Reasoning with enhanced Temporal Entity-Relationship Models

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

Abstract

Recent efforts in the Conceptual Modelling community have been devoted to properly capturing time-varying information, and several proposals of temporally enhanced Entity-Relationship (ER) exist. 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. Reasoning in the devised logic is decidable, thus allowing for automated deductions over the whole conceptual representation, which includes both the ER schema and the integrity constraints over it.

1 Introduction

In the temporal ER community two different main modelling approaches have been devised to provide support for the conceptualisation of valid time. 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 meaning 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 the expressive temporal Description Logic (DL) $\mathcal{ALCQIT}$ DL, which is able to capture conventional ER models and has the ability to express a powerful class of temporal inclusion dependency constraints. Advantages of $\mathcal{ALCQIT}$ is its high expressivity combined with desirable computational properties – such as decidability, soundness and completeness of deduction procedures, allowing for a complete calculus for temporal integrity constraints.

The paper is organised as follows. Section 2 introduces the temporally enhanced ER model, in both the implicit and explicit approaches. Section 3 briefly introduces the adopted temporal DL. Section 4 will show how temporal ER schemas can be encoded into the temporal DL, 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 time-varying and snapshot constructs.

2 The Temporal ER Model

In this Section we informally introduce the temporally enhanced ER model. Let us consider first a standard ER diagram, i.e., a diagram 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. We have defined a first-order semantics for the temporally enhanced ER model – in the implicit approach –

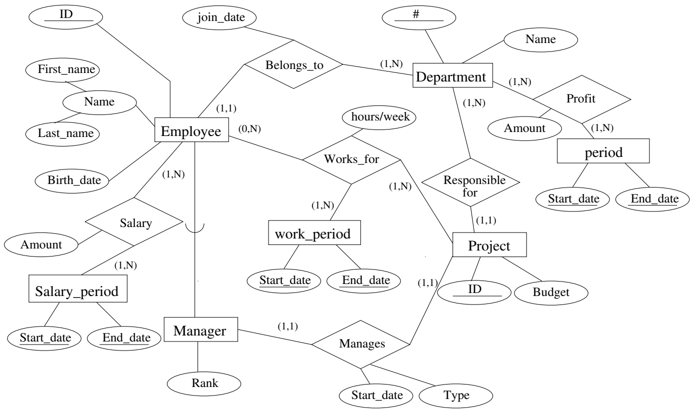

Figure 1. The temporal ER diagram.

by extending the non temporal semantics introduced in [3]. Interpretations of a temporal ER diagram are called *legal database states*. Intuitively, a legal database state is a temporally dependent database – i.e., a finite relational structure whose tuples depend on time – which conforms to the constraints imposed by the schema.

Let us consider the example ER diagram of Figure 1; this diagram is the running example considered in the survey paper [6]. As we have noticed before, the implicit approach does not consider the temporal constructs related to the validity time of entities and relationships (see, e.g., the TEER model in [4]). Thus, the example diagram should be modified, since there are some of those disallowed constructs. The *Profit* relationship becomes an attribute of the entity *Department*, the *Salary* relationship becomes an attribute of the entity *Employee*, the *Work-period* entity disappears, since it just denotes the validity time of the relationship *Works-for*. The resulting diagram is such that *every* construct has its own implicit validity time.

We consider now an enhancement of the temporal ER model by means of integrity constraints. The following constraints may be imposed over the example ER diagram:

- 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.

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. The enriched schema, which includes both the ER diagram and the integrity constraints, logically implies the following:

- for every project, there is at least an employee who is not a manager,
- each 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.

$$
\begin{array}{lll}
C, D \to & 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 \\
& & \qquad \forall w. (t < w < v) \to 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 \\
& & \qquad \forall w. (v < w < t) \to 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 \to C^{\mathcal{I}(v)}(i)\} \\
& \Box^{-} C & \{i \in \Delta^{\mathcal{I}} \mid \forall v. v < t \to C^{\mathcal{I}(v)}(i)\} \\
R, S \to & P \mid & \\
& f \mid & \\
& R^{-1} \mid & \{(i,j) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}(t)}(j,i)\} \\
& R|_C \mid & R^{\mathcal{I}(t)} \cap (\Delta^{\mathcal{I}} \times C^{\mathcal{I}(t)}) \\
& R \circ S & R^{\mathcal{I}(t)} \circ S^{\mathcal{I}(t)}
\end{array}
$$

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

In the case where an *explicit* approach to provide temporal support is adopted, new constructs are usually 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 realizable within the standard implicit temporal approach. According to our approach, both entities and relationships in the explicit temporal ER model can be either unmarked, in what case they are considered *snapshot* constructs (i.e., each of their instances has a global lifetime, as in the case they derive from a legacy diagram), or explicitly *temporary* marked (i.e., each of their instances has a temporary lifetime).

