Reference Model and Perspective schemata inference for Enterprise Data Integration

Valéria Magalhães Pequeno[1], João Carlos Moura Pires[1]

[1]CENTRIA, Departamento de Informática,
Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa
2829-516, Caparica, Portugal

vmp@fct.unl.pt, jmp@di.fct.unl.pt

keywords: ETL process, conceptual data model, data integration, correspondence assertions, object-relational database, rewriting-rules

Abstract

One of leading issues in database research is to develop flexible mechanisms for providing integrated access to distributed, heterogeneous databases and other information sources. A wide range of techniques has been developed to address this problem, the main drawback being the difficulty in developing a single (global) database schema that captures all the nuances of diverse data types, and expresses a unified view of the enterprise. We deal with this problem by taking a declarative approach, which is based on the creation of a reference model and perspective schemata. The former provides a common semantic, while the latter connects schemata. This paper focus on deduction of new perspective schemata using a proposed inference mechanism.

1. Introduction

One of the leading issues in database research is to develop flexible mechanisms for providing integrated access to multiple, distributed, heterogeneous databases and other information sources. A wide range of techniques has been developed to address this problem, including approaches based on creation of Data Warehouses (DWs), and Federated Database Systems (FDBs). DWs are highly specialised database systems which contain the unified history of an enterprise at a suitable level of detail for decision support. All data are integrated into, usually, a single repository, with a generalised and global schema. A FDB enables a unified virtual view of one or more autonomous sources of information to hide data heterogeneity from the applications and users. Tightly coupled FDB, those that occur in DWs, provides a global schema expressed in a common, ”canonical” data model. Unlike a DW, a FDB leaves data at the source.

One of the main drawbacks of these approaches is the difficulty in developing a single (global or common) database schema that captures all the nuances of diverse data types, and expresses a unified view of the enterprise. The designer should usually deal with incompatible data models, characterised by subtle differences in structure and semantic. Besides, he/she should define mappings between the global schema and the schemata of the source information. These problems are hardest to deal with because of the rapid growing of the data volume and the data model complexity (both in sources and in global schema), which implies the rise of the difficulty of managing and understanding these models [1].

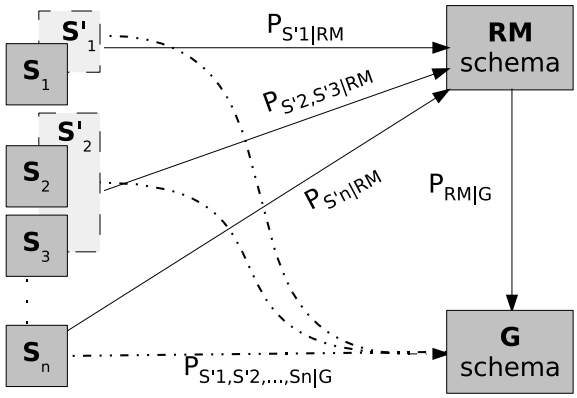

Figure 1: Proposed architecture.

In order to deal with this problem, it is proposed to take a declarative approach, based on the creation of a reference model and perspective schemata. A Reference Model is an abstract framework that provides a common semantic that can be used to guide the development of other models and help with data consistency [1]. A perspective schema describes a data model, part or whole (target schema), in terms of other data models (base schemata). In Fig. 1, for instance, $\mathsf{P}_{s'1|RM}$, ..., $\mathbf{P}_{s'n|RM}$ are perspective schemata that map the reference model (**RM**) in terms of the source schemata.

In the proposed approach, the relationship between the base schemata and the target schema is made explicitly and declaratively through correspondence assertions. An advantage of the proposed approach is that by using the reference model the designer does not need to map schemata each other. This effort is theoretically reduced since schemata (source or global) must only align with the reference model, rather than each participating schema. Besides, the designer does not need to have a deep knowledge of all schemata involved in the DW system or in the federation system. Thus, the designer can describe the global system without concerns about where the sources are or how they are stored. Furthermore, the mapping between the global schema and its sources is automatically generated by an inference mechanism. This paper focuses on the deduction of new perspective schemata using a proposed inference mechanism.

The remainder of this paper is laid out as follows. Section 2 presents an overview of the reference model-based framework proposed in [2]. Section 3 briefly describes the language to define the perspective schemata. Section 4 details the process to infer new perspective schemata. Section 5 concisely mentions representative works in data integration area. The paper ends with Section 6, which points out the new features of the approach presented here and in ongoing or planned future work on this topic.

2. The framework

The proposal presented in [2] offers a way to express the existing data models (source, reference model, and global/integrated schema) and the relationship between them. The approach is based on *Schema language* (L_S) and *Perspective schema language* (L_{PS}).

Schema language (L_S) is used to describe the actual data models (source, reference model, and global/integrated schema). The formal framework focuses on an object-relational paradigm, which includes definitions adopted by the main concepts of object and relational models as they are widely ac-

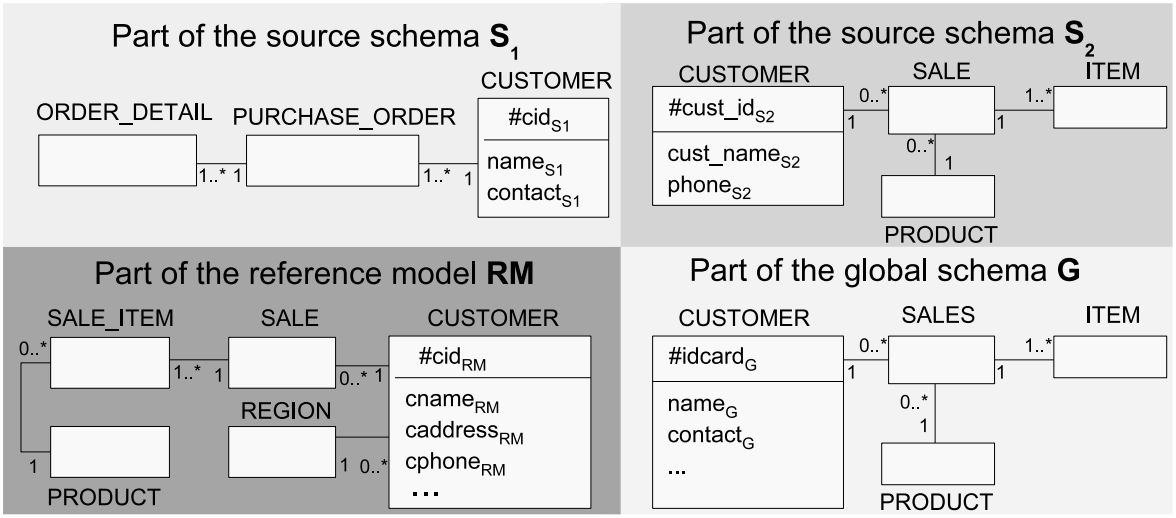

Figure 2: Motivating example.

cepted in literature – cf. [3, 4].

Perspective schema language (L_{PS}) is used to describe *perspective schemata*. A perspective schema is a special kind of schema that describes a data model (part or whole) (*target schema*) in terms of other data models (*base schemata*). L_{PS} mainly extends L_S with two components: Correspondence Assertions (CAs) and Matching Functions (MFs). Correspondence Assertions formally specify the relationship between schema components. Matching functions indicate when two data entities represent the same instance of the real world. L_{PS} includes data transformations, such as names conversion and data types conversion.

Fig. 1 illustrates the basic components of the proposed architecture and their relationships. The schemata **RM**, $\mathbf{S}_1$,...,$\mathbf{S}_n$ and **G** are defined using the language L_S and represent, respectively, the reference model, the source schemata $\mathbf{S}_1$,...,$\mathbf{S}_n$, and a global schema. The schemata $\mathbf{S'}_1$ and $\mathbf{S'}_2$ are defined using the language L_{PS}. They are special kinds of perspective schemata (called *view schema*), since the target schema is described in the scope of a perspective schema, instead of just referring to an existing schema. $\mathbf{S'}_1$ and $\mathbf{S'}_2$ represent, respectively, the view schemata $\mathbf{S'}_1$ (a viewpoint of schema $\mathbf{S}_1$), and $\mathbf{S'}_2$ (an integrated viewpoint of schemata $\mathbf{S}_2$ and $\mathbf{S}_3$). The relationships between the target schema and the base schemata are shown through the perspective schemata $\mathbf{P}_{s'1|RM}$,..., $\mathbf{P}_{s'n|RM}$,$\mathbf{P}_{RM|G}$, and $\mathbf{P}_{s'1,s'2,...,s'n|G}$ (denoted by arrows). In the current research, the perspective schema $\mathbf{P}_{s'1,s'2,...,s'n|G}$ can be automatically deduced by the proposed inference mechanism. The next Section illustrates, through the examples, the language L_{PS}, and the Section 4 presents the proposed inference mechanism. For a more detailed and formal description of L_S and L_{PS} languages, the reader is referred to [5, 6, 2].

3. Perspective Schema Language

The remainder of the paper, considers a simple sales scenario comprising two data sources $\mathbf{S}_1$ and $\mathbf{S}_2$, a reference model **RM**, and a global schema **G**. The schemata are shown in Fig. 2. All properties that are key to a relation (or class) are shown in Fig. 2 using "♯" before their names.

The language L_{PS} is used to define perspective schemata. A perspective schema describes a data

model, part or whole (*target schema*), in terms of other data models (*base schemata*). Usually, a perspective schema is formed by the following components:

1. *Name* is a schema name with the notation: $\mathbf{P_{S|T}}$, being $\mathbf{S}$ the name of one or more base schemata and $\mathbf{T}$ the name of the target schema. In Fig. 2, for instance, $\mathbf{P}_{s'1|RM}$ is a name of a perspective schema whose base schema is S'_1 and the target schema is **RM**;
2. *'Require' declarations* express the subset of the components of the target schema (classes, relations, keys, and foreign keys) that will be necessary in the perspective schema;
3. *Matching Function signatures* indicate which matching functions must be implemented to determine when two objects/tuples are distinct representations of the same object in the real-world;
4. *Correspondence Assertions* establish the semantic correspondence between schemata's components.

The target schema may have much more information than is required to represent in a perspective schema, namely when the target schema is the Reference Model. Hence, it is required to clearly indicate which elements of the target schema are in the scope of the perspective schema. This is done in L_{PS} using 'require' declarations. For instance, consider the perspective schema $\mathbf{P_{S_2|RM}}$ between the schemata **RM** (the target schema) and $\mathbf{S}_2$ (the base schema), both as presented in Fig. 2. For this perspective schema, four relations from **RM** are needed (PRODUCT, CUSTOMER, SALE, and SALE_ITEM). The 'require' declaration to relation CUSTOMER, for example, would be as follows:

$$\text{require}(\text{CUSTOMER}, \{\mathbf{cid}_{\text{RM}}, \mathbf{cname}_{\text{RM}}, \mathbf{cphone}_{\text{RM}}\})$$

Note that, for instance, the properties $\mathbf{cregion\_id}_{\text{RM}}$ and $\mathbf{caddress}_{\text{RM}}$ from **RM**.CUSTOMER are not declared as being required.

