

A Strategy to Revise the Constraints of the Mediated Schema

Marco A. Casanova¹, Tanara Lauschner¹, Luiz André P. Paes Leme¹,
Karin K. Breitman¹, Antonio L. Furtado¹, and Vânia M.P. Vidal²

¹Department of Informatics – PUC-Rio – Rio de Janeiro, RJ – Brazil

{casanova, tanara, lleme, karin, furtado}@inf.puc-rio.br

²Department of Computing, Federal University of Ceará – Fortaleza, CE – Brazil

vvidal@lia.ufc.br

Abstract. In this paper, we address the problem of changing the constraints of a mediated schema M to accommodate the constraints of a new export schema E_0 . We first show how to translate the constraints of E_0 to the vocabulary of M , creating a set of constraints C_0 in such a way that the schema mapping for E_0 is correct. Then, we show how to compute the new version of the constraints of M to accommodate C_0 so that all schema mappings, including that for E_0 , are correct. We solve both problems for subset and cardinality constraints and specific families of schema mappings.

Keywords: constraint revision, mediated schema, Description Logics.

1 Introduction

A *mediated environment* consists of a *mediated schema* M , several *export schemas* E_1, \dots, E_n , that describe data sources, and *schema mappings* $\gamma_1, \dots, \gamma_n$ such that γ_i defines (some of) the concepts of M in terms of the concepts of E_i , for each $i \in [1, n]$. To help define the mappings and maintain the constraints of M , we also introduce *import schemas* I_1, \dots, I_n such that I_i is the set of concepts of M that γ_i contains definitions for.

The constraints of the mediated schema are relevant for a correct understanding of what the semantics of the external schemas have in common. For example, consider a virtual store mediating access to online booksellers. Then, the class hierarchy of the mediated schema indicates what the booksellers' book classifications have in common; if the mediated schema enforces that all books must have ISBNs, then it means that all booksellers must have the same requirement; if it allows books with no (known) authors, then at least one bookseller must so allow; and so on.

We may break the process of adding a new export schema E_0 to the mediated environment into three steps. The *concept revision step* adjusts the vocabulary of M to perhaps include classes and properties originally defined in E_0 . The *mapping revision step* creates the local mapping γ_0 , and perhaps modifies the other mappings. The *import schema* I_0 comprises the set of concepts of M that γ_0 defines. Finally, the *constraint revision step* applies a minimum set of changes to the set of constraints of M to account for the set of constraints of E_0 .

One may have to iterate through these three steps since, in particular, revising the constraints of the mediated schema interacts with the definition of the schema mappings. For example, the local mapping γ_0 may have to be readjusted to preserve the class hierarchy of the mediated schema, or the class hierarchy of the mediated schema may have to be changed to reflect the class hierarchy of E_0 as seen through γ_0 [10].

In this paper, we are primarily concerned with the constraint revision step, with a bias to mediated environments in the context of the Web. Maintaining mediated environments in such context becomes a challenge because the number of data sources may be large and, moreover, the mediator does not have much control over the data sources, which may join or leave the mediated environment at will.

We break the constraint revision step in two sub-steps. Recall that the import schema I_0 is the set of concepts of M that γ_0 defines. The *constraint translation step* translates the constraints of E_0 to the concepts of I_0 , creating a set of constraints C_0 in such a way that γ_0 is correct with respect to C_0 . Intuitively, as a result of this step, we express the semantics of E_0 in terms of the concepts of M , which is the only schema that users have access to. The difficulty here lies in that γ_0 defines concepts of M in terms of the concepts of E_0 , whereas we need a mapping in the inverse direction to translate the constraints of E_0 to the concepts of M .

The *least constraint change step* applies a minimum set of changes to the constraints of M to accommodate C_0 in such a way that all schema mappings remain correct. This step intuitively means to harmonize the semantics of E_0 with the semantics of all export schemas previously added to the mediated environment, captured in the constraints of M . The key questions here are to precisely define what it means to apply a minimum set of changes to a set of constraints, and to guarantee that the mappings remain correct.

