# An Entity-Relationship Model Extended To Describe Historical Information

**Peter McBrien**
Imperial College
Dept. of Computing
180 Queen's Gate
London SW7 2BZ
U.K.

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

**Benkt Wangler**
SISU
PO Box 1250
16428 Kista
Sweden

email: pjm@doc.ic.ac.uk

24th July 1991

**Abstract**

The 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 relations may exhibit, and apply the results to enhance a binary ER model with temporal semantics, which may fully model a certain class of historical relational databases.

**Keywords:** Data Models, ER 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.

Research into temporal relational databases 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, the two being

- the *event time*, which records as an interval $E$ = [*start*,*end*] 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 an interval $T$ = [*start*,*end*] the time over which the information holds in the information system (IS).

For example, we may 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 $E$ = [1st January,31st January] and $T$ = [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*. 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 upon in the area of relational historical databases, a selection being [Cli85,Tan87,Sno87,Nav88,Gad88]. These all involve specifying a particular data model for the temporal information stored, but in [Tuz90] a more general *temporal structure* was introduced, which which describes any discrete temporal model. A historical database $\mathbf{D_T}$ is considered as a series of relational databases $\mathbf{D} \in \{ \mathbf{D}_t \mid 0 \leq t \leq n \}$ where the subscript $t$ denotes the time associated with the 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 schemas of $\mathbf{D}_t$. By restricting $t$ to some finite subset of the natural numbers 0 … $n$ we achieve a bounded model, and using the usual ordering on the natural numbers achieves a linear model. As a shorthand, we will term each instance of $t$ as a *tick*.

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, but the instances of each entity and attribute may vary over time, and the involvement in relationships may both vary over and cross time.

To set the results presented in this paper in perspective against previous work in modelling temporal information with the ER model, we will now review some earlier work from which he results have been derived. Ferg [Fer85] proposed that relationships could be timestamped 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 mark was placed on relationships when the timestamp was required, all attributes of entities were represented as (graphically) distinct *value classes*, and the entities themselves were permitted to be timestamped.

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 constrast, 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 philosophy applies for non-temporal and temporal relationships.

By contrast, in the 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. Non time-stamped entities are considered as lasting forever, and so roughly correspond to the conceptual entities of [Elm90b]. Note that time-stamped entities and relationships will stay in the database for as long as you want, it is only their validity in the UoD that ends, and hence they may last forever in the memory of the system.

The purpose of the ERT model described in this paper is allow the graphical representation of the different modes by which entities and the relationships between them may vary over time. The approach gives stricter definitions then in the original ERT model [The91] for the period each timestamp may take when connected to other objects in the model which are also timestamped, which leads to there being three types of relationship. The stricter semantics given to the temporal components of the model leads to a more accurate reflection of the domain being modelled, and so serves to give constraints on the values timestamps may take in the database. These constraints can be used to derive which relations in the RDB must have timestamps, and to act as runtime constraints on what value the timestamps may take.

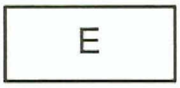
The paper is divided into three sections. Section 1 briefly introduces the binary ER model we used as a basis for the temporal extension. Section 2 describes the manner in which data may vary over time and be related between times, and introduces constructs into the ER model to represent the aspects which are of interest in conceptual modelling. Section 3 describes the constraints and guidelines we place on the use of the temporal constructs during the development of the model, and during the run-time use of a database. The query language for the model has already been described in [McB91].

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

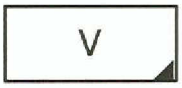
# 1 An Extended ER Model

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 components of our ER Model are listed in the following table:

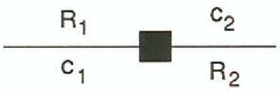
| *ER Model Construct* | *Description* |
|---|---|



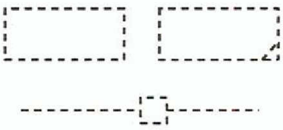

An *entity class* E, which denotes a set of objects (concrete or abstract) in the UoD which share the same type of attributes and relationships with other objects.



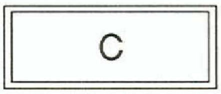

A *value class* V, which is a collection of lexical objects only of interest when associated with some entity.



