Temporal ER Modelling with Description Logics: A Preliminary Report

Alessandro Artale
Department of Computation
UMIST, Manchester, UK
artale@co.umist.ac.uk

Enrico Franconi
Department of Computer Science
University of Manchester, UK
franconi@cs.man.ac.uk
http://www.cs.man.ac.uk/~franconi/

ABSTRACT

Recent efforts in the Conceptual Modelling community have been devoted to properly capturing time-varying information. Various temporally enhanced Entity-Relationship (ER) models have been proposed that are intended to model the temporal aspects of database conceptual schemas. This work gives a logical formalisation of the various properties that characterise and extend different temporal ER models which are found in literature. The formalisation we propose is based on Description Logics (DL), which have been proved useful for a logical reconstruction of the most popular conceptual data modelling formalisms. The proposed DL has the ability to express both enhanced temporal ER schemas and integrity constraints in the form of complex inclusion dependencies.

1 Introduction

Temporal Databases are databases that store historical information, i.e., essentially past and present data. Temporally enhanced Entity-Relationship (ER) models have been developed to conceptualise the temporal aspects of database schemas, namely *valid time* – when a fact holds, i.e., it is true in the representation of the world – and *transaction time* – which records the history of database states rather than the world history [Jensen *et al.*, 1994]. In this paper we consider a logical formalisation of the validity time [Chomicki and Toman, 1998] which is supported by almost all the temporal ER models – differently from the transaction time.

In the temporal ER community two different main modelling approaches have been devised to provide temporal support. The *implicit* approach hides the temporal dimension in the interpretation structure of the ER constructs. Thus, a temporal ER model does not include any new specific temporal construct with respect to a standard ER model. Each ER construct is always interpreted with a temporal semantics, so that instances of temporal entities or relationships are always potentially time-varying objects. The *explicit* approach, on the other hand, retains the non-temporal semantics for the conventional ER constructs, while adding new syntactical constructs for representing temporal entities and relationships and their temporal interdependencies. The advantage of the explicit approach is the so called *upward compatibility*: the meanings of conventional (legacy) ER diagrams when used inside a temporal model remains unchanged. This is crucial, for example, in modelling data warehouses or federated databases, where sources may be a collection of both temporal and legacy databases.

A logical formalisation is introduced in this paper that can cover both the implicit and the explicit approaches. The idea is to provide a formalisation for implicit temporal ER models, enriched with the ability to express a powerful class of temporal integrity constraints. While instances of ER entities or relationships are potentially time-varying objects, integrity constraints can impose restrictions in the temporal validity of such objects. The formalisation is powerful enough that it is possible to explicitly state as integrity constraints the distinction between time-varying and snapshot (i.e., time invariant) constructs. In this way, an ER diagram may contain both temporal and non-temporal information, providing the ability to capture the explicit approach.

The formalisation presented in this paper is based on an expressive temporal Description Logic. Advantages of using Description Logics are their high expressivity combined with desirable computational properties. The core is the Description Logic $\mathcal{ALCQI}$, which is able to capture conventional ER models and has the ability to express a powerful class of inclusion dependency constraints. The language is then extended with a tense logic. Temporal integrity constraints can be expressed in this combined logic, called $\mathcal{ALCQIT}$.

The paper is organised as follows. Section 2 introduces the temporally enhanced ER models, in both the implicit and explicit approaches. Section 3 introduces the temporal $\mathcal{ALCQIT}$ description logic. Section 4 will show how temporal Entity-Relationship schemas can be encoded into the temporal description logic, how additional temporal integrity constraints can be imposed on schemas, and how it is possible to reason in this framework. The final Section describes how integrity constraints can encode explicit time-varying and snapshot constructs.

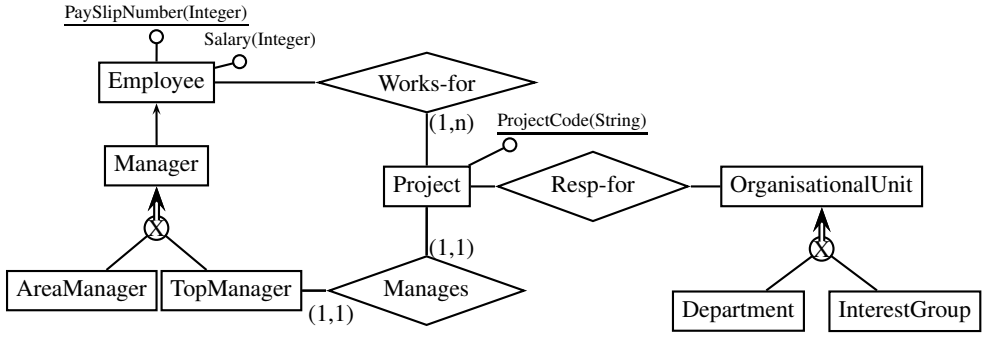

Figure 1: The reference temporal ER diagram

2 The Temporal ER Model

In this Section we introduce the temporally enhanced ER model and its semantics. We first consider standard ER diagrams.