The contributions of this paper are twofold. First, for a family of conceptual schemas and schema mappings, we show how to perform constraint translation without actually computing the inverse mapping. We prove that, in some precise sense, the translation is the best possible. Second, to define how to change the constraints of the mediated schema, we introduce a lattice of sets of constraints and the notion of least upper bound of two sets of constraints. Again for the same family of conceptual schemas and schema mappings, we show how to compute the least upper bound that generates the revised set of constraints of the mediated schema.

Research in schema matching [2], as well as in ontology matching [8], tends to concentrate on vocabulary matching techniques, ignoring the question of constraint revision. Calvanese et al. [5] introduce a Description Logics framework, similar to that in Section 2, to address schema integration and query answering. Atzeni et al. [1] cover the traditional problem of rewriting a schema from one model to another, but they do not touch on the more complex problem of generating a new set of constraints that generalizes a pair of sets of constraints from different schemas, which we address in Section 4. Curino et al. [7] describe a software tool to support schema evolution that uses mapping invertibility. Fagin et al. [9] study mapping invertibility in the context of source-to-target tuple generating dependencies and formalize the notion of quasi-inverse. By contrast, we show in Section 3 how to generate the best possible set of subset and cardinality constraints without computing the inverse mapping.

This paper is organized as follows. Section 2 introduces an expressive family of conceptual schemas and a family of mappings. Section 3 focuses on constraint

translation. Section 4 discusses constraint lattices and shows how to generate the revised set of constraints of the mediated schema. Finally, Section 5 contains the conclusions.

We refer the reader to [6] for proofs for the results stated in Sections 3 and 4, comprehensive examples, and a detailed comparison with related work.

2 Basic Definitions

2.1 A Brief Review of Concepts from Description Logics

We adopt a family of *attributive languages* [4] defined as follows. A *language* \mathcal{L} in the family is characterized by an *alphabet* \mathcal{A} , consisting of a set of *atomic concepts*, a set of *atomic roles*, the *universal concept* and the *bottom concept*, denoted by \top and \perp , respectively, the *universal role* and the *bottom role*, also denoted by \top and \perp , respectively, and a set of *constants*.

The set of *role descriptions* of \mathcal{L} is inductively defined as

- An atomic role and the universal and bottom roles are role descriptions
- If p and q are role descriptions, then the following expression is a role description
 $p \circ q$ (the composition of p and q)

The set of *concept descriptions* of \mathcal{L} is inductively defined as

- An atomic concept and the universal and bottom concepts are concept descriptions
 - If a_1, \dots, a_n are constants, then $\{a_1, \dots, a_n\}$ is a concept description
 - If e and f are concept descriptions and p is a role description, then the following expressions are concept descriptions
- | | |
|-------------------------------------------------|-------------------------------------|
| $\neg e$ (negation) | |
| $e \sqcap f$ (intersection) | $e \sqcup f$ (union) |
| $\exists p.e$ (full existential quantification) | $\forall p.e$ (value restriction) |
| $(\leq n p)$ (at-most restriction) | $(\geq n p)$ (at-least restriction) |

Given an atomic concept A , a *restriction of A* is an intersection of the form $A \sqcap e$.

An *interpretation* s for \mathcal{L} consists of a nonempty set Δ^s , the *domain* of s , whose elements are called *individuals*, and an *interpretation function*, also denoted s , where:

- $s(\top) = \Delta^s$, when \top denotes the universal concept
- $s(\perp) = \tau$, when \perp denotes the bottom concept or the bottom role
- $s(A) \subseteq \Delta^s$, for each atomic concept A of \mathcal{L}
- $s(\top) = \Delta^s \times \Delta^s$, when \top denotes the universal role
- $s(P) \subseteq \Delta^s \times \Delta^s$, for each atomic role P of \mathcal{L}
- $s(a) \in \Delta^s$, for each constant a of \mathcal{L} , such that distinct constants denote distinct individuals (the *uniqueness assumption*)