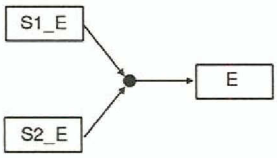

A *relationship* $R_1$, which denotes a set of *associations* or *involvements* between any entity $E_1$, and a value class $V_2$ or entity class $E_2$. The *cardinality* $c_1$=min:max denotes that each instance of $E_1$ must have at least min and at most max involvements in $R_1$. The inverse relationship is described by $R_2$ and $c_2$, and holds for the same entities as $R_1$.



Dashed outlines indicate that the objects are *derived* and not 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.




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.




*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).

## Example: People working for Companies

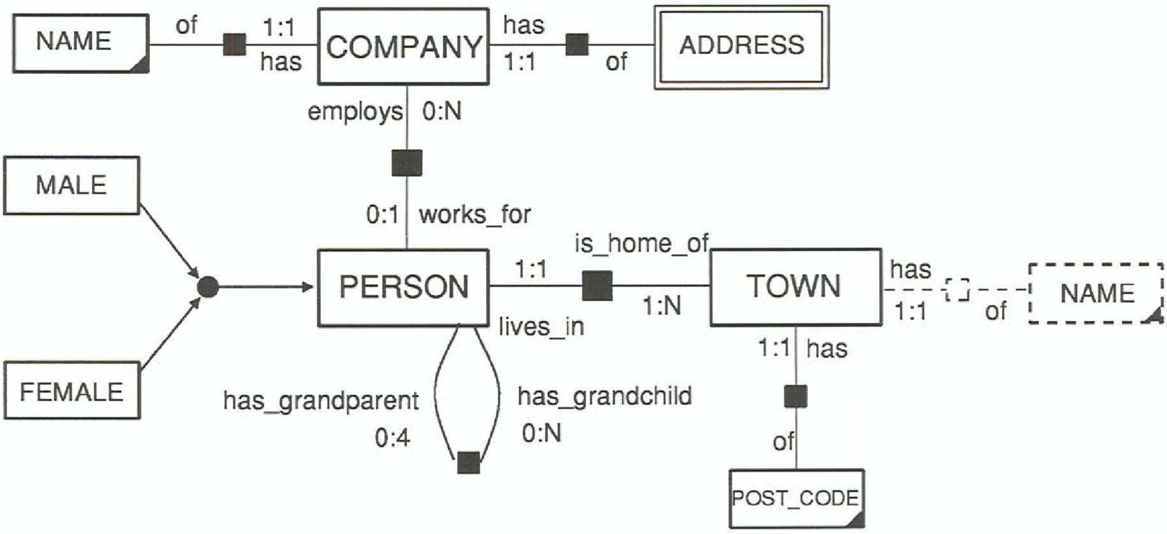



*Figure 1 People working for Companies ER Model*

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 the *persons* divide into two sub-classes of *male* and *female*, and 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 if figure 2.

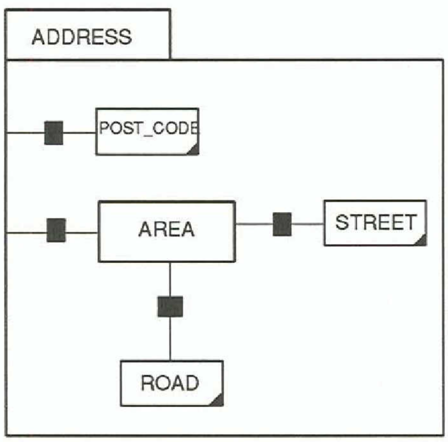



*Figure 2 Details of the 'Address' complex object*

# 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 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 the ER model will be the same at each tick, only the instances of classes and their involvement in relationships will vary over time. 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. 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 associate entities which exist at the same tick. 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, is 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 instances 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 the same ticks and an entity $E_2$ which exists at *other* ticks. 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 time 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. Since the relationship involvement may vary with time, there is no benefit in recording and modelling the temporal behaviour of value instances themselves since the only way they may be accessed (by relationships) is itself 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 $I_x = \{t_a , t_b ... t_z\}$ where $t_a , t_b ... t_z$ are the ticks at which it holds, and since the series of ticks concerned is usually continuous we 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 given the shorthand notation of $[t_a , t_z]$ for the case where the series if ticks is continuous. Our definition of an interval allow for the use of the usual set operators for finding interval union, inclusion, *etc*. In our discrete bounded structure the possible ticks of a interval are limited to a finite set $\aleph = \{0 ... n\}$, and so all $I_x \subseteq \aleph$.