3 The Temporal Description Logic

We introduce very briefly in this section the $\mathcal{ALCQIT}$ temporal DL, which is obtained by combining a standard tense logic and the standard non-temporal $\mathcal{ALCQI}$ DL [2].

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.

The tense-logical extension of $\mathcal{ALCQI}$ has been inspired by the works of [9, 12]. It is possible to show that reasoning in $\mathcal{ALCQIT}$ (i.e., deciding knowledge base satisfiability and deciding logical implication) is decidable; the proof is based on a reduction to the decidable language introduced in [12]. The computational complexity of reasoning in $\mathcal{ALCQIT}$ is EXPTIME-hard.

As an example let us consider the axiom stating that any living mortal should live in some place, remains alive until it will die, and at some point in the past was born:

$$\begin{aligned}&\texttt{Mortal} \sqcap \texttt{LivingBeing} \sqsubseteq (\exists \texttt{LIVES-IN.Place}) \sqcap \\ &\quad (\texttt{LivingBeing}\, \mathcal{U}\, \Box^{+} \neg \texttt{LivingBeing}) \sqcap \\ &\quad (\texttt{LivingBeing}\, \mathcal{S}\, \Box^{-} \neg \texttt{LivingBeing})\end{aligned}$$

4 Encoding the Implicit Model

We show in this Section how an ER schema with implicit representation of time – as informally introduced in Section 2 – can be expressed as a $\mathcal{ALCQIT}$ knowledge base. Let us first consider the translation from an ER diagram (without considering the integrity constraints) to a $\mathcal{ALCQIT}$ knowledge base: an ER diagram $\mathcal{D}$ is translated according to table 3 into a corresponding knowledge base Σ where each domain, entity or relationship symbol corresponds to a primitive concept, and each attribute or ER-role symbol corresponds to a primitive feature.

Temporal integrity constraints are expressed by means of additional terminological axioms in Σ. More precisely, an integrity constraint is any inclusion dependency which can be expressed by means of a terminological axiom of the kind $C \sqsubseteq D$. It is important to emphasise the fact that in this approach the integrity constraints are part of the schema, so that reasoning is carried on by taking in complete account all the information contained in the schema.

$\mathcal{D}$	Σ
ISA link between two entities E, F	$E \sqsubseteq F$
attribute A with domain D of an entity E	$E \sqsubseteq A : D$
ISA link between two relationships R, S	$R \sqsubseteq S$
attribute A with domain D of a relationship R	$R \sqsubseteq A : D$
relationship R relating n entities $E_1 \dots E_n$ by means of the ER-roles $P^R_{E_1} \dots P^R_{E_n}$	$R \sqsubseteq (P^R_{E_1} : E_1) \sqcap \dots \sqcap (P^R_{E_n} : E_n)$
minimum cardinality constraint $n \neq 0$ in a ER-role P^R_E relating a relationship R with and entity E	$E \sqsubseteq \geq n\, (P^R_E)^{-1}.R$
maximum cardinality constraint $n \neq \infty$ in a ER-role P^R_E relating a relationship R with and entity E	$E \sqsubseteq \leq n\, (P^R_E)^{-1}.R$

Figure 3. The translation ER $\to$ DL.

Based on the results of [3], we have proved a theorem stating 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. 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 – for example, by using a temporal extension of a state-of-the-art DL system such as iFaCT [7]. 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 [3].

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 DL knowledge base Σ_{ER}:

$$\begin{aligned}&\texttt{WORKS-FOR} \sqsubseteq \texttt{has-prj} : \texttt{Project} \sqcap \texttt{has-emp} : \texttt{Employee}\\ &\texttt{Project} \sqsubseteq \exists \texttt{has-prj}^{-1}.\texttt{WORKS-FOR}\\ &\texttt{Manager} \sqsubseteq \texttt{Employee}\end{aligned}$$

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:
  $\texttt{Employee} \sqcap \forall \texttt{has-emp}^{-1}.\neg\texttt{WORKS-FOR} \sqsubseteq \texttt{Manager}$
- A manager becomes qualified after a period when she/he was just an employee:
  $\texttt{Manager} \sqsubseteq \texttt{Qualified}\ \mathcal{S}\ (\texttt{Employee} \sqcap \neg\texttt{Manager})$

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:
  $\Sigma_{ER} \cup \Sigma_{IC} \models$
  $\texttt{Project} \sqsubseteq \exists(\texttt{has-prj}^{-1} \circ \texttt{has-emp}).\neg\texttt{Manager}$
- A manager worked in a project before managing some (possibly different) project:
  $\Sigma_{ER} \cup \Sigma_{IC} \models$
  $\texttt{Manager} \sqsubseteq \Diamond^{-}\exists(\texttt{has-emp}^{-1} \circ \texttt{has-prj}).\texttt{Project}$

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 Model

This Section shows how the proposed formalisation can encode explicit temporal ER models by simply imposing specific constraints defining snapshot and temporary constructs, thus maintaining upward compatibility.