The function s is extended to role and concept descriptions of \mathcal{L} as follows:

- $s(p \circ q)$ is the composition of $s(p)$ with $s(q)$
- $s(\{a_1, \dots, a_n\})$ is the set $\{s(a_1), \dots, s(a_n)\}$
- $s(\neg e)$ is the complement of $s(e)$ with respect to the domain Δ^s
- $s(e \sqcap f)$ is the intersection of $s(e)$ and $s(f)$

- $s(e \sqcup f)$ is the union of $s(e)$ and $s(f)$
- $s(\forall p.e)$ is the set of individuals that $s(p)$ relates only to individuals in $s(e)$, if any
- $s(\exists p.e)$ is the set of individuals that $s(p)$ relates to some individual in $s(e)$
- $s(\geq n p)$ is the set of individuals that $s(p)$ relates to at least n distinct individuals
- $s(\leq n p)$ is the set of individuals that $s(p)$ relates to at most n distinct individuals

A *formula* of \mathcal{L} is an expression of the form $u \beta v$, called an *inclusion*, or of the form $u \equiv v$, called an *equivalence*, where u and v are both concept descriptions or they are both role descriptions of \mathcal{L} . A *definition* is an equivalence of the form $T \equiv u$, where T is an atomic concept and u is a concept description, or T is an atomic role and u is a role description. An interpretation s for \mathcal{L} *satisfies* $u \sqsubseteq v$ iff $s(u) \subseteq s(v)$, and s *satisfies* $u \equiv v$ iff $s(u) = s(v)$.

In the rest of the paper, we will use the following notation:

- $s \models \sigma$ indicates that an interpretation s satisfies a formula σ
- $s \models \Sigma$ indicates that an interpretation s satisfies all formulas in a set of formulas Σ
- $\Sigma \models \sigma$ indicates that a set of formulas Σ *logically implies* a formula σ , that is, for any interpretation s , if $s \models \Sigma$, then $s \models \sigma$
- $\Sigma \models \Gamma$ indicates that a set of formulas Σ *logically implies* a set of formulas Γ , that is, for any interpretation s , if $s \models \Sigma$, then $s \models \Gamma$
- $Th(\Sigma)$ denotes the *theory induced* by Σ , which is the smallest set of formulas that contains Σ and is closed under logical implication.

Also, in Section 2.3, we will use concept and role descriptions over an alphabet \mathcal{A} which is the union of disjoint alphabets $\mathcal{A}_1, \dots, \mathcal{A}_n$. The syntax of concept and role descriptions remains the same. An interpretation s for \mathcal{A} is constructed from interpretations s_1, \dots, s_n for $\mathcal{A}_1, \dots, \mathcal{A}_n$ in the obvious way, except that we assume that

- (*Domain Disjointness Assumption*) Any pair of interpretations for \mathcal{A}_i and \mathcal{A}_j have disjoint domains, for each $i, j \in [1, n]$, with $i \neq j$

2.2 Extralite Schemas

We will work with *extralite schemas* [10] that, in OWL terminology [3], support *classes* and *properties*, and that admit *domain* and *range* constraints, *subset constraints*, *minCardinality* and *maxCardinality constraints*, with the usual meaning. Formally, an *extralite schema* is a pair $S = (\mathcal{A}, C)$ such that

- \mathcal{A} is an alphabet, called the *vocabulary* of S , whose atomic concepts and atomic roles are called the *classes* and *properties* of S , respectively
- C is a set of formulas, called the *constraints* of S , which must be of one the forms
 - *Domain Constraint:* $\exists P . \top \sqsubseteq D$ (property P has domain D)
 - *Range Constraint:* $\top \sqsubseteq \forall P . R$ (property P has range R)
 - *minCardinality constraint:* $D \sqsubseteq (\geq k P)$, where D is the domain of P (property P maps each individual in its domain D to at least k distinct individuals)
 - *maxCardinality constraint:* $D \sqsubseteq (\leq k P)$, where D is the domain of P (property P maps each individual in its domain D to at most k distinct individuals)
 - *Subset Constraint:* $C \sqsubseteq D$ (class C is a subclass of class D)
- C must have exactly one domain and one range constraint for each property in \mathcal{A}