In the following definitions, when we speak of the interval $I_E$ of entity class E, we are referring to the interval 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.

## 2.1 Entity Temporal Variation

*Entities which are not sub-classes*

The *temporal variation* in entity existence describes the property of an entity only existing at certain times in our UoD. We may define the semantics of our T-mark in definitions 1 and 2.

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

**Definition 2** An entity class E which is not a specialization of another entity class, and is not marked by T indicates that $I_E = \aleph$. (All value classes V are not marked, so we always have $I_V = \aleph$).

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 about the UoD. For instance, for the people working for companies ER model (figure 1), we might have *company* and *person* entities changing over time, and there being 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). 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.

Figure 3 shows the instances of entities and their relationships of figure 1 over the period [1,3]. Since we allow temporal variation of the *company* entity class, it may have instances $\{a_1,a_2,a_3\}$ at ticks 1 and 2, and just instance $\{a_3\}$ at tick 3, which we model 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 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 a specialization of another entity class E, and is not marked by T indicates that $I_{S\_E} = I_E$.

**Definition 4** An entity class S_E which is a specialization of another entity class E, and is marked by T 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 *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. An example of this may be an *order* entity, which during its life-time may move through the states of being a *new_order*, a *backorder* 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. Also we may change a *person* from *male* to *female* subclasses, but the recorded information will only reflect that *person* as belonging to just one subclass for its entire existence.

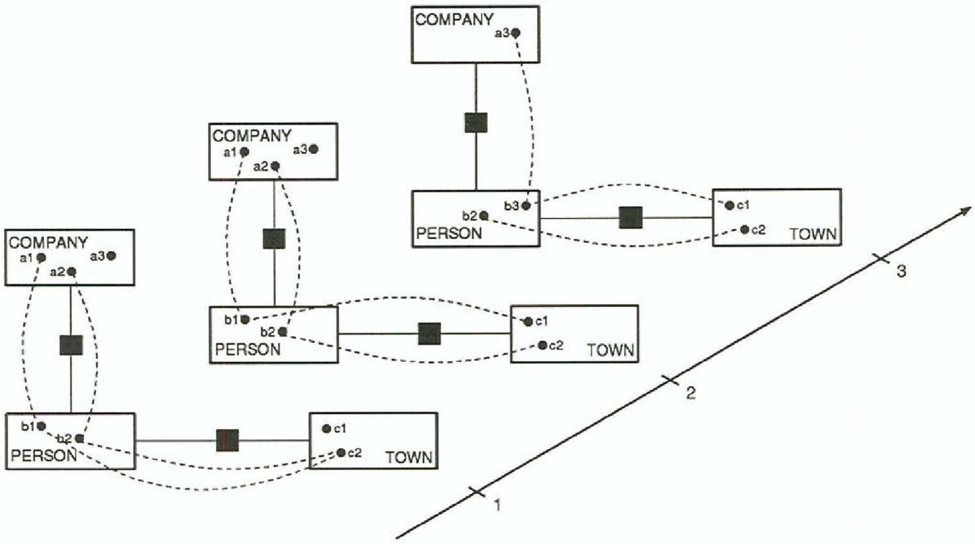



*Figure 3 Temporal variation of entity and relationship involvement*

## 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 $I_R \subseteq I_{E1}$ and $I_R \subseteq I_{E2}$.

**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 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*. It is for this reason definitions 5 and 6 restrict the interval of a relationship to be a subset of the union of the entities it involves.

An example of temporal relationship variation 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 associated from *c2* to *c1*. We represent this in the ERT model of figure 6 by placing a T-mark on the relationship line.

