Schema Translation between Heterogeneous Data Models in a Lattice Framework*

P. Atzeni and R. Torlone
Dipartimento di Discipline Scientifiche
Terza Università di Roma
Via Segre, 2 – 00185 Roma, Italy
email: {atzeni,torlone}@inf.uniroma3.it

Abstract

In this paper we study the problem of translating schemes between different data models, in a formal framework that refers to a wide range of models. We first introduce a graph-theoretic formalism that allows us to uniformly represent schemes and models, to compare different data models and to describe the behavior of basic translations. The formalism is based on a classification of the constructs used in the known data model into a limited set of types. Then, we study in this framework formal properties of scheme translation between heterogeneous data models, and we develop a method for deriving translations that enjoy those properties.

Keywords

Data models, Metamodel, Heterogeneity, Schema translation, Lattice structure.

1 INTRODUCTION

1.1 Motivations and background

It is widely accepted now that a conceptual data model be used in the analysis phase and many tools exist that support the analysis and design of information systems (see for instance the book of Batini et al. (1992)). At the same time, many data models have been defined (see the survey of Hull and King (1987), and the book of Tsichritzis and Lochovski (1982)), and each of the tools usually implements only one of the models. Incidentally, although it is true that most of the tools support the Entity-Relationship model, introduced by Chen (1976), it is also true that there are in fact many versions of this model, which represent actually different models, often compatible only to a limited

*This work was partially supported by Consiglio Nazionale delle Ricerche and by MURST.

extent. It is reasonable that in a complex environment different models should coexist, for a number of reasons: (i) different subproblems may be analyzed independently (for examples by companies that later merged or got involved in a federated project) or (ii) different analysts may prefer different models; or (iii) different subproblems could be tackled with different models, because of the specific aspects of each.

We have already discussed the general ideas of our approach (Atzeni and Torlone (1993)). Our long term goal is an environment that allows the definition of "any reasonable model," by means of a suitable formalism called a *metamodel*. Then, for any two models M_1 and M_2 defined in this way, and for each scheme S_1 (the *source scheme*) of M_1 (the *source model*), it should be possible to obtain a scheme S_2 (the *target scheme*) that be the translation of S_1 into M_2 (the *target model*). A major point in this plan is related to the expressive power of the metamodel, that is, the set of models that can be defined. In fact, the notion of a model is widely accepted and understood, but there is no general, formal definition. This problem can be overcome by noting that, according to a classification of Hull and King (1987), all the constructs used in most known models fall in a rather limited set of categories: lexical type, abstract type, aggregation, generalization, function, grouping. Therefore, we have argued (see Atzeni and Torlone (1993)) that a metamodel can be defined by means of a basic set of *metaconstructs*, corresponding to the above categories. Then, a model can be described by defining its constructs by means of the metaconstructs in the metamodel. The generality and variety of the metaconstructs determine the expressive power of the metamodel. In a sense, this approach is "asymptotically" complete: if there is a model that cannot be expressed by means of the metamodel, because a construct in the model has no counterpart in the metamodel, then the metamodel can be extended by introducing a new, suitable metaconstruct. We believe that a wide class of models can be managed with a few extensions to the basic metamodel.

A second key point in our approach is that there is no clear notion of when a translation is correct: at first, one could think that the target scheme should be "equivalent" to the source scheme, that is, they should represent "the same information." A lot of research has been conducted in the last decades on scheme equivalence (or "comparison of information capacity") with reference to the relational model (refer for instance to Atzeni et al. (1982), Hull (1986) and Rissanen (1982)) or to heterogeneous frameworks (Abiteboul and Hull (1988), Kalinichenko (1990), Lien (1982), and Miller et al. (1993)), but there is no general, agreed definition.

To the best of our knowledge, there is not much literature related to the problem we tackle and the goal we set. Some work exists on the idea of a metamodel for the representation of models (Barsalou and Gangopadhyay (1992), and Kalinichenko (1990)), but the goal is more on the integration of heterogeneous databases in a federated environment (Sheth and Larson (1990) provided a survey on federated database systems) than on the translation of schemes to generic target models.

On the basis of the above arguments, one of the conclusions of our preliminary study was that in order to study transformations between different data models, a pragmatic approach is needed. As a matter of fact, the various constructs that correspond to the same metaconstruct can be assumed to have the same semantics (at least with respect to the translation process). For example, as argued by Hull and King (1987), the semantics of entities is the same in all versions of the E-R model, as well as in the functional model and corresponds to the semantics of abstracts (or non-printable) types. As a consequence,

the process of scheme translation can be based on translations of constructs (or simple combinations thereof) defined with respect to the corresponding metaconstruct. In this way, schemes and translations need not refer to specific models. For example, a scheme S_1 with entities, binary relationships and is-a hierarchies belongs to all versions of the E-R model that allow is-a hierarchies, whereas a scheme S_2 with binary and ternary relationships and no hierarchies belongs to all versions that allow ternary relationships. The translation of S_1 into the model of S_2 would be performed within the framework of a model that allows both ternary relationships and is-a hierarchies, but its result would not include is-a hierarchies.

Another result of our preliminary study was that the translation process can be obtained as the composition of a predefined set of elementary transformations which implement the standard translations between constructs studied in the literature (see Batini et al. (1987)). The operation that eliminates is-a hierarchies (and, say, replaces them with binary relationships) is an example of basic translation that should be used in many translations that go from a model with hierarchies to a model without them.

1.2 Contributions of the paper

The approach has been studied within a graph-theoretic framework that allows us to define in a uniform way schemes and models (Atzeni and Torlone (1994)). We have the notion of a *structure*, a directed graph whose nodes have different types (corresponding to the basic metaconstructs we mentioned above, such as lexical, abstract, aggregation and function). Then, we use structures in two ways: we have *patterns*, structures where the edges have cardinalities as labels, and *schemes*, structures whose nodes and edges are labeled with names. Then a model M is defined by means of a set of patterns $\mathcal{P}$: a scheme S *is allowed* in M if S can be mapped (according to a specific, but natural notion) to one of the patterns in $\mathcal{P}$. In this way, there is a set of schemes associated with each model. Clearly, a scheme can be allowed in several models. Our major result (1994) is that a partial order can be introduced on patterns, which, suitably extended, becomes a lattice on sets of patterns.