Note that this formalization does not distinguish between object and datatype properties, in OWL terminology. The distinction will be visible in the examples, where the range of an object property will be a class defined in the schema, whereas the range of a datatype property will be a XML Schema type (i.e, a set of datatype values or *literals*). The formal development does not capture this distinction since the notion of domain does not separate individuals that denote class elements from individuals that correspond to datatype values. However, this formal liberality does not reduce the usefulness of the results in Sections 3 and 4.

We will use the terms *class*, *property*, *vocabulary* and *state* interchangeably with *atomic concept*, *atomic role*, *alphabet* and *interpretation*, respectively.

Example 1: Figures 1(a) and 1(c) show schemas for fragments of the Amazon and the eBay databases, using an informal notation. We use the namespace prefixes “a:” and “e:” to refer to the vocabularies of the Amazon and the eBay schemas.

In Figure 1(a), for example, a:title is defined as a (datatype) property with domain a:Product and range string (an XML Schema data type), a:Book is declared as a subclass of a:Product, and a:pub is defined as an (object) property with domain a:Book and range a:Publ. Although not indicated in Figure 1(a), we assume that all properties have maxCardinality equal to 1, except a:author, which is unbounded. Just to help illustrate the results in Section 3, we assume that a:pub has minCardinality equal to 2 and that a:name has minCardinality equal to 3.

Figures 1(b) and 1(d) formalize the constraints: the first column shows the domain and range constraints; the second column, the cardinality constraints; and the third column, the subset constraints. Note that there is no maxCardinality constraint for a:author, consistently with the fact that a book may have multiple authors. □

2.3 Mediated Environment

A mediated environment contains a mediated schema M , a mediated mapping γ and, for each $k=1, \dots, n$, an export schema E_k , an import schema I_k and a local mapping γ_k .

Assume that the classes and properties in M are C_1, \dots, C_u and P_1, \dots, P_v .

Import schemas help breaking the constraint revision problem into two sub-problems, as discussed in Sections 3 and 4. They are also a notational convenience to divide the definition of the mappings into two stages: the definition of the mediated mapping and the definition of the local mappings.

We restrict the import schemas as follows:

- for $k=1, \dots, n$, the vocabulary of I_k is a subset of the vocabulary of M

We do not adopt namespace prefixes, as in the examples, but a more abstract notation to distinguish the occurrence of a symbol in the vocabulary of M from the occurrence of the same symbol in the vocabulary of I_k . For each class C_i (or property P_j) in the vocabulary of M , we denote the occurrence of C_i (or P_j) in the vocabulary of I_k by C_i^k (or P_j^k); we also say that C_i^k (or P_j^k) matches C_i (or P_j).

The mediated mapping γ defines the classes and properties of M as unions of classes and properties from the import schemas so that it becomes a simple task to revise it when an import schema is added or removed. Most of the complexity is

| | |
|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| a:Product a:title range string a:price range decimal a:currency range string | a:Publ a:name range string a:address range string |
| a:Book a:isbn range string a:author range string a:pub range a:Publ | a:Book is-a a:Product a:Music is-a a:Product a:Video is-a a:Product a:PC-HW is-a a:Product |