Basic elements of ER schemas are *entities*, denoting a set of objects called *instances*, and *relationships*, denoting a set of *tuples* made by the instances of the different entities involved in the relationship. Since the same entity can be involved in the same relationship more than once, participation of entities in relationships is represented by means of *ER-roles*, to which a unique name is assigned. ER-roles can have *cardinality constraints* to limit the number of instances of an entity involved in the relationship. Both entities and relationships can have *attributes*, i.e., properties whose value belong to some predefined domain – e.g., `Integer`, `String`. Additionally, the considered ER model includes taxonomic relationships to state inclusion assertions between entities and between relationships. ER schemas are usually built using graphical tools, as showed in Figure 1.

So far we have considered ER diagrams where no explicit temporal constructs appear. According to the implicit approach, a temporally enhanced ER diagram does not have any specific temporal construct, since it is intended that every construct has always a temporal interpretation. Thus, the syntax of the temporal model is the same as the standard one, and the temporal dimension is considered only at the semantical level.

The first-order semantics for the temporally enhanced ER model – in the implicit approach – is the standard and intuitive choice. We consider as a starting point the non temporal semantics introduced in [Calvanese *et al.*, 1998]. Interpretations of a temporal ER diagram $\mathcal{D}$ are those temporally dependent *database states* $\mathcal{B}(t) = (\Delta^{\mathcal{B}}, \cdot^{\mathcal{B}(t)})$ such that: each domain symbol D is mapped to a subset $D^{\mathcal{B}(t)}$ of the corresponding domain interpretation at time point t; each entity symbol E is mapped to $E^{\mathcal{B}(t)} \subseteq \Delta^{\mathcal{B}}$ at time point t; each attribute symbol A is mapped to $A^{\mathcal{B}(t)} : \Delta^{\mathcal{B}} \rightarrow \bigcup_i D_i^{\mathcal{B}(t)}$ at time point t; each n-ary relationship, with both ER-roles and attributes, denotes a set of labelled n-tuples over $(\Delta^{\mathcal{B}} \bigcup_i D_i^{\mathcal{B}(t)})^n$ at time point t. A *legal database state* should satisfy the following conditions: If $E_1 \text{ISA} E_2$, then $E_1^{\mathcal{B}(t)} \subseteq E_2^{\mathcal{B}(t)}$ for every t; if A is an attribute of entity E with domain D, then $\forall e \in E^{\mathcal{B}(t)}.A^{\mathcal{B}(t)}(e) \in D^{\mathcal{B}(t)}$ for every t; for each relationship R with $E_1, \ldots, E_k$ entities involved by means of ER-roles $U_1, \ldots, U_k$, and attributes $A_1, \ldots, A_l$ with domains $D_1, \ldots, D_l$, then every instance tuple of R at time t has the form $[U_1 : e_1, \ldots, U_k : e_k, A_1 : a_1, \ldots, A_l : a_l]$, where $e_i \in E_i^{\mathcal{B}(t)}$ and $a_j \in D_j^{\mathcal{B}(t)}$; for each minimum and maximum cardinality constraint, n, m respectively, in a ER-role U relating a relationship R with and entity E, then $n \leq \#\{r \in R^{\mathcal{B}(t)} \mid r[U] \in E^{\mathcal{B}(t)}\} \leq m$ for every t, where $r[U]$ is the projection of R over the ER-role U.

We consider now an enhancement of the temporal ER model by means of integrity constraints. In this section just an informal introduction will be given; the exact characterisation of integrity constraints over a temporal ER diagram will be given in Section 4. The following integrity constraints may be imposed over the example ER diagram of Figure 1:

- managers are the only employees who do not work for a project (she/he just manages it);
- a manager becomes qualified after a period when she/he was just an employee[1].

These constraints can not be expressed directly by means of ER constructs; the logical formalisation presented in the following sections has been explicitly devised for expressing a large class of (temporal) constraints. The presence of the above constraints limits the number of legal database states, since not all the unconstrained databases conform to the newly introduced constraints. It is also easy to see that the legal database states conforming to the enriched schema, which includes both the ER diagram and the integrity constraints, also conform to the following constraints:

- for every project, there is at least an employee who is not a manager,
- each top manager worked in a project before managing some (possibly different) project

Please note that these deductions are not trivial, since from the ER schema the cardinality constraints do not impose that employees necessarily work in a project.