The main goal of this paper is to study within this framework how elementary transformations can be composed, in order to form complex translations: assuming that the basic transformations are correct, we first introduce correctness and other desirable properties of complex translations, and then we present techniques for finding translations that satisfy those properties. In a sense, this could be called an "axiomatic" approach that is coherent with the observation made above on the difficulty in defining correct translations. Now, elementary transformations are described on the basis of the patterns they eliminate and the patterns they introduce: clearly, this is only part of their description (we say this is the *signature* of a translation step, as opposed to its *body* or *program*), but it is sufficient for our purposes. The main point of the approach is that if a transformation eliminates all constructs of a scheme that correspond to a pattern P_1 and replaces constructs corresponding to a pattern P_2, then, from an intuitive point of view, its application to a set of schemes described by a set of patterns $\mathcal{P}$ generates schemes described by another set of patterns (intuitively, $\mathcal{P} - \{P_1\} \cup \{P_2\}$. In this way, a translation from a model M_0 (described by the set of patterns $\mathcal{P}_0$) to another model M (described by $\mathcal{P}$) can be seen as a sequence of elementary translations $\tau_1, \ldots, \tau_k$ such that there is a sequence of sets

of patterns $\mathcal{P}_1, \ldots, \mathcal{P}_k$ such that $\mathcal{P}_k = \mathcal{P}$ and τ_i applied to schemes described by $\mathcal{P}_{i-1}$ generates schemes described by $\mathcal{P}_i$.

Thus, the main contributions of this paper are the definition and characterization of desirable properties of translations, and the development of a methodology for the automatic generation of translations that satisfy such properties. The results are obtained in an elegant way by means of the lattice framework on patterns.

1.3 Plan of the paper

The paper is organized as follows. In Section 2 we describe our graph-theoretic formalism for the various components of the metamodel, and show its properties. This formalism is used in Section 3 to define schemes, models and translations of schemes. In Section 4, we study formal properties on the activity of scheme translation and provide several characterizations. On the basis of these results, in Section 5 we develop a general method to derive translations between models. In Section 6, we show a detailed example to illustrate the various notions introduced throughout the paper. Finally, in Section 7, we sketch some conclusions. Because of space limitation, the proofs of the various results of this paper are not given here.

2 A GRAPH-THEORETIC FORMALISM

2.1 Structures and Patterns

We fix a set of *node types* $\mathcal{N}$ and the set of *edge types* $\mathcal{E}$: they will be used to represent the metaconstructs of the metamodel. We point out that the approach is essentially independent on the (meta)constructs used and therefore we assume that the elements in $\mathcal{N}$ and in $\mathcal{E}$ are chosen according to the specific application. The only restriction we impose is that, following a notation used by Hull and King (1987), *atomic types* and *type constructors* are represented with nodes, whereas *functions* (or *attributes*) and applications of type constructors to other types are represented with edges. In our examples, we will consider three types of nodes corresponding to *abstracts* (denoted by the symbol $\triangle$), *aggregations* ($\otimes$), and *lexicals* ($\Box$); and six types of edges corresponding to *functions* (denoted by $\rightarrow$), *multivalued functions* ($\twoheadrightarrow$), *components of aggregation* ($\longrightarrow\!\!\triangleright$), *keys of aggregation* ($-\!\!\bullet\!\!\triangleright$), *keys of abstract* ($-\!\!\bullet\!\!\rightarrow$) and *subset relations between abstracts* ($\Rightarrow$).

Definition 1 *A* structure *is a triple* $S = (G, \mu, \epsilon)$ *where* $G = (N, E)$ *is a directed acyclic graph, and* μ *and* ϵ *are* structuring functions*:* $\mu : N \rightarrow \mathcal{N}$ *and* $\epsilon : E \rightarrow \mathcal{E}$.

We assume that every structure satisfies a number of conditions corresponding to the usual restrictions on the composition of constructs. Note that for sake of simplification, we have assumed that a structure has no directed cycles. This is not a significant limitation in practice: intuitively, "cyclic" schemes can be obtained by composing structures, in the same way as in conceptual models cycles in the schemes appear, without recursive constructs (a formal justification for this argument was given by Kuper (1985)).

Given two structures $S_1 = ((N_1, E_1), \mu_1, \epsilon_1)$ and $S_2 = ((N_2, E_2), \mu_2, \epsilon_2)$, a *mapping* Φ from S_1 to S_2 is a pair of functions $\theta : N_1 \to N_2$ and $\phi : E_1 \to E_2$. A mapping $\Phi = (\theta, \phi)$ is *structure-preserving* if: (1) for each $n \in N_1$, $\mu_1(n) = \mu_2(\theta(n))$, and (2) for each $e = (n, n') \in E_1$, $\epsilon_1(e) = \epsilon_2(\phi(e))$ and $\phi(e) = (\theta(n), \theta(n'))$. If θ and ϕ are bijections then we say that Φ is an *isomorphism* and that S_1 and S_2 are *isomorphic* (note that two finite isomorphic structures are indeed identical).

Let $\mathbf{N}$ be the set of natural numbers and $\mathbf{N}^+$ be the set of positive natural numbers. A *range of cardinality* (or simply, a *range*) is a pair (n, m) (*minimum* and *maximum* cardinality respectively) such that $n \in \mathbf{N}$, $m \in \mathbf{N}^+$ and $n \leq m$. If $x \in \mathbf{N}$ and $r = (n, m)$ is a range, then $x \in r$ if $n \leq x \leq m$.

Definition 2 *A* pattern *is a pair $P = (S, \rho)$ where $S = (G, \mu, \epsilon)$ is a structure such that G is a rooted tree, and ρ is a function that associates a range with each edge of G.*

Roughly speaking, a range denotes the number of times a certain edge can appear in a structure. Thus, a pattern describes a collection of structures that involve a specific composition of metaconstructs.

Two patterns $P_1 = (S_1, \rho_1)$ and $P_2 = (S_2, \rho_2)$ are *isomorphic* if S_1 and S_2 are isomorphic and each edge of P_1 is associated with an edge of P_2 with the same range.

For sake of conciseness, in our examples we will also represent patterns by means of a parenthetical notation (this is always possible since a pattern is a tree). Specifically: $n_1 \to n_2$ will be denoted by $n_1\langle n_2\rangle$, $n_1 \twoheadrightarrow n_2$ by $n_1\langle\langle n_2\rangle\rangle$, $n_1 \longrightarrow\!\triangleright n_2$ by $n_1[n_2]$, $n_1 \multimap\!\triangleright n_2$ by $n_1[n_2]^k$, $n_1 \multimap\!\!\to n_2$ by $n_1\langle n_2\rangle^k$, and $n_1 \Rightarrow n_2$ by $n_1[\![n_2]\!]$.

Now, we say that a structure S *matches* with a pattern $P = (S', \rho)$ if there is a structure-preserving mapping $\Phi = (\theta, \phi)$ from S to S' that verifies the following condition: given a node n of S let $E_n^{\bar{e}}$ denote the set of all edges e of S, outgoing from n, such that $\phi(e) = \bar{e}$ for some edge $\bar{e}$ of S'; then, for each node n of S and for each set of edges $E_n^{\bar{e}}$, it is the case that $|E_n^{\bar{e}}| \in \rho(\bar{e})$.

