time)`
	`E_S1(E#)` `E_S2(E#)`
	`E_S1(E#,Start_Time,End_Time)` `E_S2(E#,Start_Time,End_Time)`

Table 2 : Mapping of ERT objects to an RDB schema

In Table 2, the mapping for relationships and subclasses has been given for the case where E is T-marked, but the definitions apply equally well if E is unmarked. The overall point to note is that time-stamp information is only recorded where necessary. For example, in the case of a non T-marked relationship between two T-marked entities, we can derive its period of validity from the intersection of the period of validity of the entities, and thus do not need to store the time-stamp.

H-marked relationships do not require a time-stamp; what an H mark tells us is that in the relational schema the two entity surrogates `E1#` and `E2#` involved may have time-stamps at different times, and thus must be related by some temporal operator. In effect `E2#` is a *temporal key* for a relation at some tick in the temporal structure, and does not need to be a key for a relation at the tick time of `E#1`. The TRA may be used for querying a relational schema produced by this mapping, and contains the temporal operators necessary to use the information provided by historical relationships.

Conclusions

The ERT model captures and represents the fundamental characteristics of temporal variation and historical perspective. We have informally related the ERT model with a equivalent relational model called the temporal structure, which models all discrete bounded historical relational database models. Thus we can be satisfied that our model provides a fully expressive tool for the conceptual modelling of historical relational database schemata. We have also sketched the mapping that can be used to produce a relational database schema suitable for use with a temporal relational algebra. Future work will formalize the mapping from ERT to the relational temporal structure, and detail the rules necessary to derive time-stamps for relationships and subclasses where they are not stored directly.

Acknowledgements

The work reported in this paper has been partly funded by the Commission of the European Communities under the ESPRIT R&D programme. The TEMPORA project (number E2469) is a collaborative project between: BIM, Belgium; Hitec, Greece; Imperial College, UK; LPA, UK; SINTEF, Norway; SISU, Sweden; University of Liège, Belgium and UMIST, UK. SISU is sponsored by the National Swedish Board for Technical Development (STU), ERICSSON and Swedish Telecomm.

We would like to thank all the following members of the TEMPORA project for their contributions during the design of the ERT: R. Andersen, P. Bergsten, J-L. Binot, J. Bubenko Jnr, D. Gabbay, E. Katsouli, P. Loucopoulos, M. Niézette, R. Owens, G. Papastamatiou, U. Persson, F. Schumacker, B. Scurlock, A. Sølvberg, U. Sundin, C. Theodoulidis, R. Wohed, P. Wolper and I. de Zegher.

References

[Elm89] *Fundamentals of Database Systems,* R.Elmasri and S.Navathe, Redwood City, California: Benjamin/Cummings, 1989

[Elm90a] *A Temporal Model and Query Language for ER Databases,* R.Elmasri and G.Wuu, IEEE Data Engineering Conference, 1990

[Elm90b] *Semantics of Temporal Data in an Extended ER Model*, R.Elmasri, I.El-Assal and V.Kouramajian, 9th International Conference on Entity-Relationship Approach, Lausanne (CH): 249-264, 1990.

[Fer85] *Modeling the Time Dimension in an ER Diagram*, S.Ferg, 4th Entity-Relationship Conference, 1985.

[Gab91] *Temporal Logic & Historical Databases,* Dov Gabbay & Peter Mc Brien, Proceedings of the 17th International Conference on Very Large Data Bases, Barcelona, 1991.

[Lou90] *TEMPORA - Integrating Database Technology, Rule Based Systems and Temporal Reasoning for Effective Software,* Pericles Loucopoulos, Peter Mc Brien, Ulf Persson, François Schumacker & Phil Vasey, ESPRIT Conference Proceedings, 1990

[McB91] *The Tempora External Rule Language,* Peter Mc Brien, Marc Niézette, Sakis Pantazis, Babis Thedoulidis, Gregoris Tziallas, Anne Helga Selveit, Ulf Sundin & Rolf Wohed, CAiSE-91 Conference Proceedings, May 1991, Springer-Verlag LNCS

[McB92] *The Querying and Updating of a Historical Database held in an RDBMS,* Peter Mc Brien, Dept. of Computing, Imperial College, Internal Report 1992.

[McK91] *Evaluation of Relational Algebra Incorporating the Time Dimension in Databases,* L.Edwin McKenzie Jr. & Richard T.Snodgrass, ACM Computing Surveys, Vol.23 No.4, December 1991.

[Sno85] A Taxonomy of Time in Databases, R.Snodgrass & I.Ahn, Proceedings of ACM SIGMOD International Conference on Management of Data, Ed S.Navathe, ACM Press, 1985.

[The91] *The Entity-Relationship Time Model and the Conceptual Rule Language,* C. Theodoulidis, P. Loucopoulos, B. Wangler & D.Pantazis, 9th Entity-Relationship Conference, 1991

[Tuz90] *A Temporal Relational Algebra as a Basis for Temporal Relational Completeness,* A.Tuzhilin & J.Clifford, Proceedings of the 16th International Conference on Very Large Data Bases, Brisbane, 1990.

[Ull88] *Principles of Database and Knowledge-Base Systems Volume 1,* J.D.Ullman, Computer Science Press, 1988.