[1] Let us assume that an entity called *Qualified* appears somewhere else in the ER diagram.

$$
\begin{array}{rcll}
C, D & \rightarrow & A \mid & \\
 & & \top \mid & \Delta^{\mathcal{I}} \\
 & & \bot \mid & \emptyset \\
 & & \neg C \mid & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}(t)} \\
 & & C \sqcap D \mid & C^{\mathcal{I}(t)} \cap D^{\mathcal{I}(t)} \\
 & & C \sqcup D \mid & C^{\mathcal{I}(t)} \cup D^{\mathcal{I}(t)} \\
 & & \forall R.C \mid & \{i \in \Delta^{\mathcal{I}} \mid \forall j.\, R^{\mathcal{I}(t)}(i,j) \Rightarrow C^{\mathcal{I}(t)}(j)\} \\
 & & \exists R.C \mid & \{i \in \Delta^{\mathcal{I}} \mid \exists j.\, R^{\mathcal{I}(t)}(i,j) \wedge C^{\mathcal{I}(t)}(j)\} \\
 & & f\!\uparrow\! \mid & \Delta^{\mathcal{I}} \setminus \mathrm{dom}\, f^{\mathcal{I}(t)} \\
 & & f : C \mid & \{i \in \mathrm{dom}\, f^{\mathcal{I}(t)} \mid C^{\mathcal{I}(t)}(f^{\mathcal{I}(t)}(i))\} \\
 & & \geq n\, R.C \mid & \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}(t)}(i,j) \wedge C^{\mathcal{I}(t)}(j)\} \geq n\} \\
 & & \leq n\, R.C \mid & \{i \in \Delta^{\mathcal{I}} \mid \sharp\{j \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}(t)}(i,j) \wedge C^{\mathcal{I}(t)}(j)\} \leq n\} \\
 & & C \mathcal{U} D \mid & \{i \in \Delta^{\mathcal{I}} \mid \exists v.\, v > t \wedge D^{\mathcal{I}(v)}(i) \wedge \forall w.\, (t < w < v) \rightarrow C^{\mathcal{I}(w)}(i)\} \\
 & & C \mathcal{S} D \mid & \{i \in \Delta^{\mathcal{I}} \mid \exists v.\, v < t \wedge D^{\mathcal{I}(v)}(i) \wedge \forall w.\, (v < w < t) \rightarrow C^{\mathcal{I}(w)}(i)\} \\
 & & \Diamond^{+} C \mid & \{i \in \Delta^{\mathcal{I}} \mid \exists v.\, v > t \wedge C^{\mathcal{I}(v)}(i)\} \\
 & & \Diamond^{-} C \mid & \{i \in \Delta^{\mathcal{I}} \mid \exists v.\, v < t \wedge C^{\mathcal{I}(v)}(i)\} \\
 & & \Box^{+} C \mid & \{i \in \Delta^{\mathcal{I}} \mid \forall v.\, v > t \rightarrow C^{\mathcal{I}(v)}(i)\} \\
 & & \Box^{-} C & \{i \in \Delta^{\mathcal{I}} \mid \forall v.\, v < t \rightarrow C^{\mathcal{I}(v)}(i)\} \\
 & & & \\
R, S & \rightarrow & P \mid & \\
 & & f \mid & \\
 & & R^{-1} \mid & \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}(t)}(j,i)\}
\end{array}
$$

Figure 2: $\mathcal{ALCQIT}$ and its semantics.

Explicit Approach

In the *explicit* approach new constructs are added to represent the temporal dimension of the model. At the cost of adding new constructs, this approach has the advantage of preserving the atemporal meaning of conventional (legacy) ER schemas when embedded into temporal ER diagrams: this property is called *upward compatibility*. This crucial property is not realisable within the standard implicit temporal approach. Indeed, if the implicit approach has the attractive of leaving unchanged the original ER model, "this approach rules out the possibility of designing non-temporal databases or databases where some part of a database is non-temporal and the rest is temporal" [Gregersen and Jensen, 1999]. In particular, we suppose that both entities and relationships in a explicit temporal ER model can be either *snapshot* (i.e., each of their instances has a global lifetime, as in the case they derive from a legacy diagram), or *temporary* (i.e., each of their instances has a temporary lifetime). Moreover, we also consider the ability to specify *dynamic entities*, i.e., entities whose instances migrate over time [Gupta and Hall, 1992; Spaccapietra *et al.*, 1998].