The main definition of this section establishes correspondences between structures and sets of patterns. A preliminary notion is needed: we say that the *components* of a structure S are the trees composed by all the nodes and edges reachable from the outer nodes of S.

Definition 3 *A structure S is an* instance *of a set of patterns $\mathcal{P}$ if, for each* component *S_i of S there is a pattern $P \in \mathcal{P}$ such that S_i matches with P. Given a set of patterns $\mathcal{P}$, we will denote with $Inst(\mathcal{P})$ the set of all instances of $\mathcal{P}$.*

Definition 4 *A set of patterns $\mathcal{P}_1$* is contained *in a set of patterns $\mathcal{P}_2$, in symbols $\mathcal{P}_1 \subseteq \mathcal{P}_2$, if $Inst(\mathcal{P}_1) \subseteq Inst(\mathcal{P}_2)$. If both $\mathcal{P}_1 \subseteq \mathcal{P}_2$ and $\mathcal{P}_2 \subseteq \mathcal{P}_1$ then we say that the two sets of patterns have* the same representation capacity *(in symbols, $\mathcal{P}_1 \equiv \mathcal{P}_2$).*

Example 1 *Figure 1 shows a pattern and one of its instances. The pattern represents unary and binary aggregations of abstracts, and (optional) functions from abstracts to lexicals. It is possible to see that the structure has two components (the trees having the aggregation node as the root) that match with the pattern.*

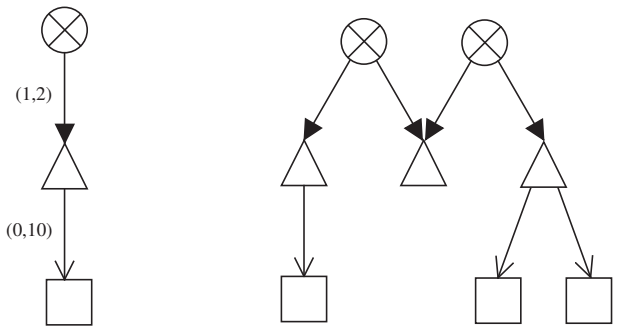

Figure 1 A pattern and one of its instances.

2.2 An algebra on patterns

Before introducing a number of operations on sets of patterns, we define a partial order relationship between them that yields a practical way to compare their representation capacity. It refers to patterns of a particular kind called *unitary*. We have shown (Atzeni and Torlone (1994)) that by using a set of *decomposition rules*, it is always possible to transform (in a unique way) a set of patterns $\mathcal{P}$ into a set of unitary patterns $\mathcal{P}^*$, called the *decomposition* of $\mathcal{P}$, having the same representation capacity as $\mathcal{P}$. A partial order relationship on sets of patterns is then defined as follows.

Definition 5 *A set of patterns $\mathcal{P}_1$ is* subsumed *by a set of patterns $\mathcal{P}_2$ (in symbol $\mathcal{P}_1 \preceq \mathcal{P}_2$) if for each pattern $P_1 \in \mathcal{P}_1^*$ there is a pattern $P_2 \in \mathcal{P}_2^*$ such that P_1 and P_2 are isomorphic. If both $\mathcal{P}_1 \preceq \mathcal{P}_2$ and $\mathcal{P}_2 \preceq \mathcal{P}_1$ then $\mathcal{P}_1$ and $\mathcal{P}_2$ are* equivalent *(in symbols $\mathcal{P}_1 \sim \mathcal{P}_2$).*

Given a pair of sets of patterns $\mathcal{P}_1$ and $\mathcal{P}_2$, the subsumption relationship yields a practical way for testing whether $\mathcal{P}_1$ has more representation capacity than $\mathcal{P}_2$ (that is, whether $\mathcal{P}_1$ is able to represent all the structures represented by $\mathcal{P}_2$ and possibly more). This is confirmed by the following result.

Theorem 1 [Atzeni and Torlone (1994)] *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two sets of patterns. Then $\mathcal{P}_1 \preceq \mathcal{P}_2$ if and only if $\mathcal{P}_1 \subseteq \mathcal{P}_2$.*

We now define a number of binary operations on sets of patterns, based on the notion of decomposition.

- The *join* of a pair of sets of patterns $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \sqcup \mathcal{P}_2$, is the set of patterns $\mathcal{P}_1^* \cup \mathcal{P}_2^*$.
- The *meet* of a pair of sets of patterns $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \sqcap \mathcal{P}_2$, is the set of patterns $\mathcal{P}_1^* \cap \mathcal{P}_2^*$.
- The *difference* of a pair of sets of patterns $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 - \mathcal{P}_2$, is the set of patterns $\mathcal{P}_1^* - \mathcal{P}_2^*$.

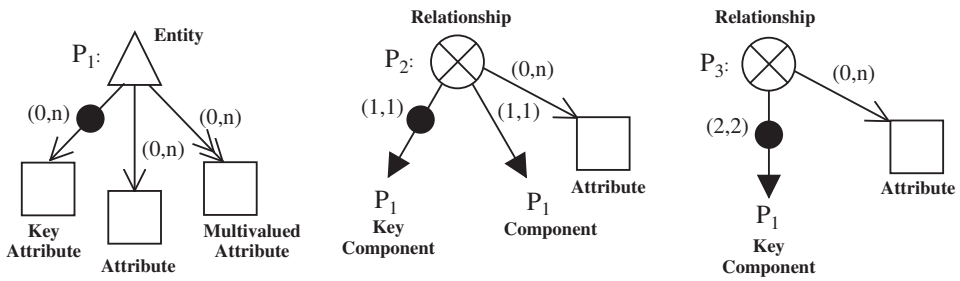

Figure 2 A set of patterns describing a version of the E-R model.

Lemma 1 *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be a pair of sets of patterns. Then: (1) $Inst(\mathcal{P}_1 \sqcup \mathcal{P}_2) = Inst(\mathcal{P}_1) \cup Inst(\mathcal{P}_2)$, (2) $Inst(\mathcal{P}_1 \sqcap \mathcal{P}_2) = Inst(\mathcal{P}_1) \cap Inst(\mathcal{P}_2)$, and (3) $Inst(\mathcal{P}_1 - \mathcal{P}_2) = Inst(\mathcal{P}_1) - Inst(\mathcal{P}_2)$.*

The join and the meet operators are both commutative and associative and so we can speak of join and meet of a finite *collection* of sets of patterns. Then, it is possible to show that the partial order relation $\preceq$ induces a *lattice* on the set of sets of (unitary) patterns.