3.1. Matching Functions

From a conceptual viewpoint, it is essential to provide a way to identify instances of different schemata that represent the same entity in the real-world. The proposal presented in [2] is to use matching functions, which can include various techniques for matching instances, including some of those used in data cleaning, such as lookup tables, user-defined functions, heuristics and past matching. These functions, as occur in [7], define a 1:1 correspondence between the objects/tuples in families of corresponding classes/relations. In particular, the work shown in [2] is based on the following matching function signature:

$$\mathbf{match} : ((\mathbf{S}_1 [\mathrm{R}_1], \tau_1) \times (\mathbf{S}_2 [\mathrm{R}_2], \tau_2)) \rightarrow \text{Boolean}, \quad (1)$$

being $\mathbf{S}_i$ schema names, R_i class/relation names, and τ_i the data type of the instances of R_i, for i $\in \{1,2\}$. When both arguments are instanced, **match** verifies whether two instances are semantically equivalent or not. If only one argument is instanced, e.g. $\mathbf{S}_1.\mathrm{R}_1$, then it obtains the semantically equivalent $\mathbf{S}_2.\mathrm{R}_2$ instance of the given $\mathbf{S}_1.\mathrm{R}_1$ instance, returning true when it is possible, and false when nothing is found or when there is more than one instance to match.

In some scenarios one-to-many correspondence between instances are common, e.g. when historical data is stored in the DW. In this case, a variant of **match** should be used, which has the following form:

$$\mathbf{match} : ((\mathbf{S}_1 [\mathrm{R}_1], \tau_1) \times (\mathbf{S}_2 [\mathrm{R}_2 (\mathbf{predicate})], \tau_2)) \rightarrow \text{Boolean}. \quad (2)$$

predicate is a boolean condition that determines the context in which the instance matching must be applied in $\mathbf{S}_2.\mathrm{R}_2$.

An example of a matching function signature involving schemata of Fig. 2 is presented in Fig. 3. The implementation of the matching functions shall be externally provided, since their implementation is very close to the application domain. However, in order to make easer the implementation of a simple prototype, a new variants of **match** is introduced in L_{PS}:

$$\textbf{match} : \left(\left(\mathbf{S}_1\left[\mathrm{R}_1\right], \tau_1, \{\boldsymbol{p}'_1 : \tau'_1, ..., \boldsymbol{p}'_n : \tau'_n\}\right) \times \left(\mathbf{S}_2\left[\mathrm{R}_2\right], \tau_2, \{\boldsymbol{p}''_1 : \tau''_1, ..., \boldsymbol{p}''_n : \tau''_n\}\right)\right) \rightarrow \text{Boolean} \quad (3)$$

being that $\boldsymbol{p}'_i{:}\tau'_i \in$ ***type***(R_1); and $\boldsymbol{p}''_i{:}\tau''_i \in$ ***type***(R_2), $1 \leq \mathrm{i} \leq \mathrm{n}$.[1] This variant of the matching function is automatically generated by the system and indicates that the matching is done by simple attribute comparison, i.e. each property $\boldsymbol{p}'_i$ of R_1 will be compared with the property $\boldsymbol{p}''_i$ of R_2, for $1 \leq \mathrm{i} \leq \mathrm{n}$.

$$\textbf{match}{:}((\mathbf{RM}[\text{CUSTOMER}],\tau_1)\times(\mathbf{G}[\text{CUSTOMER}],\tau_2))\rightarrow\text{Boolean}$$

Figure 3: Example of a matching function signature.

3.2. Correspondence Assertions

The semantic correspondence between schemata's components is declared in the proposal presented in [2] through the Correspondence Assertions (CAs), which are used to formally assert the correspondence between schema components in a declarative fashion. CAs are classified in four groups: Property Correspondence Assertion (PCA), Extension Correspondence Assertion (ECA), Summation Correspondence Assertion (SCA), and Aggregation Correspondence Assertion (ACA). Examples of CAs are shown in Fig. 4 and explained in this Section.

Property Correspondence Assertions (PCAs)
ψ_1: $\mathbf{P}_{\mathbf{RM}\mid\mathbf{G}}\left[\text{CUSTOMER}\right] \bullet \mathbf{idcard}_{\mathrm{G}} \rightarrow \text{numberTOtext}\left(\mathbf{RM}\left[\text{CUSTOMER}\right] \bullet \mathbf{cid}_{\mathrm{RM}}\right)$
ψ_2: $\mathbf{P}_{\mathbf{RM}\mid\mathbf{G}}\left[\text{CUSTOMER}\right] \bullet \mathbf{contact}_{\mathrm{G}} \rightarrow \mathbf{RM}\left[\text{CUSTOMER}\right] \bullet \mathbf{cphone}_{\mathrm{RM}}$
Extension Correspondence Assertions (ECAs)
ψ_3: $\mathbf{P}_{\mathbf{RM}\mid\mathbf{G}}\left[\text{CUSTOMER}\right] \rightarrow \mathbf{RM}\left[\text{CUSTOMER}\right]$
ψ_4: $\mathbf{S}_v\left[\text{CUSTOMER}\right] \rightarrow \mathbf{S}_1\left[\text{CUSTOMER}\right] \sqsupset\!\bowtie\!\sqsubset \mathbf{S}_2\left[\text{CUSTOMER}\right]$
Summation Correspondence Assertion (SCA)
ψ_5: $\mathbf{P}_{\mathbf{S}_3\mid\mathbf{RM}}\left[\text{PRODUCT}\right]\left(\mathbf{pid}_{\mathrm{RM}}\right) \rightarrow \text{normalise}\left(\mathbf{S}_3\left[\text{PRODUCT\_SALES}\right]\left(\mathbf{product\_number}_{\mathrm{S3}}\right)\right)$

Figure 4: Examples of correspondence assertion.

Property CAs relate properties of a target schema to the properties of base schemata. They allow dealing with several kinds of semantic heterogeneity such as: *naming conflict* (for instance synonyms and homonyms properties), *data representation conflict* (that occur when similar contents are represented by different data types), and *encoding conflict* (that occur when similar contents are represented by different formats of data or unit of measures). For example, the PCAs ψ_1 and ψ_2 (see Fig. 4) deal with, respectively, *data representation conflict* and *naming conflict*. ψ_1 links the property $\mathbf{idcard}_G$ to

[1] ***type***() is a function defined in language L_S that returns the structural type of a relation/class.

the property **cid**$_{RM}$ using the function **numberTOtext** to convert the data type from *number* to *text*. ψ_2 assigns **contact**$_G$ to **cphone**$_{RM}$.

The Extension CAs are used to describe which objects/tuples of a base schema should have a corresponding semantically equivalent object/tuple in the target schema. For instance, the relation **G**.CUSTOMER is linked to relation **RM**.CUSTOMER through the ECA ψ_3 presented in Fig. 4. ψ_3 determines that **G**.CUSTOMER and **RM**.CUSTOMER are equivalent, i.e., for each tuple of CUSTOMER of the schema **RM** there is one semantically equivalent tuple in CUSTOMER of the schema **G**, and vice-versa.

There are five different kinds of ECAs: equivalence, selection, difference, union, and intersection, being the ECA of union similar to the *natural outer-join* of the usual relational models. For instance, consider the view schema S_v with the relation CUSTOMER, which is related to the relation CUSTOMER of the schema S_1 and to the relation with the same name of S_2 through the ECA ψ_4 shown in Fig. 4. ψ_4 determines that CUSTOMER in S_v is the union/join of CUSTOMER in S_1 and CUSTOMER in S_2, i.e., for each tuple of CUSTOMER of the schema S_1 there is one semantically equivalent tuple in CUSTOMER of the schema S_v, or for each tuple of CUSTOMER of the schema S_2 there is one semantically equivalent tuple in CUSTOMER of the schema S_v, and vice-versa. In an ECA, any relation/class can appear with a selection condition, which determines the subset of instances of the class/relation being considered. This kind of ECA is especially important to the DW because through it the current instances of the DW can be selected and related to the instances of their sources (which usually do not have historical data).

The Summation CAs are used to describe the summary of a class/relation whose instances are related to the instances of another class/relation by breaking them into logical groups that belong together. They are used to indicate that the relationship between classes/relations involve some type of aggregate functions (called SCA of groupby) or a normalisation process (called SCA of normalisation)[2]. For example, consider the source schema **S**$_3$ (not presented in any figure), which contains a denormalised relation PRODUCT_SALES(**product_number**$_{S3}$, **product**$_{S3}$, **quantity**$_{S3}$, **price**$_{S3}$, **purchase_order**$_{S3}$) and the schema **RM** presented in Fig. 2. PRODUCT_SALES holds information about sold items in a purchase order as well as information logically related to products themselves, which could be in another relation, occurring in schema **RM**. The SCA ψ_5, displayed in Fig. 4, determines the relationship between PRODUCT_SALES and **RM**.PRODUCT when a normalisation process is involved, i.e., it determines that **RM**.PRODUCT is a normalisation of **S**$_3$.PRODUCT_SALES based on distinct values of property **product_number**$_{S3}$.

The Aggregation CAs link properties of the target schema to the properties of the base schema when a SCA is used. ACAs associated to SCAs of groupby contains aggregation functions supported by most of the queries languages, like SQL-99 [8], i.e. *summation*, *maximum*, *minimum*, *average* and *count*. The ACAs, similar to the PCAs, allow for the description of several kinds of situations; therefore, the aggregate expressions can be more detailed than simple property references. Calculations performed can include, for example, ordinary functions (such as sum or concatenate two or more properties' values before applying the aggregate function), and Boolean conditions (e.g. count all male students whose grades are greater or equal to 10).

[2]This research also deal with denormalisations, which is defined using *path expressions* (component of the language L_S).

4. Inference Mechanism

This proposal provides an inference mechanism to automatically infer a new perspective schema (see Fig. 5(c)), given:

1. a set of *origin* schemata and their associated perspective schemata, which take the *origin* schemata as *base* and the reference model as *target* (see Fig. 5(a));
2. a *destination* schema and its associated perspective schema, which take the reference model as *base* and the *destination* schema as *target* (see Fig. 5(b)).

In context of the Fig. 1, the perspective schema $\mathbf{P}_{s'1,s'2,...,s'n|G}$ can be inferred taking as *origin* the schemata $\mathbf{S}_1$,...,$\mathbf{S}_n$ as well as the perspective schemata $\mathbf{P}_{s'1|RM}$,..., $\mathbf{P}_{s'n|RM}$, and as *destination* the schema **G** as well as the perspective schema $\mathbf{P}_{RM|G}$.

The inferred perspective schema will have as *base* a subset of *origin* schemata, and as *target* the *destination* schema. Its *'require' declarations* will be the same *'require' declarations* present in the perspective schema associated to the *destination* schema. The *MF signatures* and *CAs* of the inferred perspective schema will be automatically generated using a rule-based rewritten system.