3 The Temporal Description Logic

We introduce very briefly in this section the $\mathcal{ALCQIT}$ temporal DL (for a survey see [Artale and Franconi, 2000; 2001; Wolter *et al.*, 2001]), which is obtained by combining a standard point-based tense logic [Gabbay *et al.*, 1994] and the standard non-temporal $\mathcal{ALCQI}$ DL [Calvanese *et al.*, 1999]. For an approach based on temporal intervals see [Artale and Franconi, 1998]; for more expressive temporal description logics see [Wolter and Zakharyaschev, 1998; 1999].

The basic types of the DL are *concepts*, *roles*, and *features*. According to the syntax rules at the left of Figure 2, $\mathcal{ALCQI}$ *concepts* (denoted by the letters C and D) are built out of *primitive concepts* (denoted by the letter A), *roles* (denoted by the letter R, S), and *primitive features* (denoted by the letter f); roles are built out of *primitive roles* (denoted by the letter P) and *primitive features*.

We define the *meaning* of concepts as sets of individuals and the meaning of roles as sets of pairs of individuals. A temporal structure $\mathcal{T} = (\mathcal{P}, <)$ is assumed, where $\mathcal{P}$ is a set of time points and $<$ is a strict linear order on $\mathcal{P}$. Formally, an $\mathcal{ALCQIT}$ *temporal interpretation* over $\mathcal{T}$ is a triple $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(t)} \rangle$, consisting of a set $\Delta^{\mathcal{I}}$ of individuals (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}(t)}$ (the *interpretation function* of $\mathcal{I}$) mapping, for each $t \in \mathcal{P}$, every concept to a subset of $\Delta^{\mathcal{I}}$, every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every feature to a partial function from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}}$, such that the equations at the right of Figure 2 are satisfied.

A *knowledge base* is a finite set Σ of *terminological axioms* of the form $C \sqsubseteq D$. An interpretation $\mathcal{I}$ over a temporal structure $\mathcal{T} = (\mathcal{P}, <)$ satisfies a terminological axiom $C \sqsubseteq D$ if $C^{\mathcal{I}(t)} \subseteq D^{\mathcal{I}(t)}$ for every $t \in \mathcal{P}$. A knowledge base Σ is *satisfiable* in the temporal structure $\mathcal{T}$ if there is a temporal interpretation $\mathcal{I}$ over $\mathcal{T}$ which satisfies every axiom in Σ; in this case $\mathcal{I}$ is called a *model* over $\mathcal{T}$ of Σ. Checking for KB

satisfiability is deciding whether there is at least one model for the knowledge base. Σ *logically implies* an axiom $C \sqsubseteq D$ in the temporal structure $\mathcal{T}$ (written $\Sigma \models C \sqsubseteq D$) if $C \sqsubseteq D$ is satisfied by every model over $\mathcal{T}$ of Σ. In this latter case, the concept C is said to be *subsumed* by the concept D in the knowledge base Σ and the temporal structure $\mathcal{T}$. Concept subsumption can be reduced to concept satisfiability since C is subsumed by D in Σ if and only if $(C \sqcap \neg D)$ is unsatisfiable in Σ.
Reasoning in $\mathcal{ALCQIT}$ (i.e., deciding knowledge base satisfiability and logical implication) is an EXPTIME-complete problem [Schild, 1993; Wolter *et al.*, 2001].

4 Encoding the Implicit Temporal ER Model

Now, it is shown how an ER schema with implicit representation of time can be expressed in a $\mathcal{ALCQIT}$ knowledge base whose models correspond with legal database states of the ER schema – allowing for reasoning services such as satisfiability of a schema or the computation of logically implied integrity constraints. It is important to emphasise the fact that in this approach the integrity constraints are part of the schema, and that reasoning is carried on by taking in complete account all the information contained in the schema.
Let us first consider the translation between an ER diagram (without considering the integrity constraints) and an $\mathcal{ALCQIT}$ knowledge base as follows.

Definition 4.1 (Translation) *An ER diagram $\mathcal{D}$ is translated into a corresponding knowledge base Σ where each domain, entity or relationship symbol corresponds to an atomic concept, and each attribute or ER-role symbol corresponds to an atomic feature, in such a way that the following axioms hold:*

- *for each pair of distinct domains D_i, D_j in $\mathcal{D}$, add to Σ the terminological axiom:*
  $D_i \sqsubseteq \neg D_j$

- *for each pair of distinct relations R_i, R_j in $\mathcal{D}$, add to Σ the terminological axiom:*
  $R_i \sqsubseteq \neg R_j$

- *For each* ISA *link between two entities E, F in $\mathcal{D}$, add to Σ the terminological axiom:*
  $E \sqsubseteq F$

- *For each attribute A in $\mathcal{D}$ with domain D of an entity E (resp. of a relationship R), add to Σ the terminological axiom:*
  $E \sqsubseteq A : D$ *(resp.* $R \sqsubseteq A : D$*)*

- *For each relationship R in $\mathcal{D}$ relating n entities $E_1 \ldots E_n$ by means of the ER-roles*[2] *$P^R_{E_1} \ldots P^R_{E_n}$, add to Σ the terminological axiom:*
  $R \sqsubseteq (P^R_{E_1} : E_1) \sqcap \ldots \sqcap (P^R_{E_n} : E_n)$

[2]We assume that a unique name is given within a relationship to each ER-role, representing a specific participation of an entity in the relationship.