Theorem 2 [Atzeni and Torlone (1994)] *Every finite collection* $\mathbf{P}$ *of sets of patterns has both a* greatest lower bound ($glb(\mathbf{P})$) *equal to the meet of the elements in* $\mathbf{P}$*, and a* least upper bound ($lub(\mathbf{P})$) *equal to the join of the elements in* $\mathbf{P}$.

3 MODELS, SCHEMES AND TRANSLATIONS

In this section we show how structures and patterns can be used to represent the various components of our framework.

3.1 Models and Schemes

Definition 6 *A* model *is a pair $M = (\mathcal{P}, \gamma)$ where $\mathcal{P}$ is a set of patterns and γ is a labeling function that maps each element of $\mathcal{N} \cup \mathcal{E}$ occurring in $\mathcal{P}$ to a label.*

The labels corresponds to the names associated with to a construct in a specific model (e.g., the abstract construct is called *entity* in the E-R model).

Example 2 *In Figures 2 and 3, we show two sets of patterns representing two different data models (the symbol n denotes a parameter representing a fixed integer). Specifically, the patterns in Figure 2 represent a version of the E-R model involving binary relationships with entities that can have simple and/or multivalued attributes. Labels have been assigned to the constructs of the model, according to a quite standard terminology. Note that, for the sake of conciseness, the identifier of the pattern P_1 is used in patterns P_2 and P_3 to denote their subpatterns. The patterns in Figure 3 represent another version of the E-R*

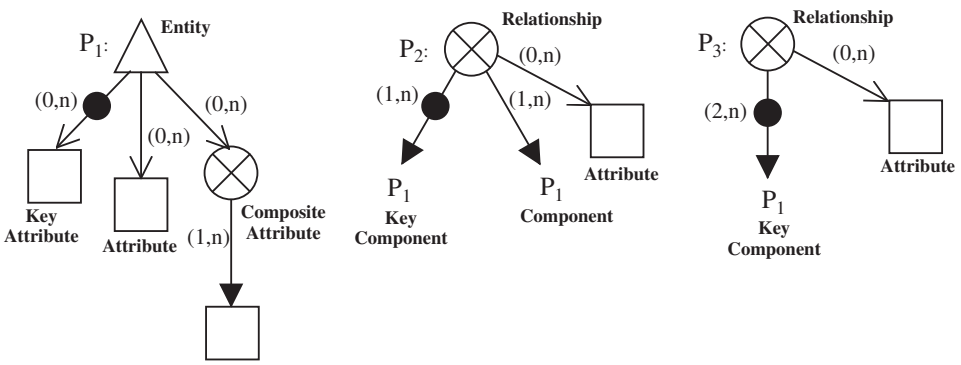

Figure 3 A set of patterns describing another version of the E-R model.

model involving n-ary relationships with entities which can have simple and composite attributes (that is, attributes that are aggregations of lexicals).

Using the parenthetical notation, the two sets of patterns can be represented as follows:

$$\mathcal{P}_1 = \{\, P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\langle\Box\rangle\rangle_{(0,n)}),\ P_2 : \otimes([P_1]^k_{(1,1)}[P_1]_{(1,1)}\langle\Box\rangle_{(0,n)}),\ P_3 : \otimes([P_1]^k_{(2,2)}\langle\Box\rangle_{(0,n)})\ \}$$

$$\mathcal{P}_2 = \{\, P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\otimes([\Box]_{(1,n)})\rangle_{(0,n)}),\ P_2 : \otimes([P_1]^k_{(1,n)}[P_1]_{(1,n)}\langle\Box\rangle_{(0,n)}),\ P_3 : \otimes([P_1]^k_{(2,n)}\langle\Box\rangle_{(0,n)})\ \}$$

Definition 7 *A* scheme *is a pair* $\mathbf{S} = (S, \lambda)$ *composed by a structure* S *and by a* labeling function λ *that maps each node and each edge of* S *to a label, such that different labels are associated with different elements of the structure.*

The labels used in a scheme correspond to names associated with the various concepts in a specific scheme (e.g., persons, books and so on). It is important to note that in our approach the definition of scheme is completely independent on the notion of model. Clearly, it is possible to establish a correspondence between schemes and models as follows.

Definition 8 *A scheme* $\mathbf{S} = (S, \lambda)$ *is* allowed *in a model* $M = (\mathcal{P}, \gamma)$ *if* $S \in Inst(\mathcal{P})$.

Example 3 *In Figure 4, we report a scheme, representing a real-word situation, that is allowed in the model described by the patterns in Figure 3.*

Let $\mathbf{S} = (S, \lambda)$ be a scheme and $M = (\mathcal{P}, \gamma)$ be a model: it is possible to show that, on the basis of the subsumption relationship, we can easily check whether $\mathbf{S}$ is allowed in M (Atzeni and Torlone (1994)).

3.2 Schema translations

We say that a *translation function* τ from a model M_1 to a model M_2 is a function $\tau : Inst(M_1) \rightarrow Inst(M_2)$. In our framework, translation functions can be specified by means of a specific language (a good candidate is the graph based database language

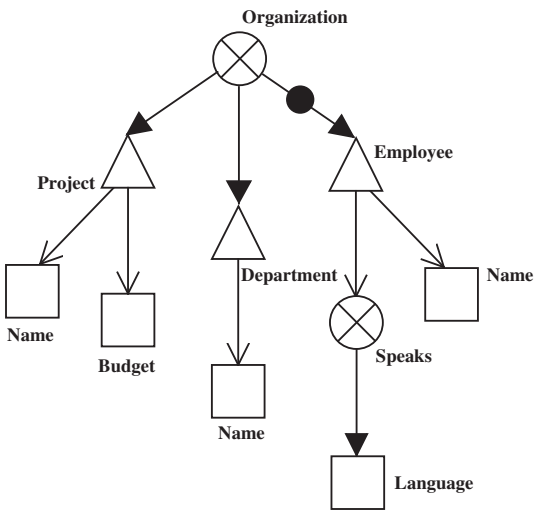

Figure 4 A scheme allowed in the model described in Figure 3.

introduced by Gyssen et al. (1990)). However, since the goal of this paper is to study general properties of translations between models, we need an higher level description of the behavior of a translation function. Again, this will be done by using the notion of pattern.