5.1 Snapshot Vs. Temporary Entities

The $\mathcal{ALCQIT}$ DL is able to capture explicit temporal ER models by first applying the translation given in the previous Section, and then adding precise axioms to distinguish between snapshot and temporal constructs. In the following, axioms for entities are illustrated. In the next Section, the analogous for relationships will be showed.

A snapshot entity is axiomatised as follows:

$$E \sqsubseteq (\Box^{+} E) \sqcap (\Box^{-} E) \qquad \textit{(Snapshot axiom)}$$

expressing that whenever the entity is true it is necessarily true in every past and future time point. Indeed, instances of snapshot entities have necessarily a global lifetime. On the other hand, a temporary entity is axiomatised by the following constraint:

$$E \sqsubseteq (\Diamond^{+}\neg E) \sqcup (\Diamond^{-}\neg E) \qquad \textit{(Temporary axiom)}$$

asserting that there must be a past or future time point where the entity does not hold. Indeed, instances of temporary entities have necessarily a limited lifetime.

Using the reasoning capabilities of $\mathcal{ALCQIT}$ it is possible to support the database designer to discover relevant schema properties. As an example of the logical implications holding in a diagram making use of both snapshot and temporary entities, let us consider the interaction between entities via ISA links. Let us suppose that there is an ISA link between a snapshot entity E_1 and a temporary entity E_2. This temporal ER diagram is translated into the following unsatisfiable knowledge base:

$$E_1 \sqsubseteq (\Box^{+} E_1) \sqcap (\Box^{-} E_1)$$
$$E_2 \sqsubseteq (\Diamond^{+}\neg E_2) \sqcup (\Diamond^{-}\neg E_2)$$
$$E_1 \sqsubseteq E_2$$

Thus, a snapshot entity can not be a subclass of a temporary entity, this is true also whenever such a kind of taxonomic relation is derived in the temporal ER model.

From these considerations it is easy to understand why the following implications hold:

$$\{\ E_2 \sqsubseteq (\Diamond^{+}\neg E_2) \sqcup (\Diamond^{-}\neg E_2),\ E_1 \sqsubseteq E_2\ \} \models E_1 \sqsubseteq (\Diamond^{+}\neg E_1) \sqcup (\Diamond^{-}\neg E_1)$$
$$\{\ E_1 \sqsubseteq (\Box^{+} E_1) \sqcap (\Box^{-} E_1),\ E_1 \sqsubseteq E_2\ \} \models E_2 \sqsubseteq (\Box^{+} E_2) \sqcap (\Box^{-}\neg E_2)$$

i.e., necessarily, every subclass of a temporary entity must be temporary; and a superclass of a snapshot entity must be a snapshot entity. Conversely, nothing can be said with respect to subclasses of snapshot entities. For example, a schema where a temporary entity is a subclass of a snapshot entity is consistent.

An incorrect ER schema can be the result of disjoint subclasses – i.e., a partitioning. A schema is inconsistent if exactly one of a whole set of snapshot disjoint subclasses is temporary [8]. Without loss of generality, let us illustrate the case where E_1, E_2 are disjoint subclasses of the entity E, with E_1 snapshot and E_2 temporary, then such an ER schema is inconsistent. Indeed, the corresponding knowledge base is unsatisfiable (note that the first set of axioms correspond to the disjoint subclass axioms):

$$E \sqsubseteq E_1 \sqcup E_2, \quad E_1 \sqsubseteq E \sqcap \neg E_2, \quad E_2 \sqsubseteq E$$
$$E_1 \sqsubseteq (\Box^{+} E_1) \sqcap (\Box^{-} E_1)$$
$$E_2 \sqsubseteq (\Diamond^{+}\neg E_2) \sqcup (\Diamond^{-}\neg E_2)$$

The following is an immediate consequence of the above inconsistent schema:

$$\{\ E \sqsubseteq E_1 \sqcup E_2, \quad E_1 \sqsubseteq E \sqcap \neg E_2, \quad E_2 \sqsubseteq E, \quad E_1 \sqsubseteq (\Box^{+} E_1) \sqcap (\Box^{-} E_1)\ \} \models E_2 \sqsubseteq (\Box^{+} E_2) \sqcap (\Box^{-} E_2)$$

i.e., an ER schema with exactly one entity whose temporal behaviour is unknown among a whole set of snapshot disjoint subclasses, implies that this entity is snapshot.