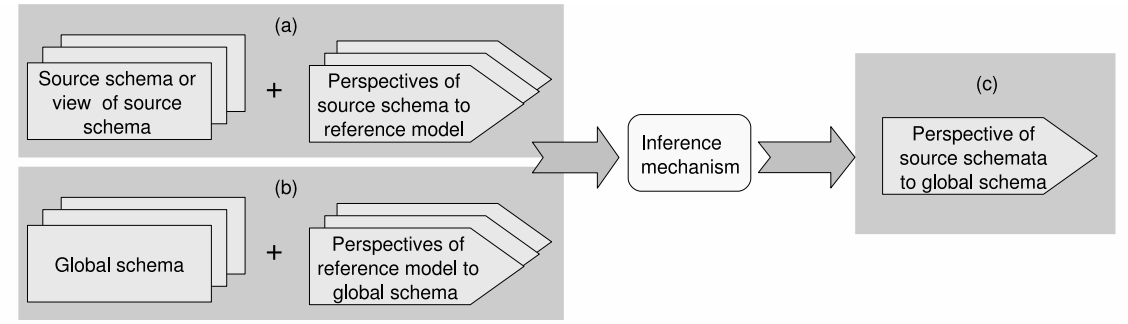

Figure 5: Sketch of the inference mechanism.

The rule-based rewriting system is formed by a set of rules having the general form:

$$\textbf{Rule:}\frac{X \Rightarrow Y}{Z} \quad (\text{ read } X \text{ is rewritten in } Y \text{ if } Z \text{ is valid }), \tag{4}$$

In (4), **Rule** is the name of the rule. X and Y can be formed by any of the following expressions: a CA pattern expression, a MF pattern signature, or a component pattern expression. CA pattern expressions and MF pattern signatures are expressions conforming to the L_{PS} syntax to declare, respectively, CAs and MF signatures, being that some of their elements are variables to be used in a unification process. Component pattern expressions are expressions conforming to the L_S or the L_{PS} syntaxes to represent properties, path expressions, functions with n-ary arguments, values, or conditions of selection (predicates), being that some of their elements are variables to be used in a unification process. Z is a condition formed by a set of CA pattern expressions, or expressions of the forms: a) $A \Rightarrow B$ such that A and B are component pattern expressions; b) (***FK***,C,_,C′,_) such that ***FK*** is a foreign key name of a class/relation C that refers to a class/relation C′; c) **p**:♮C ∈ ***type***(C′), such that **p** is a property name declared in a class/relation C′ that refers to a class C.[3] CA pattern expressions, MF pattern signatures, and component pattern expressions are formally defined in the following text.

[3] ***type***() is a function defined in language L_S that returns the structural type of a relation/class.

Definition 1 (*CA pattern expression) Let $\hat{\mathcal{A}}$ be a set of correspondence assertions defined in language L_{PS}. A CA pattern expression is an expression having the general form:*

$$\underline{K} \rightarrow \underline{L}$$

with K and L being variables that can be instatiated with, respectively, the left-side and the right-side of a correspondence assertion $\hat{\psi} \in \hat{\mathcal{A}}$.

Definition 2 (*MF pattern signature) Let $\mathcal{T}$ a set of data types and $\hat{\mathcal{L}}$ a set of schemata. A MF pattern signature is an expression having one of the following forms:*

match : $((\underline{\boldsymbol{S}_1}\,[\underline{\mathrm{C}_1}]\,, \underline{\tau_1}) \times (\underline{\boldsymbol{S}_2}\,[\underline{\mathrm{C}_2}]\,, \underline{\tau_2})) \rightarrow$ *Boolean*;

match : $\left((\underline{\boldsymbol{S}_1}\,[\underline{\mathrm{C}_1}]\,, \underline{\tau_1}) \times \left(\underline{\boldsymbol{S}_2}\left[\underline{\mathrm{C}_2}\left(\underline{\boldsymbol{pred}}\right)\right], \underline{\tau_2}\right)\right) \rightarrow$ *Boolean*;

match : $\left(\left(\underline{\boldsymbol{S}_1}\,[\underline{\mathrm{C}_1}]\,, \underline{\tau_1}, \{\underline{\boldsymbol{p}'_1} : \underline{\tau'_1}, \ldots, \underline{\boldsymbol{p}'_n} : \underline{\tau'_n}\}\right) \times \left(\underline{\boldsymbol{S}_2}\,[\underline{\mathrm{C}_2}]\,, \underline{\tau_2}, \{\underline{\boldsymbol{p}''_1} : \underline{\tau''_1}, \ldots, \underline{\boldsymbol{p}''_n} : \underline{\tau''_n}\}\right)\right) \rightarrow$ *Boolean*;

match : $\left(\left(\underline{\boldsymbol{S}_1}\,[\underline{\mathrm{C}_1}]\,, \underline{\tau_1}, \{\underline{\boldsymbol{p}'_1} : \underline{\tau'_1}, \ldots, \underline{\boldsymbol{p}'_n} : \underline{\tau'_n}\}\right) \times \left(\underline{\boldsymbol{S}_2}\left[\underline{\mathrm{C}_2}\left(\underline{\boldsymbol{pred}}\right)\right], \underline{\tau_2}, \{\underline{\boldsymbol{p}''_1} : \underline{\tau''_1}, \ldots, \underline{\boldsymbol{p}''_n} : \underline{\tau''_n}\}\right)\right) \rightarrow$
$\rightarrow$ *Boolean*

*With $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ being variables that can be instatiated with any of the schemata belonging to $\hat{\mathcal{L}}$; C_1 and C_2 being variables that can be instantiated with any class/relation of the schemata belonging to $\hat{\mathcal{L}}$; **pred** being a variable that can be instantiated with a predicate (as defined in L_{PS}); $\boldsymbol{p}'_i$ and $\boldsymbol{p}''_i$, for $1 \leq i \leq n$, are variables that can be instantiated with any property of a class/relation of the schemata belonging to $\hat{\mathcal{L}}$; τ_1, τ_2, τ'_i, τ''_i, for $1 \leq i \leq n$, are variable that can be instantiated with data types belonging to $\mathcal{T}$ (defined in schemata belonging to $\hat{\mathcal{L}}$).*

Definition 3 (*Component pattern expression) Let $\hat{\mathcal{L}}$ be a set of schemata. A Component pattern expression is a expression formed by a single variable that can be instantiated with a predicate **pred**; or is an expression having one of the following forms:*

$$\underline{\mathrm{S}}\,[\underline{\mathrm{C}_1}] \bullet \underline{\boldsymbol{p}}$$
$$\underline{\mathrm{S}}\,[\underline{\mathrm{C}_1}] \bullet \underline{\varrho}$$
$$\underline{\varphi}\,(\underline{X_1}, \underline{X_2}, \ldots, \underline{X_n})$$
$$\underline{\mathrm{S}}\,[\underline{\mathrm{C}_1}] \bullet \underline{\boldsymbol{p}}\{\underline{\boldsymbol{p}''}\}$$
$$\underline{\ell_1} : \underline{\mathrm{C}_1} \rightarrowtriangle \underline{\mathrm{C}_2}$$

*With **S** being a variable that can be instatiated with any of the schemata belonging to $\hat{\mathcal{L}}$; C_1 and C_2 being variables that can be instantiated with any class/relation of the schemata belonging to $\hat{\mathcal{L}}$; ϱ being a variable that can be instantiated with a (value or reference) path expression as defined in L_S; ℓ_1 being a variable that can be instantiated with a link of a path expression; **p** and $\boldsymbol{p}''$ being variables that can be instantiated with any property of a class/relation of the schemata belonging to $\hat{\mathcal{L}}$, being that $\boldsymbol{p}''$ is part of the structural type of **p**; φ being a variable that can be instatiated with any function with $n \geq 1$ arguments that returns a value; and X_i, $1 \leq i \leq n$, being variables that can be instantiated with other component pattern expressions.*

A condition *Z* is valid when all of its expressions are valid: a) the CA pattern expression is valid if there is a CA, which is declared in one of the perspective schemata associated to the *origin* schemata or the *destination* schema, that unifies with it; b) the expression of the form $A \Rightarrow B$, such that *A* and *B* are component pattern expressions, is valid if there is a rule which unifies with it and which is recursively applied; c) the expression of the form (***FK***,C,_,C′,_) is valid if there is a foreign key declaration declared as required in one of the perspective schemata associated to the *origin* schemata that unifies with it; d) the expression of the form **p**:♮C ∈ ***type***(C′) is valid if there are both a class/relation and a property, both declared as required in one of the perspective schemata associated to the *origin* schemata, that unify with them. A formal definition of a valid condition is as follows:

Definition 4 (*Valid condition (in an inference rule)) Let $\hat{\mathcal{L}}_p$ be a set of perspective schemata and $\hat{\mathcal{A}}_i$ a set of correspondence assertions declared in some perspective schema belonging to $\hat{\mathcal{L}}_p$. Let also $\{X_1, X_2, ..., X_n\}$ be a condition Z in an inference rule. Z is a valid condition iff for each X_i, $1 \leq i \leq n$, if:*

- *X_i is a CA pattern expression, then there is a $\hat{\psi} \in \hat{\mathcal{A}}_i$ that unifies with X_i;*
- *X_i is an expression the form $A \Rightarrow B$, such that A and B are component pattern expressions, then there is an inference rule that unifies with X_i whose condition is a valid condition,*
- *X_i is an expression of the form (**FK**,C,_,C′,_) then there is a foreign key declaration, as defined in L_S, declared as required in some perspective schema belonging to $\hat{\mathcal{L}}_p$ that unifies with X_i;*
- *X_i is an expression of form **p**:♮C ∈ **type**(C′), then there are both a class/relation and a property, both declared as required in some perspective schema belonging to $\hat{\mathcal{L}}_p$, such that **p**:♮C ∈ **type**(C′).*

When *X* and *Y* are CA pattern expressions, the rule are rewritten-rules that rewrite CAs in other CAs (RR-CAs). When *X* and *Y* are MF pattern expressions, the rule are rewritten-rules that rewrite MFs in other MFs (RR-MFs). When *X* and *Y* are component pattern expressions, the rule are substitution-rules that rewrite components in other components (RR-Cs). The latter are used as an intermediary process by the RR-CAs and RR-MFs.

An example of a RR-CA is as follows:

$$\textbf{RR-CA1:}\frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^{D}}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^{D}}\right] \rightarrow \underline{\mathbf{K}^{S}}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathbf{K}^{S}}}. \quad (5)$$

In (5) all variables are indicated by an underline. **D** is the *destination* schema, **RM** is the reference model schema, and **S** is a variable that will be instantiated with some of the *origin* schemata. C^{D} is a variable that will be instantiated with a class/relation of the schema **D**; mutatis mutandis to C^{RM}. **K** is a variable that will be instantiated with the right side of a CA pattern expression of extension. The letter S in $\mathbf{K}^{S}$ means that all elements in that expression belong to schema **S**. The value of **S** and **K** will depend on which CA, that is declared in the perspective schema associated to some *origin* schemata, will unify with the condition of the rule. The notation in (5) will be used through the paper to explain examples of rules.

The rule **RR-CA1** rewrites an ECA of equivalence, which connects a class/relation C^{D} of the *destination* schema to a class/relation C^{RM} of the reference model schema, into an ECA, which connect the

class/relation C^D to a class/relation C^S of some *origin* schema; when is provided an ECA that connect the class/relation C^{RM} to C^S.