Definition 9 *A* translation signature σ *for a translation function* τ *is a pair of patterns* $\sigma = (P_1, P_2)$ *such that for each structure* S *and for each component* S_i *of* S*: (1)* $\tau(S_i)$ *is an instance of* $\{P_2\}$ *if* S_i *matches with* P_1*, and (2)* $\tau(S_i) = S_i$ *otherwise.*

In plain words, a translation signature represents: (1) the constructs on which τ operates and (2) the constructs it introduces as effect of its execution. For instance, let us consider the translation signature:

$$\sigma = (\triangle(\langle\Box\rangle^k_{(0,10)}\langle\Box\rangle_{(0,10)}), \otimes([\Box]^k_{(0,10)}[\Box]_{(1,10)}))$$

It represents a translation that replaces abstracts and (optional) functions from abstracts to lexical (e.g., an entity of the E-R model with its attributes), with an aggregation on lexicals (e.g., a relation of the relational model). It is important to note that a translation signature is independent of a specific model.

Definition 10 *A* translation rule *has the form* $\sigma[\tau]$*, where* τ *is a translation function and* σ *is a translation signature for* τ*. A* translation T *is a sequence of translation rules* $T = \sigma_1[\tau_1], \ldots, \sigma_k[\tau_k]$.

A nice property of a translation signature is that it can be used to characterize translations in terms of models. Let $\mathcal{P}$ be a set of patterns and $\sigma[\tau]$ be a translation rule where $\sigma = (P_1, P_2)$. Then, the *effect* of σ on $\mathcal{P}$, denoted by $\sigma(\mathcal{P})$, is defined as follows:

$$\sigma(\mathcal{P}) = \begin{cases} (\mathcal{P} - \{P_1\}) \sqcup \{P_2\} & \textit{if } \{P_1\} \preceq \mathcal{P} \\ \mathcal{P} & \textit{otherwise} \end{cases}$$

We have the following result.

Theorem 3 *Let $\sigma[\tau]$ be a translation rule, $\mathcal{P}$ a set of patterns and S a structure such that $S \in Inst(\mathcal{P})$. Then, it is the case that $\tau(S) \in Inst(\sigma(\mathcal{P}))$.*

The *signature* σ_T of a translation $T = \sigma_1[\tau_1], \ldots, \sigma_k[\tau_k]$ is the sequence of signatures $\sigma_T = \sigma_1, \ldots, \sigma_k$, whereas the *body* τ_T of T is the composition of the translation functions of T: $\tau_T = \tau_k \circ \ldots \circ \tau_1$. Then, the effect of the signature $\sigma_T = \sigma_1, \ldots, \sigma_k$ over a set of patterns $\mathcal{P}$ is the set of patterns $\sigma_k(\ldots\sigma_1(\mathcal{P})\ldots)$. It turns out that the results above can be easily extended to entire translations.

Corollary 1 *Let T be a translation (with signature σ_T and body τ_T), $\mathcal{P}$ be a set of patterns and S a structure such that $S \in Inst(\mathcal{P})$. Then, it is the case that $\tau_T(S) \in Inst(\sigma_T(\mathcal{P}))$.*

4 PROPERTIES OF SCHEMA TRANSLATIONS

According to our approach, we fix a set $\mathcal{R}_b$ of basic translation rules and we assume hereinafter that a translation T is *based on* $\mathcal{R}_b$, that is, a sequence of translation rules in $\mathcal{R}_b$. We recall that these basic translations implement the standard translations between the constructs present in the traditional data models (e.g., from an entity of the Entity-Relationship model to a relation of the relational model or from a n-ary relation to a set of binary ones).

4.1 Correctness of translations

We first introduce a very general and natural notion of correctness of translations: we say that a translation T is a *correct translation from a source model M_s to a target model M_t* if τ_T is a translation function from M_s to M_t. By the results of the previous section it follows that this property can be easily verified by means of the translation signatures.

Lemma 2 *A translation T is a correct translation from a source model $M_s = (\mathcal{P}_s, \gamma_s)$ to a target model $M_t = (\mathcal{P}_t, \gamma_t)$ if and only if $\sigma_T(\mathcal{P}_s) \preceq \mathcal{P}_t$.*

The following important result follows by the lattice structure on the set of patterns.

Theorem 4 *Let $\mathcal{M} = \{M_1, \ldots, M_k\}$ be a set of models $M_i = (\mathcal{P}_i, \gamma_i)$ $(i = 1, \ldots, k)$, $\mathcal{P}_\top$ be the least upper bound of $\{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ and $M_t = (\mathcal{P}_t, \gamma_t)$ be a model such that $\mathcal{P}_t \preceq \mathcal{P}_\top$. Then a translation T such that $\sigma_T(\mathcal{P}_\top) \preceq \mathcal{P}_t$ is a correct translation from any model $M_i \in \mathcal{M}$ to M_t.*

An important consequence of the above result is that in the lattice framework, given a set of models $\mathcal{M}$, there is no need to specify a translation for each pair of models in $\mathcal{M}$ as it is sufficient to look for translations from $\mathcal{P}_\top$ to every model. Thus, the number of required translations is linear in terms of the number of involved models, rather than quadratic. It could be said that the set of patterns $\mathcal{P}_\top$ represents a "supermodel" that contains all the possible constructs used in the various models.

4.2 Completeness of sets of translation rules

Definition 11 *A set of basic translation rules $\mathcal{R}$ is* complete *with respect to a set of models $\mathcal{M}$ if for every pair of models M_s and M_t in $\mathcal{M}$ there is a correct translation T based on $\mathcal{R}$ from M_s to M_t.*

By the results on correctness, we can verify this property as follows. We say that a model $M_m = (\mathcal{P}_m, \gamma) \in \mathcal{M}$ is *minimal* in $\mathcal{M}$ if there is no model $M = (\mathcal{P}, \gamma) \in \mathcal{M}$, different from M_m, such that $\mathcal{P} \preceq \mathcal{P}_m$.

Theorem 5 *A set of translation rules $\mathcal{R}$ is* complete *with respect to a set of models $\mathcal{M}$ if and only if for each minimal model $M_m = (\mathcal{P}_m, \gamma) \in \mathcal{M}$ there is a translation T such that $\sigma_T(\mathcal{P}_\top) \preceq \mathcal{P}_m$.*

The importance of this result relies also on the fact that it suggests a methodology to achieve completeness. If a set of translation rules turns out to be incomplete because we are not able to find a translation from $\mathcal{P}_\top$ (the lub of our models) to a certain minimal model, then we can add new rules to guarantee that this translation can be done.

From a practical point of view, the supermodel and the minimal models state a set of maximal and minimal conditions respectively that all the models must satisfy to be described and managed within the framework. This is reasonable since in general not all possible combinations of constructs form a model.

4.3 Comparisons between translations

The various characterizations on correctness of translations give rise to a natural measure of the quality of a translation from one model to another. We have observed (1993) that the structures we should obtain as a result of a "good" translation have to exploit as much as possible the constructs of the target model. This notion can be formalized as follows.