- *For each minimum cardinality constraint $n \neq 0$ in a ER-role P^R_E in $\mathcal{D}$ relating a relationship R with and entity E, add to Σ the terminological axiom:*
  $E \sqsubseteq \geq n\,(P^R_E)^{-1}.\,R$

- *For each maximum cardinality constraint $n \neq \infty$ in a ER-role P^R_E in $\mathcal{D}$ relating a relationship R with and entity E, add to Σ the terminological axiom:*
  $E \sqsubseteq \leq n\,(P^R_E)^{-1}.\,R$

- $\top \sqsubseteq \Box^+ \top \sqcap \Box^- \top$ □

From the definition 4.1 it is clear that n-ary relationships are *reified* in the translated description logic knowledge base, i.e., they become concepts with n special feature names – the ER-roles – denoting the n arguments of the n-ary relationship. Moreover, note that $\mathcal{ALCQIT}$ temporal operators do not appear in the translation, since we are in the implicit approach.
Temporal integrity constraints are expressed by means of additional terminological axioms in Σ. Recall that a terminological axiom states an inclusion between concepts. Thus, an integrity constraint is any kind of inclusion dependency expressible in the full temporal DL $\mathcal{ALCQIT}$.

Definition 4.2 (Inclusion Dependencies) *An integrity constraint for an ER diagram $\mathcal{D}$ is any inclusion dependency which can be expressed in the corresponding knowledge base Σ by means of a terminological axiom of the kind $C \sqsubseteq D$, where atomic concepts appearing in C, D correspond to domain, entity or relationship symbols in $\mathcal{D}$ and atomic features appearing in C, D correspond to ER-role symbols in $\mathcal{D}$.*

Based on the results of [Calvanese *et al.*, 1994; 1998], we have proved that the translation is correct, in the sense that there is a precise correspondence between legal database states of $\mathcal{D}$ and models of the derived knowledge base Σ [Artale *et al.*, 2001]. The existence of this correspondence is such that, whenever the problem of checking an ER schema against a property has a specific solution, then the corresponding reasoning problem in the DL has a corresponding solution, and vice-versa. Thus, it is possible to exploit standard reasoning procedures in the DL for checking properties of the ER schema. The reasoning problems we are mostly interested in are *consistency* of a ER schema – which is mapped to a satisfiability problem in the corresponding DL knowledge base – and *logical implication* within a ER schema – which is mapped to a logical implication problem in the corresponding DL knowledge base.
As a final remark, it should be noted that the high expressivity of DL constructs can capture an extended version of the basic ER model, which includes not only taxonomic relationships, but also arbitrary boolean constructs to represent so called generalized hierarchies with disjoint unions; entity definitions by means of either necessary or sufficient conditions or both, and integrity constraints expressed by means of generalised axioms [Calvanese *et al.*, 1998].

Example

Let us consider the example introduced in Section 2. We first translate the fragment of the ER diagram (Figure 1) involving the entities *Project, Employee, Manager* and the relationship *Works-for* in the description logic knowledge base Σ_{ER}:

$\texttt{WORKS-FOR} \sqsubseteq \texttt{has-prj} : \texttt{Project} \sqcap \texttt{has-emp} : \texttt{Employee}$

$\texttt{Project} \sqsubseteq \exists \texttt{has-prj}^{-1}.\texttt{WORKS-FOR}$

$\texttt{Manager} \sqsubseteq \texttt{Employee}$

We then encode the integrity constraints, which are expressed by means of terminological axioms in a knowledge base Σ_{IC}:

- Managers are the only employees who do not work for a project:

  $\texttt{Manager} \sqsubseteq \forall \texttt{has-emp}^{-1}.\neg\texttt{WORKS-FOR}$

  $\forall \texttt{has-emp}^{-1}.\neg\texttt{WORKS-FOR} \sqsubseteq \texttt{Manager}$

  The constraints state that managers are exactly employees which are not involved in the *Works-for* relation.