An example of a RR-MF is as follows:

$$\textbf{RR-MF1:}\frac{\begin{array}{l}\textbf{match} : \left(\left(\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right], \underline{\tau^{RM}}\right) \times \left(\mathrm{D}\left[\underline{\mathrm{C}^{D}}\right], \underline{\tau^{D}}\right)\right) \rightarrow \mathrm{Boolean} \Rightarrow \\ \textbf{match} : \left(\left(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^{S}}\right], \underline{\tau^{S}}\right) \times \left(\mathrm{D}\left[\underline{\mathrm{C}^{D}}\right], \underline{\tau^{D}}\right)\right) \rightarrow \mathrm{Boolean}\end{array}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^{S}}\right]}. \quad (6)$$

In (6) τ is a data type. The rule **RR-MF1** rewrites a match function signature, which matches a class/relation C^{RM} of the reference model schema to a class/relation C^D of the *destination* schema, in a match function signature that matches a class/relation C^S of some *origin* schema to the class/relation C^D, when is provided an ECA of equivalence that connects the class/relation C^{RM} to C^S.

An example of RR-C is as follows:

$$\textbf{RR-C1:}\ \frac{\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\mathbf{A}^{S}}}. \quad (7)$$

In (7) $\boldsymbol{p}^{RM}$ is a variable that will be instantiated with a property of a class/relation and **A** is a variable that will be instantiated with a component pattern expression. Similar to $\mathbf{K}^S$ in (5), the letter S in $\mathbf{A}^S$ means that all elements into that expression belong to schema **S**. The value of **S** and **A** will depend on which CA declared in the perspective schema associated to some *origin* schemata will unify with the condition of the rule.

The rule **RR-C1** rewrites a property $\boldsymbol{p}^{RM}$ of a class/relation of the reference model schema in a property, a path expression, or a function of some *origin* schema, when is provided an PCA that connects the property $\boldsymbol{p}^{RM}$ to this property, path expression, or function. The whole set of proposed rules can be found in appendix A.

1: **procedure** INFER_CAS($A^{\mathbf{G}} \rightarrow A^{\mathbf{RM}}$,$CAs$)
2: **repeat**
3: find $A^{\mathbf{G}} \rightarrow A^{\mathbf{Si}}$ applying the inference rule R:
4: R:$\frac{A^{\mathbf{G}} \rightarrow A^{\mathbf{RM}} \Rightarrow A^{\mathbf{G}} \rightarrow A^{\mathbf{Si}}}{\texttt{conditions}}$;
5: add $A^{\mathbf{G}} \rightarrow A^{\mathbf{Si}}$ to CAs;
6: **until** all rules for rewriting CAs have been tested
7: **end procedure**

Figure 6: The pseudo-code to the inference mechanism to generate new CAs.

A pseudo-code detailing as new CAs are deduced is shown in Fig. 6. In Fig. 6, **G** is the *destination* schema, **RM** a reference model schema, and $\mathbf{S}_i$, i $\geq$ 1, *origin* schemata. The algorithm tries to find, for each CA $A^{\mathbf{G}} \rightarrow A^{\mathbf{RM}}$ assigning the global schema to the reference model schema, one or more CAs $A^{\mathbf{G}} \rightarrow A^{\mathbf{Si}}$ as a result of applying to $A^{\mathbf{G}} \rightarrow A^{\mathbf{RM}}$ some rule for rewriting CAs. Notice that, in the condition of the rule can exists expressions of the form $A \Rightarrow B$. In this case, the recursivity will be present. For instance, a new ECA:

$$\mathbf{P_{S1|G}}\,[\text{CUSTOMER}] \rightarrow \mathbf{S1}\,[\text{CUSTOMER}]$$

can be created based on ψ_3 (see Fig. 4), using the rule **RR-CA1** since that the CA ψ_6 is defined in perspective schema $\mathbf{P}_{S1|RM}$ (as shown in Fig. 7).

Extension Correspondence Assertion (ECA)	
ψ_6:	$\mathbf{P_{S1\|RM}}\,[\text{CUSTOMER}] \rightarrow \mathbf{S1}\,[\text{CUSTOMER}]$
ψ_7:	$\mathbf{P_{S2\|RM}}\,[\text{CUSTOMER}] \rightarrow \mathbf{S2}\,[\text{CUSTOMER}]$

Figure 7: More examples of correspondence assertions.

A pseudo-code detailing as new MF signatures are deduced is shown in Fig. 8. In Fig. 8 **K** and **L** are pairs (classes/relations, data type) of the reference model schema or of the *destination* schema, while $\mathbf{K}'$ and $\mathbf{L}'$ are pairs (classes/relations, data type) of some *origin* schemata or of the *destination* one. For each MF **M** that is declared in the perspective schema associated to the *destination* schema, the algorithm tries to find one or more MFs as a result of applying to **M** some rule for rewriting MFs. For instance, two new MF signatures:

$$\mathbf{match}((\mathbf{S1}[\text{CUSTOMER}],\tau_1)\times(\mathbf{G}[\text{CUSTOMER}],\tau_2))\rightarrow\text{Boolean}$$
$$\mathbf{match}((\mathbf{S2}[\text{CUSTOMER}],\tau_1)\times(\mathbf{G}[\text{CUSTOMER}],\tau_2))\rightarrow\text{Boolean}$$

can be created based on MF signature presented in Fig. 3, using the rule **RR-MF1** twice, since as the CAs ψ_6 and ψ_7 are defined, respectively, in perspective schemata $\mathbf{P}_{S1|RM}$ and $\mathbf{P}_{S2|RM}$ (as shown in Fig. 7).

```
1: procedure INFER_MFS(match(K×L)→Boolean ,MFs)
2:   repeat
3:     find match(K'×L')→Boolean applying the inference rule R:
4:     R: match(K×L)→Boolean ⇒ match(K'×L')→Boolean  (conditions);
5:     add match(K'×L')→Boolean to MFs;
6:   until all rules for rewriting MFs have been tested
7: end procedure
```

Figure 8: The pseudo-code to the inference mechanism to generate new MFs.

A pseudo-code with the iteration of the process to generate a new perspective is shown in Fig. 9. In Fig. 9 P_T is a perspective schema from the reference model to the global schema; P_j, $1 \neq \text{j} \neq \text{n}$, are perspective schemata from source schemata to the reference model; and P_I is the inferred perspective schema from source schemata to the global schema. All elements of the perspective schemata are grouped in lists: $classList$, $relationList$, $keyList$, $caList$, and $mfList$. The three first lists hold 'require' declarations of, respectively, classes, relations, and keys and foreign keys. $caList$ contains correspondence assertion declarations, and $mfList$ has match function signatures.

This mechanism has been developed as part of a proof-of-concept prototype using a Prolog language. Beside the inference mechanism module, the prototype consist of more five modules, such as the *schema manager*, and the *ISCO translator*. The *schema manager* module is employed by the designer to manage the schemata (in language L_S) as well as the perspective schemata (in language L_{PS}). The *ISCO translator* performs the mapping between schemata written in L_S or L_{PS} languages to schemata defined in a language programming called Information Systems COnstruction language (ISCO) [9]. ISCO is based

```
1: procedure GENERATENEWPERSPECTIVE(P_T, P_1, ..., P_n, P_I)
2:     for each CA A^G → A^RM in P_T.caList do
3:         infer_CAs(A^G → A^RM,{A^G → A^Si});
4:         add CAs A^G → A^Si to P_I.caList;
5:     end for
6:     for each MF m in P_T.mfList do
7:         infer_MFs(m,{m'_i});
8:         add MFs m'_i to P_I.mfList
9:     end for
10:    for each E in classList/relationList/keyList do
11:        create a require declaration to P_I;
12:        add it, appropriately, to P_I.classList/
13:        P_I.relationList/P_I.keyList
14:    end for
15: end procedure
```

Figure 9: The pseudo-code to the creation of inferred perspective schemata.

on a contextual constraint logic programming that allows the construction of information systems. It can define (object) relational schemata, represent data, and transparently access data from various heterogeneous sources in a uniform way, like a mediator system [10]. Thus, it is possible to access data from information sources using the perspective schema in ISCO. Furthermore, once the perspective schema from source schemata to the global schema has been inferred, as well as the new match functions have been implemented, it can be translated to ISCO language and so the data of the global schema can be queried.

5. Related work

The database community has been for many years engaged with the problem of data integration. Researches on this area have developed in several important directions: schema matching, data quality, to cite a few (see [11] for a survey), which can cover different architectures (e.g. FDBSs and DWs), representation of data and involved data models (e.g. relational and non-structured). Recent research in Federated Database Systems (FDBSs) has included: behaviour integration [12], integration of non-traditional data (e.g biomedical [13, 14], intelligence data [15], and web source [16]), interactive integration of data [17, 18], and federated data warehouse systems [19]. All these approaches use a global schema, but do not deal with a reference model schema. Similarly the authors' research of the current paper, [16] uses correspondence assertions (in this case, for specifying the semantics of XML-based mediators). However, their CAs only deal with part of the semantic correspondence managed here. Furthermore, they assume that there is a universal key to determine when two distinct objects are the same entity in the real-world, which is a supposition often unreal.

Researches in Data Warehouses (DWs) have focused on technical aspects such as multidimensional data models (e.g. [20, 21, 22, 23, 24, 25]) as well as the materialised view definition and maintenance (e.g. [26]). In particular, the most conceptual multidimensional models are extensions to the Entity-Relationship model (e.g. [27, 28, 29, 30]) or extensions to UML (e.g. [31, 32, 33]).

Reference in [34] focuses on an ontology-based approach to determine the mapping between at-

tributes from the source schemata and the DW schema, as well as to identify the transformations required for correctly moving data from source information to the DW. Their ontology, based on a common vocabulary as well as a set of data annotations (both provided by the designer), allows formal and explicit description of the semantic of the sources and the DW schemata. However, their strategy requires a deep knowledge of all schemata involved in the DW system, in what is usually not an usual task. In the proposed research of the present paper, it is dispensable, since each schema (source or DW) needs to be related only to the reference model one. Additionally, in [34] there is nothing about the matching of instances.

The approach closest to authors' research is described in [35]. Similar to this study, their proposal includes a reference model (cited as "enterprise model") designed using an Enriched Entity-Relationship (EER) model. However, unlike the authors' research, all their schemata, including the DW schema, are formed by relational structures, which are defined as views over the reference model. Their proposal provides the user with various levels of abstraction: conceptual, logical, and physical. In their conceptual level, they introduce the notion of intermodel assertions that precisely capture the structure of an EER schema or allow for the specifying of the relationship between diverse schemata. However, any transformation (e.g. restructuring of schema and values) or mapping of instances is deferred for the logical level, unlike the current work. In addition, they did not deal with complex data, integrity constraints, and path expressions, as this research does.

6. Conclusions and future works

In this paper, the authors have presented a proposal to automatically connect a global schema to its sources by using an inference mechanism taking into account a reference model. In proposed approach the relationship between the global schema and the source schemata is made explicitly and declaratively through correspondence assertions. This approach is particularly useful in data integration systems that define a common or canonical schema, such as in Data Warehouse (DW) systems and in Federated Database (FDB) systems. An advantage of the proposed approach is that by using the reference model the designer user does not need to have a deep knowledge of all schemata involved in the DW system or in the federation system, since that each schema (source or global) needs to be related only to the reference model one. Thus, the designer user can describe the global system without concerns about where the sources are or how they are stored. Besides, the process of data integration can be incrementally done in two sense:

1. View schemata can be created as an intermediary process to relate portions of data that have been integrated (those view schemata, in turn, are related to the reference model). Thus the data integration process can be divided in small parts, instead of being seen as a whole, turning the integration task easiest.
2. New source schemata can be added or actual source schemata can eventually change. It is completely transparent to the DW systems or FDB systems since the relationship between the global schema and its source schemata is automatically created by the inference mechanism.