## 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 $I_{ER1}$, $E_2$ involved for $I_{ER2}$, and relationship R is H-marked then we allow $I_{ER1} \neq I_{ER2}$.

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).

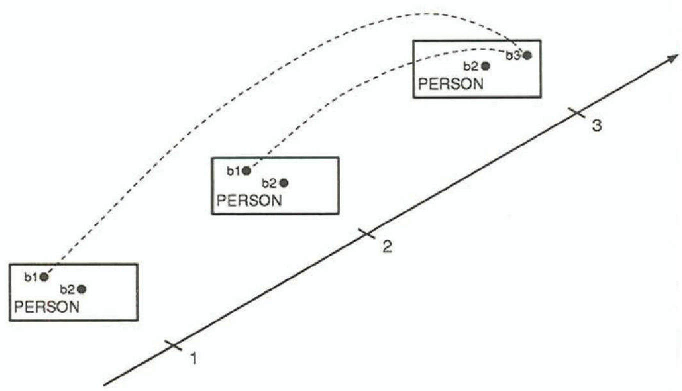



*Figure 4 Historical Perspective in 'has_grandparent' Relationship*

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 and after the time when the grandparent existed.

**Definition 8** An H relationship of $E_1$ R $E_2$ is described as:

- *past* if $E_2$ exists before $E_1$ exists.
- *current* if $E_2$ exists while $E_1$ exists.
- *future* if $E_2$ exists after $E_1$ existed.
- 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 *past_person* at a particular tick is 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 fact we have two quite distinct unmarked relationships and entity classes to model the single H-marked relationship.

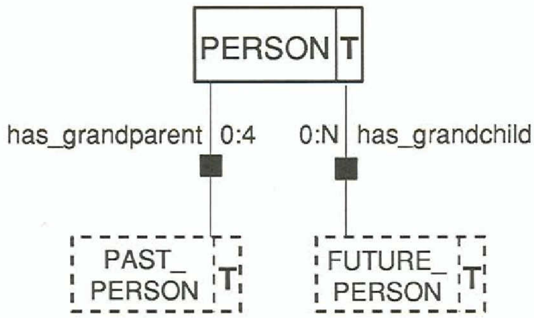



*Figure 5 'has_grandparent' Relationship Modelled Using Derived Entities*

## 2.4 The ERT Model

We name the model which includes definitions 1 to 6 the *ERT* model (Entity-Relationship with Time). To conclude the description of the model, we present the companies ER model of figure 1 extended with the temporal semantics of the information modelled to give the ERT model in figure 6. The temporal description provided by the ERT model is as follows:

> 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.