5.2 Snapshot Vs. Temporary Relationships

The case for relationships is more complex. Temporary relationships are captured by enforcing the *temporary axiom* on relationships – in a way analogous to the case of temporary entities:

$$R \sqsubseteq (\Diamond^{+}\neg R) \sqcup (\Diamond^{-}\neg R) \qquad \textit{(Temporary axiom)}$$

To capture snapshot relationships, in addition to the *snapshot axiom*, we 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: $\forall t, v \in \mathcal{P}. \star g^{\mathcal{I}(t)} = \star g^{\mathcal{I}(v)}$.

Using global features instead of generic features for ER-roles defining a relationship results in a *homogeneous relationship* – i.e., a relationship with tuples whose values are valid at the same time period. Homogeneous relationships are encoded by means of the following axiom:

$$R \sqsubseteq (\star P^{R}_{E_1} : E_1) \sqcap \ldots \sqcap (\star P^{R}_{E_n} : E_n) \qquad \textit{(Homog. ax.)}$$

Snapshot relationships are necessarily global and homogeneous relationships. Thus, if R is a snapshot relationship involving the entities $E_1, \ldots, E_n$, the following axioms should be added to Σ:

$$R \sqsubseteq (\Box^{+} R) \sqcap (\Box^{-} R)$$
$$R \sqsubseteq (\star P^{R}_{E_1} : E_1) \sqcap \ldots \sqcap (\star P^{R}_{E_n} : E_n)$$

The two axioms are such that whenever a tuple belongs to a snapshot relationship, then the very same tuple is assumed to belong to the relationship at every time.

The interaction between temporal and snapshot constructs can result in an inconsistent ER schema that can be checked and discarded automatically. This is case when a snapshot relationship R involves a temporary entity. Indeed, the following knowledge base is unsatisfiable:

$$R \sqsubseteq (\Box^{+} R) \sqcap (\Box^{-} R)$$
$$R \sqsubseteq (\star P^{R}_{E_1} : E_1) \sqcap \ldots \sqcap (\star P^{R}_{E_n} : E_n)$$
$$E_i \sqsubseteq (\Diamond^{+}\neg E_i) \sqcup (\Diamond^{-}\neg E_i)$$

i.e., snapshot relationships cannot have temporary entities as participants.

On the other hand, temporary relationships admit snapshot entities since the entity instances participate in the relationship only for a temporary time – i.e., during the validity time of the relationship.

6 Conclusions

This preliminary 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 formal language we have proposed is a member of the family of Description Logics, and it has a decidable reasoning problem, thus allowing for automated deduction over the whole conceptual representation. We have also shown how the integrity contraints can encode the distinction between time-varying and snapshot constructs.

This work is just at the beginning. 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. Currently, we are exploring the possibility to axiomatise the difference between *homogeneous* and *heterogeneous* relationships [5, 10], and to express *historical* marks (H-marks) [11].

This work was partially supported by the "Foundations of Data Warehouse Quality" (*DWQ*) European ESPRIT IV Long Term Research (LTR) Project 22469.

References

[1] A. Artale and E. Franconi. Temporal ER modeling with description logics. In *Proc. of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99)*, 1999. Also in Proc. of the 1999 International Workshop on Description Logics (DL'99).

[2] 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, 1999. To appear.

[3] 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.

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

[5] S. Gadia. A homogeneous relational model and query languages for temporal databases. *ACM Transactions On Database Systems*, 13:418–448, 1988.

[6] H. Gregersen and J. S. Jensen. Temporal Entity-Relationship models - a survey. *IEEE Transactions on Knowledge and Data Engineering*, 1999. To appear.

[7] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 1999. To appear.

[8] P. McBrien, A. Seltveit, and B. Wangler. An Entity-Relationship model extended to describe historical information. In *Proc. of CISMOD'92*, pages 244–260, 1992.

[9] K. D. Schild. Combining terminological logics with tense logic. In *Proc. of the 6th Portuguese Conference on Artificial Intelligence (EPIA'93)*, 1993.

[10] A. Tansel and E. Tin. Expressive power of temporal relational query languages and temporal completeness. In O. Etzion, S. Jajodia, and S. Sripada, editors, *Temporal Databases - Research and Practice*, pages 129–149. Springer-Verlag, 1998.

[11] C. Theodoulidis, P. Loucopoulos, and B. Wangler. A conceptual modelling formalism for temporal database applications. *Information Systems*, 16(3):401–416, 1991.

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