A prototype Prolog-based has been developed to allow the description of schemata and perspective schemata in the proposed language as well as to infer new perspective schemata based on other ones. The matching functions can be implemented using Prolog itself or external functions. In addition, the prototype include translators from the proposed language to the ISCO one. ISCO [9] allows access to

heterogeneous data sources and to perform arbitrary computations. Thus, user-queries can be done, in a transparent way, to access the information sources, like occurs in mediator systems [10].

For future work, investigations will be made into how the perspective schemata can be used to automate the materialisation of the data in the DWs or in other repository of a data integration environment. Another important direction for future work is the development of a graphical user-friend interface to declare the schemata in the proposed language, and thus, to hide some syntax details.

References

[1] Claudia Imhoff, Nicholas Galemmo, and Jonathan G. Geiger, *Mastering Data Warehouse Design - Relational and Dimensional Techniques*, Wiley Publishing, 2003.

[2] Valéria Magalhães Pequeno and João Carlos Gomes Moura Pires, "Using perspective schemata to model the ETL process", in *ICMIS 2009 :Intl. Conf. on Management Information Systems*, France, June 2009, to appear.

[3] Edgar F. Codd, "A relational model of data for large shared data banks", in *Communications of the ACM*, 1970, pp. 377–387.

[4] Rick G.G. Cattell and D. Barry, Eds., *The Object Database Standard ODMG 3.0*, Morgan Kaufmann Publishers, 2000.

[5] Valéria Magalhães Pequeno and João Carlos Gomes Moura Pires, "A formal object-relational data warehouse model", Tech. Rep., Universidade Nova de Lisboa, November 2007.

[6] Valéria Magalhães Pequeno and João Carlos Gomes Moura Pires, "Using perspective schemata to model the ETL process", Tech. Rep., Universidade Nova de Lisboa, 2009.

[7] G. Zhou, Richard Hull, and Roger King, "Generating data integration mediators that use materialization", *J. Intell. Inf. Syst.*, vol. 6(2/3), pp. 199–221, May 1996.

[8] Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of database systems*, Pearson Education, 5th edition, 2006.

[9] Salvador Abreu and Vitor Nogueira, "Using a logic programming language with persistence and contexts", in *INAP'05: 16th Intl. Conf. on applications of declarative programming and knowledge management*. 2006, vol. 4369 of *Lecture Notes in Computer Science*, pp. 38–47, Springer, (Revised Selected Papers).

[10] G. Wiederhold, "Mediators in the architecture of future information systems", in *IEEE Computer*, 1992, vol. 25(3), pp. 38–49.

[11] Alon Y. Halevy, Anand Rajaraman, and Joann J. Ordille, "Data integration: The teenage years.", in *VLDB*, 2006, pp. 9–16.

[12] Markus Stumptner, Michael Schrefl, and Georg Grossmann, "On the road to behavior-based integration", in *APCCM: First Asia-Pacific Conf. on Conceptual Modelling*, 2004, pp. 15–22.

[13] Brenton Louie, Peter Mork, Fernando Martin-Sanchez, Alon Halevy, and Peter Tarczy-Hornoch, "Data integration and genomic medicine", *Journal of Biomedical Informatics*, vol. 40, pp. 5–13, 2007.

[14] Pavithra G. Naidu, Mathew J. Palakal, and Shielly Hartanto, "On-the-fly data integration models for biological databases", in *SAC'07: Proceedings of the 2007 ACM symposium on Applied computing*, USA, 2007, pp. 118–122, ACM.

[15] S. Yoakum-Stover and T. Malyuta, "Unified architecture for integrating intelligence data", in *DAMA: Europe Conf.*, UK, 2008.

[16] Vânia Maria Ponte Vidal, Bernadette Farias Lóscio, and Ana Carolina Salgado, "Using correspondence assertions for specifying the semantics of XML-based mediators", in *Workshop on Information Integration on the Web*, 2001, vol. 3(11).

[17] Zachary G. Ives, Craig A. Knoblock, Steven Minton, Marie Jacob, Partha Pratim Talukdar, Rattapoom Tuchinda, Jos Luis Ambite, Maria Muslea, and Cenk Gazen, "Interactive data integration through smart copy & paste.", in *CIDR:4th Biennial Conference on Innovative Data Systems Research*. 2009, www.crdrdb.org.

[18] Robert Mccann, Anhai Doan, Vanitha Varadarajan, and Er Kramnik, "Building data integration systems via mass collaboration", in *WebDB: Intl. Workshop on the Web and Databases*, USA, 2003.

[19] Stefan Berger and Michael Schrefl, "From federated databases to a federated data warehouse system", in *HICSS '08: 41st Annual Hawaii Intl. Conf. on System Sciences*, USA, 2008, p. 394, IEEE Computer Society.

[20] Dov Dori, Roman Feldman, and Arnon Sturm, "From conceptual models to schemata: An object-process-based data warehouse construction method", *Inf. Syst.*, vol. 33, no. 6, pp. 567–593, 2008.

[21] E. Malinowski and E. Zimányi, "A conceptual model for temporal data warehouses and its transformation to the ER and the object-relational models", *Data knowl. eng.*, vol. 64, no. 1, pp. 101–133, 2008.

[22] Juan Manuel Pérez, Rafael Berlanga, María José Aramburu, and Torben Bach Pedersen, "A relevance-extended multi-dimensional model for a data warehouse contextualized with documents", in *DOLAP'05: Proc. of the 8th ACM Intl. Workshop on Data Warehousing and OLAP*, USA, 2005, pp. 19–28, ACM.

[23] Matteo Golfarelli, Vittorio Maniezzo, and Stefano Rizzi, "Materialization of fragmented views in multidimensional databases", *Data Knowl. Eng.*, vol. 49, no. 3, pp. 325–351, 2004.

[24] Bodo Husemann, Jens Lechtenborger, and Gottfried Vossen, "Conceptual data warehouse modeling", in *Design and Management of Data Warehouses*, 2000, p. 6.

[25] S. Rizzi, "Conceptual modeling solutions for the data warehouse", *In Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, vol. Information Science Reference, pp. 208–227, 2008.

[26] Robert Wrembel, "On a formal model of an object-oriented database with views supporting data materialisation", in *Proc. of the Conf. on Advances in Databases and Information Systems*, 1999, pp. 109–116.

[27] Enrico Franconi and Anand Kamble, "A data warehouse conceptual data model", *Proc. of the Int. Conf. on Scientific and Statistical Database Management*, vol. 00, pp. 435–436, 2004.

[28] Anand S. Kamble, "A conceptual model for multidimensional data", in *APCCM'08: Proc. of the 15th on Asia-Pacific Conf. on Conceptual Modelling*, Australia, 2008, pp. 29–38, Australian Computer Society, Inc.

[29] Carsten Sapia, Markus Blaschka, Gabriele Höfling, and Barbara Dinter, "Extending the E/R model for the multidimensional paradigm", in *Proc. of the Workshops on Data Warehousing and Data Mining*, 1999, pp. 105–116.

[30] Nectaria Tryfona, Frank Busborg, and Jens G. Borch Christiansen, "starER: a conceptual model for data warehouse design", in *DOLAP '99: Proc. of the 2nd ACM Intl. Workshop on Data warehousing and OLAP*, USA, 1999, pp. 3–8, ACM.

[31] Sergio Luján-Mora, Juan Trujillo, and Il-Yeol Song, "A UML profile for multidimensional modelling in data warehouses", *Data Knowl. Eng.*, vol. 59, no. 3, pp. 725–769, 2005.

[32] T. B. Nguyen, A. Min Tjoa, and Roland Wagner, "An object oriented multidimensional data model for OLAP", in *Web-Age Inf. Management*, 2000, pp. 69–82.

[33] Juan Trujillo, Manuel Palomar, and Jaime Gomez, "Applying object-oriented conceptual modeling techniques to the design of multidimensional databases and OLAP applications", *WAIM'00. Lecture Notes in Computer Science (LNCS)*, vol. 1846, pp. 83–94, 2000.

[34] Dimitrios Skoutas and Alkis Simitsis, "Designing ETL processes using semantic web technologies", in *DOLAP'06: Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, USA, 2006, pp. 67–74, ACM.

[35] Diego Calvanese, Luigi Dragone, Daniele Nardi, Riccardo Rosati, and Stefano M. Trisolini, "Enterprise modeling and data warehousing in TELECOM ITALIA", *Inf. Syst.*, vol. 31, no. 1, pp. 1–32, 2006.

APPENDIX

A. Inference Rules

Hereafter, condider the following notation:

- $\mathcal{L}$ is a set of schema names.
- $\mathcal{L}_p$ is a set of perspective schema names.
- $\hat{\mathcal{L}}_p$ is a set of perspective schemata.
- $\mathcal{W}$ is a set of typed values.
- $\mathcal{T}$ is a set of data types.
- **D** is the *destination* schema, **RM** is the reference model schema, and **S** is a variable that can be instantiated with any of the *origin* schemata.
- C^D is a variable that can be instantiated with any class/relation of the schema **D**; mutatis mutandis to C^{RM} and C^S.
- $\mathrm{P}_{RM|D} \in \mathcal{L}_p$ is a perspective schema name, with **RM** being the *base* schema and **D** the *target* schema, mutatis mutandis to $\mathrm{P}_{S|D}$ and $\mathrm{P}_{S|RM}$.
- All variables are indicated by an underline.

A.1. Substituition-rules

The following notation will be used in this Section and through the paper to explain rules:

- **A** is a variable that can be instantiated with a component pattern expression, being that the letter S in $\mathbf{A}^S$ means that all elements into that expression belong to schema **S**, mutatis mutandis to $\mathbf{A}^{RM}$.
- $\boldsymbol{p}^{RM}$ is a variable that can be instantiated with any property of a class/relation of the schema **RM**, mutatis mutandis to $\boldsymbol{p}^S$.
- ϱ is a variable that can be instantiated with a (value or reference) path expression as defined in L_S, being that the letter S in ϱ^S means that all elements into this expression belong to schema **S**, mutatis mutandis to ϱ^{RM}.
- ℓ_i are variables that can be instantiated with a links of a path expression, being that the letter S in ℓ_i^S means that this element belong to schema **S**, mutatis mutandis to ℓ_i^{RM}.
- **pred** is a predicate as defined in L_{PS}, being **op** operands in **pred** such that **op** $\in \{<, >, \leq, \geq, \sharp, =\}$ and **B** is an expression in **pred** such that **B** = **A** or **B** = ***w***, with $\boldsymbol{w} \in \mathcal{W}$. The letter S in $\mathbf{pred}^S$ and $\mathbf{B}^S$ means that all elements into those expressions belong to schema **S**, mutatis mutandis to $\mathbf{pred}^{RM}$ and $\mathbf{B}^{RM}$.
- φ is a variable that can be instatiated with any function with n ≥ 1 arguments that returns a value.
- ***FK*** is a foreign key name.
- The expression ***p***: ♮C means that the type of the property ***p*** is a reference to the class C.
- ***type***() is a function defined in language L_S that returns the structural type of a relation/class.