- A manager becomes qualified after a period when she/he was just an employee:

  $\texttt{Manager} \sqsubseteq \texttt{Qualified}\, \mathcal{S}\, (\texttt{Employee} \sqcap \neg\texttt{Manager})$

  The constraint states that all managers are qualified after they have been at the same time employees and not managers.

It turns out that the following integrity constraints are logically implied from $\Sigma_{ER} \cup \Sigma_{IC}$:

- For every project, there is at least an employee who is not a manager:

  $\Sigma_{ER} \cup \Sigma_{IC} \models$
  $\texttt{Project} \sqsubseteq \exists \texttt{has-prj}^{-1}.\exists \texttt{has-emp}.\neg\texttt{Manager}$

  The constraint states that every project is such that there exists somebody working for it who is not a manager.

- A top manager worked in a project before managing some (possibly different) project:

  $\Sigma_{ER} \cup \Sigma_{IC} \models$
  $\texttt{TopManager} \sqsubseteq \Diamond^{-}\exists \texttt{has-emp}^{-1}.\exists \texttt{has-prj}.\texttt{Project}$

  The constraint states that every manager should have worked some time in the past for a project.

Moreover, if we change in Σ_{ER} the minimum cardinality of the participation of employees to the *Works-for* relationship to one (i.e., we make it a mandatory participation):

$\texttt{Employee} \sqsubseteq \exists \texttt{has-emp}^{-1}.\texttt{WORKS-FOR}$

then, even if Σ_{ER} is satisfiable, $\Sigma_{ER} \cup \Sigma_{IC}$ is an unsatisfiable knowledge base, because of the first integrity constraint. For the abovementioned theorem, no legal DB state exists for the ER schema including the constraints.

5 Encoding the Explicit Temporal ER Model

In the following it will be showed how the proposed formalisation can encode explicit temporal ER models by simply imposing specific constraints defining snapshot and temporary constructs, thus maintaining the required upward compatibility. Furthermore, the formalisation within $\mathcal{ALCQIT}$ is in agreement with the desirable *orthogonality* principle [Spaccapietra *et al.*, 1998], since temporal constructs can be specified separately and independently for entities, relationships and attributes. Depending on the application requirements, the temporal support is decided by the designer.

5.1 Snapshot and Temporary Entities and Relationships

By looking at the semantics of the implicit model, it is clear that both entity and relationship instances have an existence time associated to them. $\mathcal{ALCQIT}$ formulas can express *timestamps* by enforcing either that entities (relationships) cannot last forever – we call them *temporary entities* (*temporary relationships*), or that their extension never changes across time – we call them *snapshot entities* (*snapshot relationships*). These formulas would capture the meaning of the various time-stamping marks proposed in temporal conceptual modelling systems. Temporary entities and relationships are captured by the following constraints:

$$\begin{array}{lll} E & \sqsubseteq & (\Diamond^{+}\neg E) \sqcup (\Diamond^{-}\neg E) \\ R & \sqsubseteq & (\Diamond^{+}\neg R) \sqcup (\Diamond^{-}\neg R) \end{array}$$

asserting that there must be a past or future time point where the entity (relationship) does not hold. Indeed, instances of temporary entities (relationships) have necessarily a limited lifetime.
On the other hand, snapshot entities and relationships are captured by the following constraints:

$$\begin{array}{lll} E & \sqsubseteq & (\Box^{+}E) \sqcap (\Box^{-}E) \\ R & \sqsubseteq & (\Box^{+}R) \sqcap (\Box^{-}R) \\ R & \sqsubseteq & (\star P^{R}_{E_1} : E_1) \sqcap \ldots \sqcap (\star P^{R}_{E_n} : E_n) \end{array}$$

expressing that whenever the entity (relationship) is true it is necessarily true in every past and future time point. Indeed, it is a direct consequence of the $\mathcal{ALCQIT}$ semantics that instances of snapshot entities (relationships) have necessarily a global lifetime, i.e., they are supposed to never change along time. To capture snapshot relationships, we also need to force each ER-role to be time invariant. For this purpose, the so called *global features* are needed. They are features whose value does not depend on time: we will indicate such particular kind of feature by prefixing the feature name with a "$\star$". Atomic global features are interpreted as partial functions independent from time: $\forall t, v \in \mathcal{P}.\star g^{\mathcal{I}(t)} = \star g^{\mathcal{I}(v)}$.
Entities and relationships that are not constrained by any formula have temporally indeterminate instances, i.e., they can admit both global and temporary instances. In the following, we use the notion of *weak temporary* entity or relationship to indicate them. In practice, if we imagine to translate entities and relationships to a relational data model we should con-

sider time-stamped relations for both temporary and weak temporary classes, while we do not need time-stamped relations for snapshot classes. Thus, snapshot formulas are suitable for capturing entities and relationships that are supposed to never change during the life cycle of the temporal database. Furthermore, snapshot entities and relationships are used to capture the semantics of legacy diagrams when included in a temporal model, thus enforcing the upward compatibility.