Fig. 1(a). Informal definition of the Amazon schema

| | | |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\exists a:\text{title} . \top \sqsubseteq a:\text{Product}$ $\top \sqsubseteq \forall a:\text{title} . \text{string}$... | $a:\text{Product} \sqsubseteq (\leq 1 a:\text{title})$ $a:\text{Product} \sqsubseteq (\leq 1 a:\text{price})$ $a:\text{Product} \sqsubseteq (\leq 1 a:\text{currency})$ | $a:\text{Book} \sqsubseteq a:\text{Product}$ $a:\text{Music} \sqsubseteq a:\text{Product}$ $a:\text{Video} \sqsubseteq a:\text{Product}$ $a:\text{PC-HW} \sqsubseteq a:\text{Product}$ |
| $\exists a:\text{pub} . \top \sqsubseteq a:\text{Book}$ $\top \sqsubseteq \forall a:\text{pub} . a:\text{Publ}$... | $a:\text{Book} \sqsubseteq (\leq 1 a:\text{isbn})$ $a:\text{Book} \sqsubseteq (\geq 2 a:\text{pub})$ | |
| $\exists a:\text{name} . \top \sqsubseteq a:\text{Publ}$ $\top \sqsubseteq \forall a:\text{name} . \text{string}$... | $a:\text{Publ} \sqsubseteq (\geq 3 a:\text{name})$ $a:\text{Publ} \sqsubseteq (\leq 1 a:\text{address})$ | |

Fig. 1(b). Formal definition of (some of) the constraints of the Amazon schema

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| e:Seller e:name range string | e:Product e:type range string |
| e:Offer e:qty range integer e:price range double e:currency range string e:seller range e:Seller e:product range e:Product | e:ean range integer e:title range string e:author range string e:edition range integer e:year range integer e:pub range string |

Fig. 1(c). Informal definition of the eBay schema

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| $\exists e:\text{name} . \top \sqsubseteq e:\text{Seller}$ $\top \sqsubseteq \forall e:\text{name} . \text{string}$... | $e:\text{Seller} \sqsubseteq (\leq 1 e:\text{name})$ $e:\text{Offer} \sqsubseteq (\leq 1 e:\text{qty})$ $e:\text{Offer} \sqsubseteq (\leq 1 e:\text{price})$... | (no subset constraints) |
| $\exists e:\text{seller} . \top \sqsubseteq e:\text{Offer}$ $\top \sqsubseteq \forall e:\text{seller} . e:\text{Seller}$ | $e:\text{Product} \sqsubseteq (\leq 1 e:\text{type})$ | |
| $\exists e:\text{product} . \top \sqsubseteq e:\text{Offer}$ $\top \sqsubseteq \forall e:\text{product} . e:\text{Product}$... | $e:\text{Product} \sqsubseteq (\leq 1 e:\text{ean})$ $e:\text{Product} \sqsubseteq (\leq 1 e:\text{title})$... | |

Fig. 1(d). Formal definition of (some of) the constraints of the eBay schema

therefore isolated in the local mappings. This restriction reflects the idea that, in the context of the Web, data sources are independent.

More precisely, we restrict the mediated mapping as follows:

- for each $i=1, \dots, u$, the mapping γ contains a definition of the form

$$C_i \equiv e_i^1 \sqcup \dots \sqcup e_i^n \quad (1)$$

where e_i^k is the class C_i^k of I_k that matches C_i , if it exists, or the bottom concept \perp , otherwise, for each $k=1, \dots, n$

- for each $j=1, \dots, v$, the mapping γ contains a definition of the form

$$P_j \equiv p_j^1 \sqcup \dots \sqcup p_j^n \quad (2)$$

where p_j^k is the property P_j^k of I_k that matches P_j , if it exists, or the bottom role \perp , for each $k=1, \dots, n$

We use \perp just as a notational convenience so that Equations (1) and (2) have exactly one concept description (or role description) from each import schema.

For each $k=1, \dots, n$, the local mapping γ_k defines the classes and properties of I_k in the terms of the vocabulary of the export schema E_k . We restrict γ_k as follows:

- for each class C_i^k of I_k , the local mapping γ_k contains a definition of the form

$$C_i^k \equiv \rho_i^k \quad (3)$$

where ρ_i^k is a concept description over the vocabulary of E_k

- for each property P_j^k of I_k , the local mapping γ_k contains a definition of the form

$$P_j^k \equiv \pi_j^k \quad (4)$$

where π_j^k is a role description over the vocabulary of E_k

We introduce $\bar{\gamma}_k$ as the *function induced by γ_k* , defined as the function from states of E_k into states of I_k such that, for each state s of E_k , $\bar{\gamma}_k(s) = r$ iff