The substituition-rules are formed by 12 rules as follows:

$$\textbf{RR-C1}: \frac{\text{RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}}}{\text{P}_{\underline{S}|RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\mathbf{A}^{S}}}$$

$$\textbf{RR-C2}: \frac{\text{RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\varrho^{RM}} \Rightarrow \underline{\text{S}}\,[\underline{\text{C}^{s}}] \bullet \underline{\varrho^{S}}}{\begin{array}{l}\underline{\ell_{i+1}^{RM}} \Rightarrow \underline{\ell_{i+1}^{S}},\ \text{for } 0 \leq i \leq n-1,\\ \text{P}_{\underline{S}|RM}\left[\underline{\text{C}_{n}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\text{S}}\left[\underline{\text{C}_{n}^{S}}\right] \bullet \underline{\boldsymbol{p}^{S}}\end{array}}$$

$$\textbf{RR-C3}: \frac{\text{RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\varrho^{RM}} \Rightarrow \underline{\text{S}}\,[\underline{\text{C}^{s}}] \bullet \underline{\varrho^{S}}}{\underline{\ell_{i+1}^{RM}} \Rightarrow \underline{\ell_{i+1}^{S}},\ \text{for } 0 \leq i \leq n-1}$$

$$\textbf{RR-C4}: \frac{\underline{\varphi}\left(\underline{\mathbf{A}_{1}^{RM}}, \underline{\mathbf{A}_{2}^{RM}}, \ldots, \underline{\mathbf{A}_{n}^{RM}}\right) \Rightarrow \underline{\varphi}\left(\underline{\mathbf{A}_{1}^{S}}, \underline{\mathbf{A}_{2}^{S}}, \ldots, \underline{\mathbf{A}_{n}^{S}}\right)}{\underline{\mathbf{A}_{i}^{RM}} \Rightarrow \underline{\mathbf{A}_{i}^{S}}, \text{for } 1 \leq i \leq n}$$

$$\textbf{RR-C5}: \frac{\text{RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}}\{\underline{\boldsymbol{p}_{i}}\} \Rightarrow \underline{\mathbf{A}_{i}^{S}}, \text{for } 1 \leq i \leq n}{\text{P}_{\underline{S}|RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}}\{\boldsymbol{p}_{1}, \boldsymbol{p}_{2}, ..., \boldsymbol{p}_{n}\} \rightarrow \left(\underline{\mathbf{A}_{1}^{S}}, \underline{\mathbf{A}_{2}^{S}}, \ldots, \underline{\mathbf{A}_{n}^{S}}\right)}$$

$$\textbf{RR-C6}: \underline{\boldsymbol{w}} \Rightarrow \underline{\boldsymbol{w}}$$

$$\textbf{RR-C7}: \frac{\underline{\mathbf{A}^{RM}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{S}}}{\begin{array}{l}\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}},\\ \underline{\mathbf{B}^{RM}} \Rightarrow \underline{\mathbf{B}^{S}}\end{array}}$$

$$\textbf{RR-C8}: \frac{\underline{\mathbf{A}^{RM}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{RM}} \text{ and } \underline{\textbf{pred}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{S}} \text{ and } \underline{\textbf{pred}^{S}}}{\begin{array}{l}\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}},\\ \underline{\mathbf{B}^{RM}} \Rightarrow \underline{\mathbf{B}^{S}},\\ \underline{\textbf{pred}^{RM}} \Rightarrow \underline{\textbf{pred}^{S}}\end{array}}$$

$$\textbf{RR-C9}: \frac{\underline{\mathbf{A}^{RM}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{RM}} \text{ or } \underline{\textbf{pred}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}} \quad \underline{\textbf{op}} \quad \underline{\mathbf{B}^{S}} \text{ or } \underline{\textbf{pred}^{S}}}{\begin{array}{l}\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^{S}},\\ \underline{\mathbf{B}^{RM}} \Rightarrow \underline{\mathbf{B}^{S}},\\ \underline{\textbf{pred}^{RM}} \Rightarrow \underline{\textbf{pred}^{S}}\end{array}}$$

$$\textbf{RR-C10}: \frac{\underline{\boldsymbol{p}^{RM}} : \underline{\text{C}^{RM}} \rightarrow\!\triangleright \underline{\text{C}_{1}^{RM}} \Rightarrow \underline{\boldsymbol{p}^{S}} : \underline{\text{C}^{S}} \rightarrow\!\triangleright \underline{\text{C}_{1}^{S}}}{\begin{array}{l}\text{P}_{\underline{S}|RM}\left[\underline{\text{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\text{S}}\left[\underline{\text{C}^{S}}\right] \bullet \underline{\boldsymbol{p}^{S}},\\ \underline{\boldsymbol{p}^{S}} : \natural \underline{\text{C}_{1}^{S}} \in \textit{type}(\text{C}^{S})\end{array}}$$

$$\textbf{RR-C11}: \frac{\underline{\ell^{RM}}: \underline{\mathrm{C}^{RM}} \rightarrowtriangle \underline{\mathrm{C}_1^{RM}} \Rightarrow \underline{\boldsymbol{FK}^S}: \underline{\mathrm{C}^S} \rightarrowtriangle \underline{\mathrm{C}_1^S}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ (\boldsymbol{FK}^S, \mathrm{C}^S, \_, \mathrm{C}_1^S, \_)\end{array}}$$

$$\textbf{RR-C12}: \frac{\underline{\boldsymbol{FK}^{RM}}: \underline{\mathrm{C}^{RM}} \rightarrowtriangle \underline{\mathrm{C}_1^{RM}} \Rightarrow \underline{\boldsymbol{p}^S}: \underline{\mathrm{C}^S} \rightarrowtriangle \underline{\mathrm{C}_1^S}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ \mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}_1^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}_1^S}\right], \\ \underline{\boldsymbol{p}^S}: \natural\underline{\mathrm{C}_1^S} \in \boldsymbol{type}(\underline{\mathrm{C}^S})\end{array}}$$

A.2. rewritten-rules to rewrite CAs

The rules to rewrite CAs are subdivided in four groups in accordance to kind of CA involved. Thus, there are rules for rewriting PCAs, ECAs, SCAs and ACAs, which are presented in following text.

A.2.1. rewritten-rules to rewrite PCAs

Consider the following notation to describe the RR-PCAs:

- $\mathbf{G}^S$ is a variable that can be instantied with the right side of a CA pattern expression of property consisting of one of two forms: ($\mathbf{A}_1^S$, $\mathbf{A}_2^S$, ..., $\mathbf{A}_n^S$) or ($\mathbf{B}_1^S$,$\mathbf{pred}_1^S$), ($\mathbf{B}_2^S$,$\mathbf{pred}_2^S$), ..., ($\mathbf{B}_{n-1}^S$,$\mathbf{pred}_{n-1}^S$), $\mathbf{B}_n^S$.

The RR-PCAs are formed by five rules as follows:

$$\textbf{RR-PCA1}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\mathbf{A}^{RM}} \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\mathbf{A}^S}}{\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^S}}$$

$$\textbf{RR-PCA2}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\mathbf{G}^S}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\mathbf{G}^S}}$$

$$\textbf{RR-PCA3}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D}\{\underline{\boldsymbol{p}_1}, \underline{\boldsymbol{p}_2}, ..., \underline{\boldsymbol{p}_n}\} \rightarrow \left(\underline{\mathbf{A}_1^{RM}}, \underline{\mathbf{A}_2^{RM}}, \ldots, \underline{\mathbf{A}_n^{RM}}\right) \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D}\{\underline{\boldsymbol{p}_1}, \underline{\boldsymbol{p}_2}, ..., \underline{\boldsymbol{p}_n}\} \rightarrow \left(\underline{\mathbf{A}_1^S}, \underline{\mathbf{A}_2^S}, \ldots, \underline{\mathbf{A}_n^S}\right)\end{array}}{\underline{\mathbf{A}_i^{RM}} \Rightarrow \underline{\mathbf{A}_i^S}, \text{ for } 1 \leq i \leq n}$$

$$\textbf{RR-PCA4}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \left(\underline{\mathbf{A}_1^{RM}}, \underline{\mathbf{A}_2^{RM}}, \ldots, \underline{\mathbf{A}_n^{RM}}\right) \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \left(\underline{\mathbf{A}_1^S}, \underline{\mathbf{A}_2^S}, \ldots, \underline{\mathbf{A}_n^S}\right)}{\underline{\mathbf{A}_i^{RM}} \Rightarrow \underline{\mathbf{A}_i^S}, \text{ for } 1 \leq i \leq n}$$

$$\textbf{RR-PCA5}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right]\bullet\underline{\boldsymbol{p}^D} \rightarrow \left(\underline{\mathbf{B}_1^{RM}}, \underline{\mathbf{pred}_1^{RM}}\right); \left(\underline{\mathbf{B}_2^{RM}}, \underline{\mathbf{pred}_2^{RM}}\right); ...; \left(\underline{\mathbf{B}_{n-1}^{RM}}, \underline{\mathbf{pred}_{n-1}^{RM}}\right); \underline{\mathbf{B}_n^{RM}} \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \left(\underline{\mathbf{B}_1^{S}}, \underline{\mathbf{pred}_1^{S}}\right); \left(\underline{\mathbf{B}_2^{S}}, \underline{\mathbf{pred}_2^{S}}\right); ...; \left(\underline{\mathbf{B}_{n-1}^{S}}, \underline{\mathbf{pred}_{n-1}^{S}}\right); \underline{\mathbf{B}_n^{S}}\end{array}}{\underline{\mathbf{B}_i^{RM}} \Rightarrow \underline{\mathbf{B}_i^{S}}, \underline{\mathbf{pred}_i^{RM}} \Rightarrow \underline{\mathbf{pred}_i^{S}}, \text{ for } 1 \leq i \leq n}$$

A.2.2. rewritten-rules to rewrite ECAs

Consider the following notation to describe the RR-ECAs:

- $\mathbf{K}$ is a variable that can be instantiated with the right side of a CA pattern expression of extension, being that the letter S in $\mathbf{K}^S$ means that all elements in that expression belong to schema $\mathbf{S}$.
- $\diamond$ is any operand appearing in a ECA, i.e. $-, \cap$, or $\rtimes$.
- C_i, for $1 \leq \mathrm{i} \leq \mathrm{n}$, are class/relation names in some schema belonging to $\hat{\mathcal{L}}$.