Definition 12 *Given two different correct translations T_1 and T_2 from a source model $M_s = (\mathcal{P}_s, \gamma_s)$ to a generic target model, T_1* is preferable *than T_2 if $\sigma_{T_2}(\mathcal{P}_s) \preceq \sigma_{T_1}(\mathcal{P}_s)$.*

Again, preferability of translations can be characterized using the notion of subsumption.

Lemma 3 *Let T_1 and T_2 be two different correct translations from a source model $M_s = (\mathcal{P}_s, \gamma_s)$ to a target model $M_t = (\mathcal{P}_t, \gamma_t)$. Then, T_1 is preferable to T_2 if and only if $\sigma_{T_2}(\mathcal{P}_s) \preceq \sigma_{T_1}(\mathcal{P}_s)$.*

We introduce now two reasonable types of "optimization" criteria for translations. The former is related to syntactic aspects (that is, length of the translation), whereas the latter is more involved and takes into account the above notion of preferability. Let us denote with $|T|$ the number of basic translations rules occurring in T.

Definition 13 *A correct translation T from a source model M_s to a target model M_t is* minimal *if there is no rule $R \in T$ such that $T - \{R\}$ is preferable than T.*

Definition 14 *A correct translation T from a source model M_s to a target model M_t is* optimal *if there is no other correct translation T' from M_s to M_t, such that T' is preferable than T.*

5 AUTOMATIC GENERATION OF TRANSLATIONS

In this section we present a number of results that can be used for deriving correct and (possibly) optimal translations between models.

5.1 Reductions and monotonic translations

In searching for translations between sets of patterns (and so, between models), there is an important point to take into account: in the lattice framework we have defined, it is sufficient to search for translation between sets of patterns $\mathcal{P}_t$ and $\mathcal{P}_s$ such that $\mathcal{P}_t \preceq \mathcal{P}_s$. In fact, by Theorem 4, it follows that for any pair of models (or, more generally, for any set of models) we can look for a translation from their least upper bound that, by definition, subsumes them. We will call a translation with this property a *reduction*. More specifically, we say that a translation T is a reduction with respect to a model $M = (\mathcal{P}, \gamma)$ if $\sigma_T(\mathcal{P}) \preceq \mathcal{P}$.

A very general method for generating a reduction from $\mathcal{P}_s$ to $\mathcal{P}_t$ consists in selecting rules that eliminate patterns of $\mathcal{P}_t$ which are not allowed in $\mathcal{P}_s$. Unfortunately, this cannot be done naively since the order in which the rule are selected is crucial. In fact, it may happen that a rule that eliminates a certain pattern P is selected before a rule that eliminates another pattern but, as a side effect, introduces P again. This can be avoided by looking for reductions that enjoy the following property.

Let $T = R_1, \ldots, R_k$ be a reduction with respect to a model $M = (\mathcal{P}, \gamma)$ and let $T|_j$ denote the reduction $R_1, \ldots, R_j$ composed by the first j rules of T ($j \leq k$). Then, we say that T is *monotonic* with respect to M if, for every $1 \leq i < j \leq k$, it is the case that:

$$(\mathcal{P} - \sigma_{T|_i}(\mathcal{P})) \sqcap \sigma_{T|_j}(\mathcal{P}) \neq \emptyset.$$

We now show that this property can be verified locally, by analyzing the set of rule at disposal. Given a set of rules $\mathcal{R}$, the analysis needs the construction of a graph $G_{\mathcal{R}}$, called the *precedence graph* of $\mathcal{R}$, whose nodes represent the rules in $\mathcal{R}$ and such that there is an edge from a rule $R_i = \sigma_i[\tau_i]$ to a rule $R_j = \sigma_j[\tau_j]$ if $\sigma_i = (P_1^i, P_2^i)$, $\sigma_j = (P_1^j, P_2^j)$ and $P_2^i \sqcap P_1^j \neq \emptyset$. Intuitively, the presence of an edge from R_i to R_j means that R_i must be executed before R_j. For reductions, the following result holds.

Lemma 4 *Let T be a reduction with respect to a model M involving the rules $\mathcal{R}$. Then, if for each pair of rules $\sigma_i[\tau_i]$, $\sigma_j[\tau_j]$ in T such that $i < j$, $\sigma_i[\tau_i]$ is not reachable from $\sigma_j[\tau_j]$ in $G_{\mathcal{R}}$, then T is monotonic with respect to M.*

Hence, after the selection of a set of rules that implements our reduction, we need to serialize them according to the above property. We say that a translation that satisfies the condition of Lemma 4 is *serial* and that a set of rules $\mathcal{R}$ is *serializable* if it is possible to find a serial translation that involves all the rules in $\mathcal{R}$. The following result easily follows.

Lemma 5 *A set of translation rules $\mathcal{R}$ is serializable if and only if $G_{\mathcal{R}}$ is acyclic.*

We present now an interesting result for serializable rules $\mathcal{R}$ which turns out to be very useful in the following: it states that the effect of a serial reduction based on $\mathcal{R}$ (1) is easy to compute and (2) is essentially independent of the serialization chosen. Given a set of rules $\mathcal{R}$ and a set of patterns $\mathcal{P}$ let $Del(\mathcal{R}, \mathcal{P}) = \sqcup_{\forall \sigma[\tau] \in \mathcal{R}}(\mathcal{P} - \sigma(\mathcal{P}))$.

Theorem 6 *Let $\mathcal{P}$ be a set of patterns and $\mathcal{R}$ be a set of serializable rules. Then, $\sigma_T(\mathcal{P}) = \mathcal{P} - Del(\mathcal{R}, \mathcal{P})$ for any serial reduction based on $\mathcal{R}$.*

5.2 A method for generating translations

On the basis of the above results, we can derive a methodology for the automatic generation of correct translations between models. The methodology is composed in the following steps:

1. Consider a source set of patterns $\mathcal{P}_s$ and a target set of patterns $\mathcal{P}_t$: as explained above, we can assume, without loss of generality, that $\mathcal{P}_t \preceq \mathcal{P}_s$. Then, in order to derive a reduction from $\mathcal{P}_s$ to $\mathcal{P}_t$, we first select rules whose effect deletes patterns that belong to $\mathcal{P}_s$ but do not belong to $\mathcal{P}_t$. If the effect of the obtained set of rules does not allow to generate the target model, we stop here.
2. In the second part, the set of rules $\mathcal{R}_1$ generated in the first part is (possibly) reduced by deleting the rule $R \in \mathcal{R}_1$ that are "redundant", that is, whose elimination does not affect the correctness of the translation.
3. Then, in the third part we verify whether the set of rules $\mathcal{R}_2$ obtained in the second step is serializable. If they are not, we can try to derive another reduction over the set of rules $\mathcal{R}_b - \{R\}$, where R is a rule in $\mathcal{R}_2$ that belongs to a cycle in $G_{\mathcal{R}_2}$ and so, by Lemma 5, causes the rule set to be non-serializable. If the remaining rule set does not yield a correct translation, we stop here.
4. Finally, we can try to derive an optimal rule set. This can be done by searching for a serializable reduction over the set of rules $\mathcal{R}_b - \{R\}$, where R is a rule that belongs to the set obtained in the previous step and deletes patterns that are in $\mathcal{P}_t$. The rationale under this choice is that there could be "finer" functions which are able to replace the work done by R and that do not require the deletion of patterns in the target set of patterns.