- $r(C_i^k) = s(\rho_i^k)$, if $C_i^k \equiv \rho_i^k$ is the definition for class C_i^k in γ_k
- $r(P_j^k) = s(\pi_j^k)$, if $P_j^k \equiv \pi_j^k$ is the definition for property P_j^k in γ_k

Likewise, we introduce $\bar{\gamma}$ as the *function induced by the mediated mapping γ* and the local mapping $\gamma_1, \dots, \gamma_n$ as the mapping from states of E_1, \dots, E_n into states of M such that, for states s_1, \dots, s_n of E_1, \dots, E_n , $\bar{\gamma}(s_1, \dots, s_n) = r$ iff, for $i=1, \dots, u$ and $j=1, \dots, v$

- $r(C_i) = s_1(e_i^1) \cup \dots \cup s_n(e_i^n)$, if $C_i \equiv e_i^1 \sqcup \dots \sqcup e_i^n$ is the definition of C_i in γ
- $r(P_j) = s_1(p_j^1) \cup \dots \cup s_n(p_j^n)$, if $P_j \equiv p_j^1 \sqcup \dots \sqcup p_j^n$ is the definition of P_j in γ

Example 2: Figure 2 describes a mediated environment that contains:

- the mediated schema `sales`, shown in Figure 2(a), with namespace prefix “s:” and the constraints shown in Figure 2(b); in particular, the `minCardinality` constraint for `s:Book` follows from the remarks in Example 3
- the Amazon and the eBay schemas, shown in Figure 1, as export schemas
- the import schema for the Amazon export schema (not shown in Figure 2), with the same classes and properties as `sales`, but prefixed with “ai:”
- the import schema for the eBay export schema (not shown in Figure 2), with the same classes and properties as `sales`, but prefixed with “ei:”
- the mediated mapping shown in Figure 2(c)
- the local mapping, shown in Figure 2(d), defining the classes and properties of the Amazon import schema in terms of its export schema; in particular, `ai:pub` is defined as the composition of `a:pub` with `a:name`

- the local mapping, shown in Figure 2(e), defining the classes and properties of the eBay import schema in terms of its export schema; in particular, `ei:Music` and `ei:Book` are defined as restrictions of `e:Product`. \square

| | |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <code>s:Product</code> <code>s:title</code> range string <code>s:Book</code> <code>s:pub</code> range string | <code>s:Book</code> is-a <code>s:Product</code> <code>s:Music</code> is-a <code>s:Product</code> |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|

Fig. 2(a). The Sales mediated schema

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| $\exists s:title. \top \beta s:Product$ $\top \beta \forall ai:title.string$ $\exists s:pub. \top \beta s:Book$ $\top \beta \forall ai:pub.string$ | <code>s:Product</code> $\beta (\leq 1 s:title)$ <code>s:Book</code> $\beta (\geq 6 s:pub)$ | <code>s:Book</code> $\beta s:Product$ <code>s:Music</code> $\beta s:Product$ |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|

Fig. 2(b). Constraints of the Sales mediated schema

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| <code>s:Product</code> $\equiv ai:Product \uparrow ei:Product$ <code>s:Music</code> $\equiv ai:Music \uparrow ei:Music$ <code>s:Book</code> $\equiv ai:Book \uparrow ei:Book$ | <code>s:title</code> $\equiv ai:title \uparrow ei:title$ <code>s:pub</code> $\equiv ai:pub \uparrow ei:pub$ |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|

Fig. 2(c). Mediated schema mapping

| | |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <code>ai:Product</code> $\equiv a:Product$ <code>ai:Music</code> $\equiv a:Music$ <code>ai:Book</code> $\equiv a:Book$ | <code>ai:title</code> $\equiv a:title$ <code>ai:pub</code> $\equiv a:pub) a:name$ |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|

Fig. 2(d). Local schema mappings from the Amazon export schema to its import schema