The RR-ECAs are formed by four rules as follows:

$$\textbf{RR-ECA1}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \underline{\mathbf{K}^S}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathbf{K}^S}}$$

$$\textbf{RR-ECA2}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}\left(\underline{\mathbf{pred}^{RM}}\right)}\right] \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S\left(\underline{\mathbf{pred}^S}\right)}\right]}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ \underline{\mathbf{pred}^{RM}} \Rightarrow \underline{\mathbf{pred}^S}\end{array}}$$

$$\textbf{RR-ECA3}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}\left(\underline{\mathbf{pred}^{RM}}\right)}\right] \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}_1^S\left(\underline{\mathbf{pred}_1^S}\right)}\right] \diamond \underline{\mathrm{S}}\left[\underline{\mathrm{C}_2^S\left(\underline{\mathbf{pred}_2^S}\right)}\right] \diamond \ldots \diamond \underline{\mathrm{S}}\left[\underline{\mathrm{C}_n^S\left(\underline{\mathbf{pred}_n^S}\right)}\right]\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}_1^S}\right] \diamond \underline{\mathrm{S}}\left[\underline{\mathrm{C}_2^S}\right] \diamond \ldots \diamond \underline{\mathrm{S}}\left[\underline{\mathrm{C}_n^S}\right], \\ \underline{\mathbf{pred}^{RM}} \Rightarrow \underline{\mathbf{pred}_i^S}, \text{ for } 1 \leq i \leq n\end{array}}$$

$$\textbf{RR-ECA4}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}_1^{RM}}\right] \diamond \ldots \diamond \mathrm{RM}\left[\underline{\mathrm{C}_j^{RM}\left(\underline{\mathbf{pred}_j^{RM}}\right)}\right] \diamond \ldots \diamond \mathrm{RM}\left[\underline{\mathrm{C}_n^{RM}}\right] \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \underline{\mathbf{K}_1^S} \diamond \underline{\mathbf{K}_2^S} \diamond \ldots \diamond \underline{\mathrm{C}_j^S(\mathbf{pred}_j^S)} \diamond \ldots \diamond \underline{\mathbf{K}_n^S}\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}_i^{RM}}\right] \rightarrow \underline{\mathbf{K}_i^S}, \text{ for } 1 \leq i \leq j-1, j+1 \leq i \leq n, \\ \mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}_j^{RM}}\right] \rightarrow \underline{\mathrm{C}_j^S}, \underline{\mathbf{pred}_j^{RM}} \Rightarrow \underline{\mathbf{pred}_j^S}\end{array}}$$

A.2.3. rewritten-rules to rewrite SCAs

Consider the following notation to describe the RR-SCAs:

- $\mathbf{X}$ is a variable that can be instantiated with a property $\boldsymbol{p}$ or a path expression ϱ or a property $\boldsymbol{p}'$ into another property $\boldsymbol{p}$ (a structured type) (notation: $\boldsymbol{p}\{\boldsymbol{p}'\}$).

- $\mathbf{Q}$ is a variable that can be instantiated with the right side of a CA pattern expression of summation, being that the letter S in $\mathbf{Q}^S$ means that all elements in that expression belong to schema $\mathbf{S}$.
- ϑ is a variable that can be instantiated with the keywords *groupby* or *normalize*, the two possible kinds of SCA.
- $\boldsymbol{p}_i$, for $1 \leq \mathrm{i} \leq \mathrm{n}$, are property names belonging to classes/relations in some schema belonging to $\hat{\mathcal{L}}$.

The RR-SCAs are formed by four rules as follows:

$$\textbf{RR-SCA1}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^D}\right) \rightarrow \underline{\vartheta}(\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right](\underline{\mathbf{X}_1^{RM}},\ldots,\underline{\varphi}(\underline{\mathbf{A}_1^{RM}},\ldots,\underline{\mathbf{A}_n^{RM}}),\ldots,\underline{\mathbf{X}_n^{RM}})) \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^D}\right) \rightarrow \underline{\vartheta}(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right](\underline{\mathbf{X}_1^S},\ldots,\underline{\varphi}(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S}),\ldots,\underline{\mathbf{X}_n^S}))\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\mathbf{X}_i^{RM}} \Rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right] \bullet \underline{\mathbf{X}_i^S}, \text{ for } 1 \leq i \leq n, \\ \underline{\varphi}(\underline{\mathbf{A}_1^{RM}},\ldots,\underline{\mathbf{A}_n^{RM}}) \Rightarrow \underline{\varphi}(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S})\end{array}}$$

$$\textbf{RR-SCA2}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^D}\right) \rightarrow \underline{\vartheta}(\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\left(\underline{\mathbf{pred}^{RM}}\right)\right](\underline{\mathbf{X}_1^{RM}},\ldots,\underline{\varphi}(\underline{\mathbf{A}_1^{RM}},\ldots,\underline{\mathbf{A}_n^{RM}}),\ldots, \\ \ldots,\underline{\mathbf{X}_n^{RM}})) \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^D}\right) \rightarrow \underline{\vartheta}(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\left(\underline{\mathbf{pred}^S}\right)\right](\underline{\mathbf{X}_1^S},\ldots,\underline{\varphi}(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S}),\ldots,\underline{\mathbf{X}_n^S}))\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ \underline{\mathbf{pred}^{RM}} \Rightarrow \underline{\mathbf{pred}^S}, \\ \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\mathbf{X}_i^{RM}} \Rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right] \bullet \underline{\mathbf{X}_i^S}, \text{ for } 1 \leq i \leq n, \\ \underline{\varphi}(\underline{\mathbf{A}_1^{RM}},\ldots,\underline{\mathbf{A}_n^{RM}}) \Rightarrow \underline{\varphi}(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S})\end{array}}$$

$$\textbf{RR-SCA3}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^S}\right) \rightarrow \underline{\mathbf{Q}^S}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right]\left(\underline{\boldsymbol{p}_1^{RM}},\ldots,\underline{\boldsymbol{p}_n^{RM}}\right) \rightarrow \underline{\mathbf{Q}^S}, \\ \mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}_i^D} \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}_i^{RM}}, \text{ for } 1 \leq i \leq n\end{array}}$$

$$\textbf{RR-SCA4}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\left(\underline{\mathbf{pred}^{RM}}\right)\right] \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right]\left(\underline{\boldsymbol{p}_1^D},\ldots,\underline{\boldsymbol{p}_n^D}\right) \rightarrow \underline{\vartheta}\left(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\left(\underline{\mathbf{pred}^S}\right)\right]\left(\underline{\mathbf{X}_1^S},\ldots,\underline{\varphi}\left(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S}\right),\ldots,\underline{\mathbf{X}_n^S}\right)\right)\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right]\left(\underline{\boldsymbol{p}_1^{RM}},\ldots,\underline{\boldsymbol{p}_n^{RM}}\right) \rightarrow \underline{\vartheta}(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right](\underline{\mathbf{X}_1^S},\ldots,\underline{\varphi}(\underline{\mathbf{A}_1^S},\ldots,\underline{\mathbf{A}_n^S}),\ldots,\underline{\mathbf{X}_n^S})) \\ \underline{\mathbf{pred}^{RM}} \Rightarrow \underline{\mathbf{pred}^S}, \\ \mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}_i^D} \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}_i^{RM}}, \text{ for } 1 \leq i \leq n\end{array}}$$

A.2.4. rewritten-rules to rewrite ACAs

Consider the notation used to define the rules RR-PCAs. Also consider the following notation:

- γ is a variable that can be instantiated with one of the aggregation functions (sum, count, min, max, avg) used in SCAs.

- **sca** is a variable that can be instantiated with the name of the respective SCA asigned to an ACA.

The RR-ACAs are formed by eight rules as follows (the first six rules are to rewrite ACAs related to SCA of normalisation, while the last two rules are to rewrite ACAs related to SCA of group by):

$$\textbf{RR-ACA1}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\mathbf{A}^{RM}} \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\mathbf{A}^S}}{\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^S}}$$

$$\textbf{RR-ACA2}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\mathbf{G}^S}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\mathbf{G}^S}}$$

$$\textbf{RR-ACA3}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D}\{\underline{\boldsymbol{p}_1}, \underline{\boldsymbol{p}_2}, \ldots, \underline{\boldsymbol{p}_n}\} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{A}_1^{RM}}, \underline{\mathbf{A}_2^{RM}}, \ldots, \underline{\mathbf{A}_n^{RM}}\right) \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D}\{\underline{\boldsymbol{p}_1}, \underline{\boldsymbol{p}_2}, \ldots, \underline{\boldsymbol{p}_n}\} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{A}_1^S}, \underline{\mathbf{A}_2^S}, \ldots, \underline{\mathbf{A}_n^S}\right)\end{array}}{\underline{\mathbf{A}_i^{RM}} \Rightarrow \underline{\mathbf{A}_i^S}, \text{ for } 1 \leq i \leq n}$$

$$\textbf{RR-ACA4}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{A}_1^{RM}}, \underline{\mathbf{A}_2^{RM}}, \ldots, \underline{\mathbf{A}_n^{RM}}\right) \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{A}_1^S}, \underline{\mathbf{A}_2^S}, \ldots, \underline{\mathbf{A}_n^S}\right)\end{array}}{\underline{\mathbf{A}_i^{RM}} \Rightarrow \underline{\mathbf{A}_i^S}, \text{ for } 1 \leq i \leq n}$$

$$\textbf{RR-ACA5}: \frac{\begin{array}{l}\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{B}_1^{RM}}, \underline{\textbf{pred}_1^{RM}}\right); \ldots; \left(\underline{\mathbf{B}_{n-1}^{RM}}, \underline{\textbf{pred}_{n-1}^{RM}}\right); \underline{\mathbf{B}_n^{RM}} \Rightarrow \\ \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \left(\underline{\mathbf{B}_1^S}, \underline{\textbf{pred}_1^S}\right); \ldots; \left(\underline{\mathbf{B}_{n-1}^S}, \underline{\textbf{pred}_{n-1}^S}\right); \underline{\mathbf{B}_n^S}\end{array}}{\begin{array}{l}\underline{\mathbf{B}_i^{RM}} \Rightarrow \underline{\mathbf{B}_i^S}, \\ \underline{\textbf{pred}_i^{RM}} \Rightarrow \underline{\textbf{pred}_i^S}, \text{ for } 1 \leq i \leq n\end{array}}$$

$$\textbf{RR-ACA6}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\mathbf{G}^S}}{\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}^{RM}} \rightarrow \underline{\textbf{sca}}, \underline{\mathbf{G}^S}}$$

$$\textbf{RR-ACA7}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\gamma}\left(\underline{\mathbf{A}^{RM}}\right) \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\gamma}\left(\underline{\mathbf{A}^S}\right)}{\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^S}}$$

$$\textbf{RR-ACA8}: \frac{\mathrm{P}_{RM|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\gamma}\left(\underline{\mathbf{A}^{RM}}, \underline{\textbf{pred}^{RM}}\right) \Rightarrow \mathrm{P}_{\underline{S}|D}\left[\underline{\mathrm{C}^D}\right] \bullet \underline{\boldsymbol{p}^D} \rightarrow \underline{\textbf{sca}}, \underline{\gamma}\left(\underline{\mathbf{A}^S}, \underline{\textbf{pred}^S}\right)}{\begin{array}{l}\underline{\mathbf{A}^{RM}} \Rightarrow \underline{\mathbf{A}^S}, \\ \underline{\textbf{pred}^{RM}} \Rightarrow \underline{\textbf{pred}^S}\end{array}}$$

A.3. rewritten-rules to rewrite matching function signatures

Consider the following notation to describe the RR-MFs:

- **Y** is a variable that can be instantiated with a component pattern expression of forms: $\underline{\mathbf{S}[\mathrm{C}^S]\bullet\boldsymbol{p}^S}$ or $\underline{\varphi(\mathbf{S}[\mathrm{C}^S]\bullet\boldsymbol{p}^S)}$; being that the letter S in $\mathbf{Y}^S$ means that all elements into that expression belong to schema **S**.
- $\mathbf{S}_1$ and $\mathbf{S}_2$ are variables that can be instantiated with distinct schema names belonging to $\mathcal{L}$.
- C_i^D, for $1 \leq \mathrm{i} \leq \mathrm{n}$, are variable that can be instantiated with any class/relation of the schema **D**; mutatis mutandis to C_i^{RM} and C_i^S.
- $\boldsymbol{p}_i^{RM}$, for $1 \leq \mathrm{i} \leq \mathrm{n}$, are variables that can be instantiated with any property of a class/relation of the schema **RM**, mutatis mutandis to $\boldsymbol{p}_i^S$ and $\boldsymbol{p}_i^D$.
- τ and τ_i are variable that can be instantiated with any data type belonging to $\mathcal{T}$, being that the letter S in τ^S (and τ_i^S) means that the type was described in the schema **S**, mutatis mutandis to τ^{RM}, τ_i^{RM}, τ^D and τ_i^D.
- **E** is a variable that can be instantiated with a class/relation with a selection condition (notation: C(**pred**)) or without a selection condition (notation: C), being that the letter D in $\mathbf{E}^D$ means that all elements into that expression belong to schema **D**.