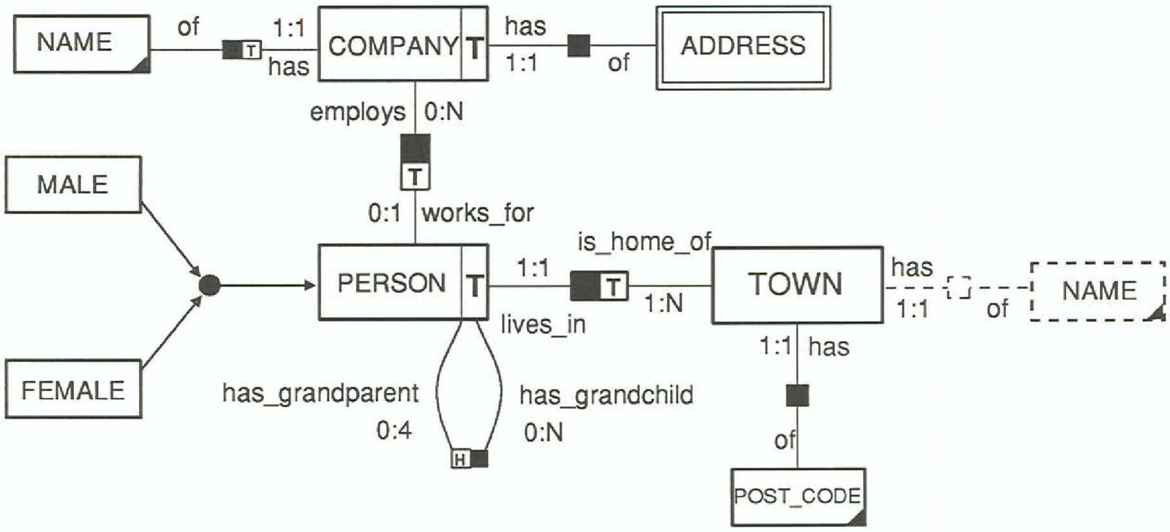



*Figure 6: People working for Companies ERT Model*

# 3 Constraints and Guidelines on the use of the ERT

Using definitions 1-8, 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* (numbered **R**) 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 data base is maintained.

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

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

## 3.1 Sub-Class Membership

**R1** If the entity class E has a subclass S_E, then $I_{S\_E} \subseteq I_E$.

If the superclass is not T-marked then this is automatically true since its instances are regarded as having a interval of $\aleph$. From **R1** we can deduce the following rule:

**R2** If a non T-marked entity class has a T-marked subclass, then its instances can not be erased from the database unless they are also erased as members of the T-marked subclass.

If **R2** did not hold, it would be possible for members of the subclass to exist at times when they were not members of the superclass.

**R3** 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, provided 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 $I_{Si} \cap I_{Sj} = \varnothing$ where $\{i,j \in 1 \ldots n\}$.

**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 none 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 $\varnothing \subset I_{Sn} \subset I_E$, 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.

**R4** If the disjoint subclasses $S_1$_E to $S_n$_E of entity E are total, then the sum of the intervals over which the instances of subclasses hold must be the interval of the associated superclass instances, 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:

**R5** For every tick over which the entity exists, an instance of a total relationship R involving that E must exist, i.e. $(\cup I_R) = I_E$.

**R6** 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.

### 3.3 Complex Entities

**M2** If a complex entity class is T-marked, then also 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.

## 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 which models all discrete bounded historical relational database models, and thus can be satisfied that our model provides a fully expressive tool for the conceptual modelling of historical relational database schemas. Future work will formalize this connection, and in addition explore two new areas. Firstly, what semantics and use would we have if we allowed relationships to be both T and H marked, and secondly what rules can be provided for the mapping of an ERT schema to a historical database schema.

## 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, A. Conti, G. Diakonikolaou, D. Gabbay, E. Katsouli, P. Loucopoulos, R. Owens, G. Papastamatiou, U. Persson, S. Pantazis, F. Schumacker, B. Scurlock, A. Sølvberg, U. Sundin, C. Theodoulidis, G. Tziallas, G. Uddeberg, P. Vasey, R. Wohed, P. Wolper and I. de Zegher.

## References

[Cli85] *On An Algebra for Historical Relational Databases: Two Views*
J.Clifford & A.U.Tansel
Proc. ACM SIGMOD Conference 1985.

[Cli87] *The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans,*
J.Clifford and A.Croker,
IEEE Data Engineering Conference, 1987.

[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.

[Gad88] *A Homogeneous Relational Model and Query Languages for Temporal Databases*
S.K.Gadia
ACM TODS, Vol 13, No. 4 pp 418-448, 1988.

[Lou90] *TEMPORA - Integrating Database Technology, Rule Based Systems and Temporal Reasoning for Effective Software*
Pericles Loucopoulos, Peter M<u>c</u> 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

[Nav88] *TSQL - A Language Interface for History Databases*
S.B.Navathe & R.Ahmed
Temporal Aspects of Information Systems, pp 109-122

*Ed.* C.Rolland, F.Bodart & M.Leonard
North-Holland, 1988.

[Nav90] *A Temporal Data Model and Query Language,*
S.Navathe and R.Ahmed,
Information Sciences, 1990.

[Sno87] *The Temporal Query Language TQuel*
Richard Snodgrass
ACM TODS Vol 12, No. 2 pp 247-298, 1987.

[Tan87] *Adding Time Dimension to Relational Model and Extending Relational Algebra*
A.U.Tansel
Information Systems Vol 11, No. 4, pp 343-355, 1986.

[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 VLDB Conference 1990, Brisbane.