Please note that, while an integrity constraint involving future times – like one of the above – may have little sense for checking the integrity of a temporal database where no stored fact has a valid time greater than the current time, it may have a lot of sense for checking the satisfiability or for computing the set of logically implied facts in a conceptual model. Using the reasoning capabilities of $\mathcal{ALCQIT}$ it is possible to support the database designer to discover relevant schema properties. With reference to Figure 1, let us consider the following example:

$\texttt{Employee} \sqsubseteq (\Box^+\texttt{Employee}) \sqcap (\Box^-\texttt{Employee})$
$\texttt{Department} \sqsubseteq (\Box^+\texttt{Department}) \sqcap (\Box^-\texttt{Department})$
$\texttt{Manager} \sqsubseteq (\Diamond^+\neg\texttt{Manager}) \sqcup (\Diamond^-\neg\texttt{Manager})$
$\texttt{Responsible-for} \sqsubseteq$
$(\Box^+\texttt{Responsible-for}) \sqcap (\Box^-\texttt{Responsible-for})$
$\texttt{Responsible-for} \sqsubseteq$
$(\star\texttt{resp-for} : \texttt{Project}) \sqcap$
$(\star\texttt{resp-of} : \texttt{OrganisationalUnit})$
$\texttt{Works-for} \sqsubseteq (\Diamond^+\neg\texttt{Works-for}) \sqcup (\Diamond^-\neg\texttt{Works-for})$

We have asserted that Employee and Department are snapshot entities, Manager is a temporary entity, Responsible-for is a snapshot relationship, and Works-for is a temporary relationship. Among the valid logical implications or consistency checks involving timestamps, the following are classical cases found in the literature which apply to the example:

- Sub-entities of temporary entities must be temporary – thus, since Manager is a temporary entity, constraining either AreaManager or TopManager as snapshot entities would result in an inconsistency.
- Sub-entities of snapshot entities can be either snapshot, temporary or weak temporary entities – thus, it is consistent having Manager as a sub-entity of Employee.
- Super-entities of snapshot entities must be snapshot – thus, the fact that OrganisationalUnit must be necessarily snapshot is a valid logical implication.
- Super-entities of (weak) temporary entities can be either snapshot, temporary or weak temporary entities.
- A schema is inconsistent if exactly one of a whole set of snapshot partitioning sub-entities is temporary – thus, the fact that InterestGroup must be a snapshot entity is a valid logical implication.
- Participants of snapshot relationships must be snapshot entities – thus, Project must be a snapshot entity.
- Participants of (weak) temporary relationships can be either snapshot, temporary or weak temporary entities – thus, it is correct for the snapshot entity Project to participate in the weak temporary relationship Manages.

5.2 Dynamic entities

$\mathcal{ALCQIT}$ allows us to model objects whose properties change along time; this is called in the temporal database community *object migration* from a source to a target entity. Indeed, in the temporal conceptual modelling literature two notions of object migration are introduced [Gupta and Hall, 1992; Spaccapietra *et al.*, 1998]: *evolution*, when an object ceases to be an instance of a source entity; *extension*, when an object continues to belong to the source. Let E_s be the source entity and E_t be the target; the case of dynamic extension is captured by the following $\mathcal{ALCQIT}$ formula:

$$E_t \sqsubseteq (\Diamond^- E_s)$$

Thus, every instance of the target entity should have been an instance of the source at some point in the past. To enforce dynamic evolution both the dynamic extension formula and a disjointness formula are required:

$$E_t \sqsubseteq (\Diamond^- E_s)$$
$$E_t \sqsubseteq \neg E_s$$

An interesting consequence of dynamic evolution is that the target is necessarily a temporary entity while the source can be both temporary or weak temporary. $\mathcal{ALCQIT}$ can force *uni-directional evolutions* by adding to the dynamic evolution the following formula constraining the past of the source:

$$E_s \sqsubseteq (\Box^- \neg E_t)$$

i.e., when the target (E_t) is reached, the object can not be involved a second time in the source (E_s). For example, while there can be either a dynamic extension or a dynamic evolution between being a student and being an employee, there is a uni-directional evolution between being alive and being dead. $\mathcal{ALCQIT}$ can also enforce the absence of an evolution between two entities:

$$E_s \sqsubseteq (\Box^- \neg E_t) \sqcap (\Box^+ \neg E_t)$$

A typical example of incompatible entities can be male and female. A generalisation hierarchy with dynamic entities is called *dynamic hierarchy*, while a generalisation hierarchy with incompatible entities is called *static hierarchy* [Spaccapietra *et al.*, 1998].