The RR-MFs are formed by seven rules defined as follows:

$$\textbf{RR-MF1}: \frac{\begin{array}{l}\textbf{match}: \left(\left(\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right], \underline{\tau^{RM}}\right) \times \left(\mathrm{D}\left[\underline{\mathbf{E}^D}\right], \underline{\tau^D}\right)\right) \rightarrow \mathrm{Boolean} \Rightarrow \\ \textbf{match}: \left(\left(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \underline{\tau^S}\right) \times \left(\mathrm{D}[\underline{\mathbf{E}^D}], \underline{\tau^D}\right)\right) \rightarrow \mathrm{Boolean}\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\ \underline{\tau^S} = \textbf{\textit{type}}(\underline{\mathrm{C}^S})\end{array}}$$

$$\textbf{RR-MF2}: \frac{\begin{array}{l}\textbf{match}: \left(\left(\mathrm{RM}\left[\underline{\mathrm{C}_1^{RM}}\right], \underline{\tau_1^{RM}}\right) \times \left(\mathrm{RM}\left[\underline{\mathrm{C}_2^{RM}}\right], \underline{\tau_2^{RM}}\right)\right) \rightarrow \mathrm{Boolean} \Rightarrow \\ \textbf{match}: \left(\left(\underline{\mathrm{S}_1}\left[\underline{\mathrm{C}_1^{S1}}\right], \underline{\tau_1^{S1}}\right) \times \left(\underline{\mathrm{S}_2}[\underline{\mathrm{C}_2^{S2}}], \underline{\tau_2^{S2}}\right)\right) \rightarrow \mathrm{Boolean}\end{array}}{\begin{array}{l}\mathrm{P}_{\underline{S1}|RM}\left[\underline{\mathrm{C}_1^{RM}}\right] \rightarrow \underline{\mathrm{S}_1}\left[\underline{\mathrm{C}_1^{S1}}\right], \\ \mathrm{P}_{\underline{S2}|RM}\left[\underline{\mathrm{C}_2^{RM}}\right] \rightarrow \underline{\mathrm{S}_2}\left[\underline{\mathrm{C}_2^{S2}}\right], \\ \underline{\tau_1^{S1}} = \textbf{\textit{type}}(\underline{\mathrm{C}_1^{S1}}), \\ \underline{\tau_2^{S2}} = \textbf{\textit{type}}(\underline{\mathrm{C}_2^{S2}})\end{array}}$$

$$\textbf{RR-MF3}: \begin{array}{l}\textbf{match}: \left(\left(\mathrm{D}\left[\underline{\mathbf{E}_1^D}\right], \underline{\tau_1^D}\right) \times \left(\mathrm{D}\left[\underline{\mathbf{E}_2^D}\right], \underline{\tau_2^D}\right)\right) \rightarrow \mathrm{Boolean} \Rightarrow \\ \textbf{match}: \left(\left(\mathrm{D}\left[\underline{\mathbf{E}_1^D}\right], \underline{\tau_1^D}\right) \times \left(\mathrm{D}\left[\underline{\mathbf{E}_2^D}\right], \underline{\tau_2^D}\right)\right) \rightarrow \mathrm{Boolean}\end{array}$$

RR-MF4 :

$$\begin{array}{l}
\textbf{match}: \left(\left(\mathrm{RM}\left[\underline{\mathrm{C}^{RM}}\right], \underline{\tau^{RM}}, \{\underline{\boldsymbol{p}_1^{RM}} : \underline{\tau_1^{RM}}, \ldots, \underline{\boldsymbol{p}_n^{RM}} : \underline{\tau_n^{RM}}\}\right) \times \right. \\
\left. \times \left(\mathrm{D}\left[\underline{\mathbf{E}^D}\right], \underline{\tau^D}, \{\underline{\boldsymbol{p}_1^D} : \underline{\tau_1^D}, \ldots, \underline{\boldsymbol{p}_n^D} : \underline{\tau_n^D}\}\right)\right) \rightarrow \text{Boolean} \Rightarrow \\
\textbf{match}: \left(\left(\underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \underline{\tau^S}, \{\underline{\boldsymbol{p}_1^S} : \underline{\tau_1^S}, \ldots, \underline{\boldsymbol{p}_n^S} : \underline{\tau_n^S}\}\right) \times \right. \\
\left. \times \left(\mathrm{D}[\underline{\mathbf{E}^D}], \underline{\tau^D}, \{\underline{\boldsymbol{p}_1^D} : \underline{\tau_1^D}, \ldots, \underline{\boldsymbol{p}_n^D} : \underline{\tau_n^D}\}\right)\right) \rightarrow \text{Boolean} \\
\hline
\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \rightarrow \underline{\mathrm{S}}\left[\underline{\mathrm{C}^S}\right], \\
\underline{\tau^S} = \textbf{\textit{type}}(\underline{\mathrm{C}^S}), \\
\mathrm{P}_{\underline{S}|RM}\left[\underline{\mathrm{C}^{RM}}\right] \bullet \underline{\boldsymbol{p}_i^{RM}} \rightarrow \underline{\mathbf{Y}_i^S} \text{ for } 1 \leq i \leq n, \\
\underline{\boldsymbol{p}_i^S} : \underline{\tau_i^S} \in \textbf{\textit{type}}(\mathrm{C}^S) \text{ for } 1 \leq i \leq n
\end{array}$$

RR-MF5 :

$$\begin{array}{l}
\textbf{match}: \left(\left(\mathrm{RM}\left[\underline{\mathrm{C}_1^{RM}}\right], \underline{\tau_1^{RM}}, \{\underline{\boldsymbol{p}_1^{RM}} : \underline{\tau_1^{RM}}, \ldots, \underline{\boldsymbol{p}_n^{RM}} : \underline{\tau_n^{RM}}\}\right) \times \right. \\
\left. \times \left(\mathrm{RM}\left[\underline{\mathrm{C}_2^{RM}}\right], \underline{\tau_2^{RM}}, \{\underline{\boldsymbol{p}_{21}^{RM}} : \underline{\tau_{21}^{RM}}, \ldots, \underline{\boldsymbol{p}_{2n}^{RM}} : \underline{\tau_{2n}^{RM}}\}\right)\right) \rightarrow \text{Boolean} \Rightarrow \\
\textbf{match}: \left(\left(\underline{\mathrm{S}_1}\left[\underline{\mathrm{C}_1^{S1}}\right], \underline{\tau_1^{S1}}, \{\underline{\boldsymbol{p}_1^{S1}} : \underline{\tau_1^{S1}}, \ldots, \underline{\boldsymbol{p}_n^{S1}} : \underline{\tau_n^{S1}}\}\right) \times \right. \\
\left. \times \left(\underline{\mathrm{S}_2}\left[\underline{\mathrm{C}_2^{S2}}\right], \underline{\tau_2^{S2}}, \{\underline{\boldsymbol{p}_1^{S2}} : \underline{\tau_1^{S2}}, \ldots, \underline{\boldsymbol{p}_n^{S2}} : \underline{\tau_n^{S2}}\}\right)\right) \rightarrow \text{Boolean} \\
\hline
\mathrm{P}_{\underline{S1}|RM}\left[\underline{\mathrm{C}_1^{RM}}\right] \rightarrow \underline{\mathrm{S}_1}\left[\underline{\mathrm{C}_1^{S1}}\right], \\
\underline{\tau_1^{S1}} = \textbf{\textit{type}}(\underline{\mathrm{C}_1^{S1}}), \\
\mathrm{P}_{\underline{S2}|RM}\left[\underline{\mathrm{C}_2^{RM}}\right] \rightarrow \underline{\mathrm{S}_2}\left[\underline{\mathrm{C}_2^{S2}}\right], \\
\underline{\tau_2^{S2}} = \textbf{\textit{type}}(\underline{\mathrm{C}_2^{S2}}), \\
\mathrm{P}_{\underline{S1}|RM}\left[\underline{\mathrm{C}_1^{RM}}\right] \bullet \underline{\boldsymbol{p}_i^{RM}} \rightarrow \underline{\mathbf{Y}_i^{S1}} \text{ for } 1 \leq i \leq n, \\
\mathrm{P}_{\underline{S2}|RM}\left[\underline{\mathrm{C}_2^{RM}}\right] \bullet \underline{\boldsymbol{p}_{2i}^{RM}} \rightarrow \underline{\mathbf{Y}_i^{S2}} \text{ for } 1 \leq i \leq n, \\
\underline{\boldsymbol{p}_i^{S1}} : \underline{\tau_i^{S1}} \in \textbf{\textit{type}}(\mathrm{C}_1^{S1}) \text{ for } 1 \leq i \leq n, \\
\underline{\boldsymbol{p}_i^{S2}} : \underline{\tau_i^{S2}} \in \textbf{\textit{type}}(\mathrm{C}_2^{S2}) \text{ for } 1 \leq i \leq n
\end{array}$$

RR-MF6 :

$$\begin{array}{l}
\textbf{match}: \left(\left(\mathrm{D}\left[\underline{\mathbf{E}_1^D}\right], \underline{\tau_1^D}, \{\underline{\boldsymbol{p}_1^D} : \underline{\tau_1^D}, \ldots, \underline{\boldsymbol{p}_n^D} : \underline{\tau_n^D}\}\right) \times \right. \\
\left. \times \left(\mathrm{D}\left[\underline{\mathbf{E}_2^D}\right], \underline{\tau_2^D}, \{\underline{\boldsymbol{p}_{21}^D} : \underline{\tau_{21}^D}, \ldots, \underline{\boldsymbol{p}_{2n}^D} : \underline{\tau_{2n}^D}\}\right)\right) \rightarrow \text{Boolean} \Rightarrow \\
\textbf{match}: \left(\left(\mathrm{D}\left[\underline{\mathbf{E}_1^D}\right], \underline{\tau_1^D}, \{\underline{\boldsymbol{p}_1^D} : \underline{\tau_1^D}, \ldots, \underline{\boldsymbol{p}_n^D} : \underline{\tau_n^D}\}\right) \times \right. \\
\left. \times \left(\mathrm{D}\left[\underline{\mathbf{E}_2^D}\right], \underline{\tau_2^D}, \{\underline{\boldsymbol{p}_{21}^D} : \underline{\tau_{21}^D}, \ldots, \underline{\boldsymbol{p}_{2n}^D} : \underline{\tau_{2n}^D}\}\right)\right) \rightarrow \text{Boolean}
\end{array}$$