In general, according to the results presented in Section 4, we can find correct translations between a generic set of models $\mathcal{M} = \{M_1, \ldots, M_k\}$ such that $M_i = (\mathcal{P}_i, \gamma_i)$, for $i = 1, \ldots, k$, by first finding the lub $\mathcal{P}_\top$ of $\mathcal{P}_1, \ldots, \mathcal{P}_k$, and then applying the methodology above to $\mathcal{P}_\top$ and, in turn, $\mathcal{P}_1, \ldots, \mathcal{P}_k$.

6 A SIMPLE CASE STUDY

In this section we present a concrete example to illustrate the various notions introduced in the paper. Let us consider the following data models:

- M_1 is a version of the E-R model with binary relationships and multivalued attributes for entities (see Figure 2):

  $P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\langle\Box\rangle\rangle_{(0,n)})$
  $P_2 : \otimes([P_1]^k_{(1,1)}[P_1]_{(1,1)}\langle\Box\rangle_{(0,n)})$
  $P_3 : \otimes([P_1]^k_{(2,2)}\langle\Box\rangle_{(0,n)})$

- M_2 is a version of the E-R model with n-ary relationships and composite attributes for entities (see Figure 3):

  $P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\otimes([\Box]_{(1,n)})\rangle_{(0,n)})$
  $P_2 : \otimes([P_1]^k_{(1,n)}[P_1]_{(1,n)}\langle\Box\rangle_{(0,n)})$
  $P_3 : \otimes([P_1]^k_{(2,n)}\langle\Box\rangle_{(0,n)})$

- M_3 is a version of the E-R model with n-ary relationships and subset relations between entities:

  $P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})$
  $P_2 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}[\![P_1]\!]_{(0,n)})$
  $P_3 : \otimes([P_2]^k_{(1,n)}[P_2]_{(0,n)}\langle\Box\rangle_{(0,n)})$
  $P_4 : \otimes([P_2]_{(1,n)}\langle\Box\rangle_{(0,n)})$

- M_3 is the relational model:

  $P_1 : \otimes([\Box]^k_{(1,n)}[\Box]_{(0,n)})$

The lub of the patterns describing the above models is then as follows:

$$\begin{aligned}\mathcal{P}_\top = \{\quad & P_1 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\langle\Box\rangle\rangle_{(0,n)}\langle\otimes([\Box]_{(1,n)})\rangle_{(0,n)})\\ & P_2 : \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}\langle\langle\Box\rangle\rangle_{(0,n)}\langle\otimes([\Box]_{(1,n)})\rangle_{(0,n)}[\![P_1]\!]_{(0,n)})\\ & P_3 : \otimes([P_2]^k_{(1,n)}[P_2]_{(1,n)}\langle\Box\rangle_{(0,n)})\\ & P_4 : \otimes([P_2]^k_{(1,n)}\langle\Box\rangle_{(0,n)})\\ & P_5 : \otimes([\Box]^k_{(1,n)}[\Box]_{(0,n)})\quad\}\end{aligned}$$

Now, let us consider a set of translation rules $\mathcal{R}_b$ described by the following translation signatures:

- R_1 translates abstracts (with simple attributes) into aggregations on lexicals:

$\sigma_1 = (\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}), \otimes([\Box]^k_{(1,n)}[\Box]_{(0,n)}))$

- R_2 translates aggregations on lexicals into abstracts (with simple attributes):

  $\sigma_2 = (\otimes([\Box]^k_{(1,n)}[\Box]_{(0,n)}), \triangle(\langle\Box\rangle^k_{(1,n)}\langle\Box\rangle_{(0,n)}))$

- R_3 translates aggregations on abstracts (with simple attributes) into abstracts (with simple attributes):

  $\sigma_3 = (\otimes([\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})]^k_{(1,n)}[\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})]_{(1,n)}), \triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}))$

- R_4 translates n-ary aggregations on abstracts (with simple attributes) into binary aggregations on abstracts (with simple attributes):

  $\sigma_5 = (\otimes([\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})]^k_{(1,n)}[\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})]_{(0,n)}),$
  $\otimes([\triangle(\langle\Box\rangle^k_{(1,1)}\langle\Box\rangle_{(0,n)})]^k_{(1,1)}[\triangle(\langle\Box\rangle^k_{(1,1)}\langle\Box\rangle_{(0,n)})]_{(1,1)}))$

- R_5 translates abstracts with composite attributes into binary aggregations on abstracts (with simple attributes):

  $\sigma_7 = (\triangle(\langle\otimes([\Box]_{(1,n)})\rangle_{(1,n)}), \otimes([\triangle(\langle\Box\rangle^k_{(1,1)}\langle\Box\rangle_{(0,n)})]^k_{(1,1)}[\triangle(\langle\Box\rangle^k_{(1,1)}\langle\Box\rangle_{(0,n)})]^k_{(1,1)}))$

- R_6 translates abstracts with multivalued attributes into abstracts with composite attributes:

  $\sigma_8 = (\triangle(\langle\!\langle\Box\rangle\!\rangle_{(1,n)}), \triangle(\langle\otimes([\Box]_{(1,n)})\rangle_{(1,n)}))$

- R_7 translates subset relations between abstracts into binary aggregations on abstracts:

  $\sigma_9 = (\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}[\![\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)}]\!]_{(1,n)}), \otimes([\triangle(\langle\Box\rangle^k_{(0,n)}\langle\Box\rangle_{(0,n)})]^k_{(2,2)}))$

In this context, the precedence graph G_P has the rules in $\mathcal{R}_b$ as nodes and the following edges:

$(R_1, R_2), (R_2, R_1), (R_3, R_1), (R_4, R_3), (R_5, R_3), (R_6, R_5), (R_7, R_3), (R_7, R_4)$.