6 Conclusions

This work gives a logical formalisation of a temporal ER model, which has the ability to express both enhanced temporal ER schemas and (temporal) integrity constraints in the form of general axioms imposed on the schema itself. The proposed formal language we have proposed is a member of the family of Description Logics. Reasoning tasks over the temporal representations have been defined. We have shown how the integrity constraints can encode the distinction between time-varying and snapshot constructs, and the meaning of dynamic entities.

The most promising research direction to be explored is to better characterise the expressivity of temporal integrity constraints in order to axiomatise several extensions as proposed in the literature of temporal ER models. Other challenging research directions we are currently exploring are: a computational characterisation of the query containment problem under temporal constraints; and the study of the computational properties of a temporal logic able to capture just the needed expressivity for temporal conceptual modelling.

Acknowledgements

This paper is a revised version of [Artale and Franconi, 1999b; 1999a]. The work was partially supported by the "Foundations of Data Warehouse Quality" (*DWQ*) European ESPRIT IV Long Term Research (LTR), and by the British EPSRC research council projects "Formalising the Design of temporal Databases using Description Logics" and "Knowledge representation meets Databases".

REFERENCES

[Artale and Franconi, 1998] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 9:463–506, 1998.

[Artale and Franconi, 1999a] A. Artale and E. Franconi. Reasoning with enhanced temporal entity-relationship models. In *Proc. of the International Workshop on Spatio-Temporal Data Models and Languages*. IEEE Computer Society Press, August 1999. Also in Proc. of the 6^{th} International Workshop on Knowledge Representation meets Databases (KRDB'99), and in Proc. of the 1999 International Workshop on Description Logics (DL'99).

[Artale and Franconi, 1999b] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*. Springer-Verlag, November 1999.

[Artale and Franconi, 2000] A. Artale and E. Franconi. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence*, 2000. To appear.

[Artale and Franconi, 2001] A. Artale and E. Franconi. Temporal description logics. In M. Fisher, D. Gabbay, and L. Vila, editors, *Handbook of Time and Temporal Reasoning in Artificial Intelligence*. MIT Press, 2001. To appear.

[Artale *et al.*, 2001] A. Artale, E. Franconi, M. Mosurovic, F. Wolter, and M. Zakharyashev. Temporal description logics for conceptual modelling: expressivity and complexity. Manuscript, 2001.

[Calvanese *et al.*, 1994] D. Calvanese, M. Lenzerini, and D. Nardi. A unified framework for class-based representation formalisms. In *Proc. of the 4 th International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany, May 1994.

[Calvanese *et al.*, 1998] D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*. Kluwer, 1998.

[Calvanese *et al.*, 1999] D. Calvanese, G. De Giacomo, M. Lenzerini, and D. Nardi. Reasoning in expressive description logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier Science Publishers, Amsterdam, 1999. To appear.

[Chomicki and Toman, 1998] J. Chomicki and D. Toman. Temporal logic in information systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, chapter 1. Kluwer, 1998.

[Gabbay *et al.*, 1994] D. M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.

[Gregersen and Jensen, 1999] H. Gregersen and J. S. Jensen. Temporal Entity-Relationship models - a survey. *IEEE Transaction on Knowledge and Data Engineering*, 11(3):464–497, 1999.

[Gupta and Hall, 1992] R. Gupta and G. Hall. An abstraction mechanism for modeling generation. In *Proc. of ICDE'92*, pages 650–658, 1992.

[Jensen *et al.*, 1994] C. S. Jensen, M. Soo, and R. T. Snodgrass. Unifying temporal data models via a conceptual model. *Information Systems*, 9(7):513–547, 1994.

[Schild, 1993] K. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6^{th} Portuguese Conference on Artificial Intelligence, EPIA'93*, October 1993.

[Spaccapietra *et al.*, 1998] S. Spaccapietra, C. Parent, and E. Zimanyi. Modeling time from a conceptual perspective. In *Int. Conf. on Information and Knowledge Management (CIKM98)*, 1998.

[Wolter and Zakharyaschev, 1998] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In *Proceedings of FroCoS'98*, Amsterdam, NL, 1998.

[Wolter and Zakharyaschev, 1999] F. Wolter and M. Zakharyaschev. Modal description logics: Modalizing roles. *Fundamentæ Informaticæ*, 39:411–438, 1999.

[Wolter *et al.*, 2001] F. Wolter, M. Zakharyaschev, and M. Mosurovic. Temporal description logics: a point-based approach. *This volume*, 2001.