Then, by applying the first step of the methodology described in Section 5, we can generate the following sets of rules:

- From $\mathcal{P}_\top$ to $\mathcal{P}_1$: we obtain the set $\mathcal{R}_1 = R_2; R_4; R_5; R_7$.
- From $\mathcal{P}_\top$ to $\mathcal{P}_2$: we obtain the set $\mathcal{R}_2 = R_2; R_6; R_7$.
- From $\mathcal{P}_\top$ to $\mathcal{P}_3$: we obtain the set $\mathcal{R}_3 = R_2; R_5; R_6$.
- From $\mathcal{P}_\top$ to $\mathcal{P}_4$: we obtain the set $\mathcal{R}_4 = R_1; R_3; R_5; R_6; R_7$.

Indeed, all of these rule sets can be serialized and therefore the following translations can be finally obtained:

- $T_{M_1} = R_2; R_7; R_4; R_5$.
- $T_{M_2} = R_2; R_6; R_7$.
- $T_{M_3} = R_2; R_6; R_5$.
- $T_{M_4} = R_6; R_5; R_7; R_3; R_1$.

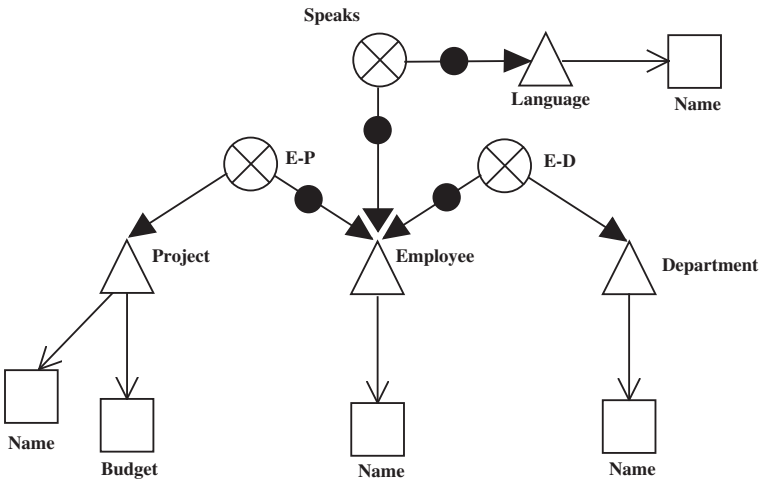

Figure 5 The result of the translation of the scheme in Figure 4 to the model M_1.

Now, if we want to translate the scheme S reported in Figure 4 (that is allowed in the model M_2) to the model M_1, we can use the translation T_{M_1}. Actually, the first two rules of T_{M_1} (namely, R_2 and R_7) do not produce any effect on the input scheme, since both of them apply to combination of constructs (aggregations on lexicals and subset relations between abstracts, respectively) that do not occur in S. Rule R_5 instead replaces the ternary relationship in scheme S with two binary relationships, in a standard way. Finally, rule R_5 replaces the composite attribute of the entity **Employee** with a relationship between this entity and the new entity **Language**. The final scheme is reported in Figure 5. It is easy to see that this scheme is indeed allowed in the model M_1.

7 CONCLUSIONS

In this paper we have presented a formal approach to the problem of translating schemes of different data models. We have introduced the notion of pattern as a graph-theoretic tool for the description of models and we have defined a partial order relationship among sets of patterns that induces a lattice on them. This lattice structure allows us to compare different data models and to define and characterize various interesting properties of translations. On the basis of these results, we have then developed a general methodology for deriving correct translations between models that enjoy the above properties.

We believe that this approach brings a contribution to several problems related to cooperative activities within heterogeneous database frameworks, and is promising for further investigations. From a theoretical point of view, we are currently working on extending the results of this paper to more general cases and testing the various features of the approach in an involved case. From a practical point of view we are working on the development of a first prototype of the metamodel.

REFERENCES

S. Abiteboul and R. Hull. Restructuring hierarchical database objects. *Theoretical Computer Science*, 62(3):3–38, 1988.

P. Atzeni, G. Ausiello, C. Batini, and M. Moscarini. Inclusion and equivalence between relational database schemata. *Theoretical Computer Science*, 19(2):267–285, 1982.

P. Atzeni and R. Torlone. A metamodel approach for the management of multiple models and the translation of schemes. *Information Systems*, 18(6):349–362, 1993.

P. Atzeni and R. Torlone. Management and translation of heterogeneous database schemes. *Second International East-West Database Workshop, Klagenfurt, 1994*, Workshops in Computing, Springer–Verlag. To appear.

T. Barsalou and D. Gangopadhyay. M(DM): An open framework for interoperation of multimodel multidatabase systems. In *International Conference on Data Engineering*, pages 218–227, 1992.

C. Batini, S. Ceri, and S.B. Navathe. *Conceptual Database Design, an Entity-Relationship Approach.* Benjamin and Cummings Publ. Co., Menlo Park, California, 1992.

P.P. Chen. The entity-relationship model: toward a unified view of data. *ACM Trans. on Database Syst.*, 1(1):9–36, March 1976.

M. Gyssens, J. Paredaens, and D. Van Gucht. A graph-oriented object database model. In *Ninth ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems*, pages 417–424, 1990.

R.B. Hull. Relative information capacity of simple relational schemata. *SIAM Journal on Computing*, 15(3):856–886, 1986.

R.B. Hull and R. King. Semantic database modelling: survey, applications and research issues. *ACM Computing Surveys*, 19(3):201–260, September 1987.

L.A. Kalinichenko. Methods and tools for equivalent data model mapping construction. In *International Conf. on Extending Database Technology (EDBT'90), Lecture Notes in Computer Science 416*, pages 92–119, Springer-Verlag, 1990.

G.M. Kuper. *The Logical Data Model: A New Approach to Database Logic.* PhD thesis, Stanford University, 1985.

Y.E. Lien. On the equivalence of database models. *Journal of the ACM*, 29(2):333–362, 1982.

R.J. Miller, Y.E. Ioannidis, and R. Ramakrishnan. The use of information capacity in schema integration and translation. In *Eighteenth International Conf. on Very Large Data Bases*, pages 120–133, 1993.

J. Rissanen. On equivalence of database schemes. In *ACM SIGACT SIGMOD Symp. on Principles of Database Systems*, pages 23–26, 1982.

A.P. Sheth and J.A. Larson. Federated database systems for managing distribuited database systems for production and use. *ACM Computing Surveys*, 22(3):183–236, 1990.

D. Tsichritzis and F.H. Lochovski. *Data Models.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

BIOGRAPHIES

Paolo Atzeni is a professor at Terza Università di Roma. Previously, he was a research associate at IASI-CNR in Rome, an associate professor at Università di Napoli and a professor at Università di Roma "La Sapienza." His research interests concern database theory, deductive and object databases, database design and tools.

Riccardo Torlone is a researcher at Terza Università di Roma. In the past, he was a research associate at IASI-CNR in Rome. His research interests include database theory, database languages, active and deductive databases, database design and tools.