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <code>ei:Product</code> $\equiv e:Product$ <code>ei:Music</code> $\equiv e:Product \uparrow \exists e:type.\{ 'music' \}$ <code>ei:Book</code> $\equiv e:Product \uparrow \exists e:type.\{ 'book' \}$ | <code>ei:title</code> $\equiv e:title$ <code>ei:pub</code> $\equiv e:pub$ |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|

Fig. 2(e). Local schema mappings from the eBay export schema to its import schema

3 Constraint Translation

Consider a mediated environment with a mediated schema M , a mediated mapping γ and, for each $k=1,\dots,n$, an export schema E_k , an import schema I_k and a local mapping γ_k . By *constraint translation* we mean the problem of translating the constraints of E_k to the vocabulary of I_k , creating the set of constraints IC_k in such a way that γ_k induces a mapping from consistent states of E_k into consistent states of I_k .

To motivate the discussion, we start with an example.

Example 3: We first observe that the definitions in a local mapping are adequate to translate queries over the import schema (and hence the mediated schema) into queries over the export schema. They also help translating constraints of the import schema into constraints of the export schema. For example, suppose that

$$\exists ai:pub. \top \sqsubseteq ai:Book \quad (5)$$

is a constraint of the Amazon import schema. Using the definitions in Figure 2(d), we may translate the constraint in (5) to the Amazon export schema by replacing $ai:pub$ by $a:pub) a:name$ and $ai:Book$ by $a:Book$, obtaining

$$\exists (a:pub) a:name). \top \sqsubseteq a:Book \quad (6)$$

However, the constraint translation problem is in the opposite direction: how to express the constraints of the Amazon export schema in terms of the vocabulary of its import schema, thereby eventually exposing the semantics of the Amazon export schema to the users.

Figure 3(a) contains the translation of the constraints of the Amazon export schema, shown in Figure 1(b), to the corresponding import schema, in view of the local mapping defined in Figure 2(d). In particular, recall from Figure 2(d) that $ai:pub \equiv a:pub) a:name$ and $ai:Book \equiv a:Book$. This has several consequences. First, the domain and range of $ai:pub$ are $ai:Book$ and $string$. Second, $ai:pub$ has $minCardinality$ 6 with respect to $ai:Book$ since, observing Figure 1(b), $a:pub$ has $minCardinality$ 2 with respect to $a:Book$ and $a:name$ has $minCardinality$ 3 with respect to $a:Publ$. The other constraints follow directly from those of the Amazon export schema, since each of the other classes and properties of the import schema are defined in terms of a single class or property of the Amazon export schema.

Figure 3(b) contains the translation of the constraints of the eBay export schema, shown in Figure 1(c), to the corresponding import schema, in view of the local mapping defined in Figure 2(e). In particular, recall from Figure 2(e) that $ei:Music$ and $ei:Book$ are defined as restrictions of $e:Product$. As a consequence, we have the two subset constraints shown on the third column of Figure 3(b). Note that the original eBay schema has no subset constraints (see Figure 1(d)). \square

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| $\exists ai:title. \top \sqsubseteq ai:Product$ $\top \sqsubseteq \forall ai:title.string$ $\exists ai:pub. \top \sqsubseteq ai:Book$ $\top \sqsubseteq \forall ai:pub.string$ | $ai:Product \sqsubseteq (\leq 1 ai:title)$ $ai:Book \sqsubseteq (\geq 6 ai:pub)$ | $ai:Book \sqsubseteq ai:Product$ $ai:Music \sqsubseteq ai:Product$ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------|

Fig. 3(a). Constraints of the import schema for the Amazon (export) schema

| | | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| $\exists ei:title. \top \sqsubseteq ei:Product$ $\top \sqsubseteq \forall ei:title.string$ $\exists ei:pub. \top \sqsubseteq ei:Book$ $\top \sqsubseteq \forall ei:pub.string$ | $ei:Product \sqsubseteq (\leq 1 ei:title)$ $ei:Book \sqsubseteq (\leq 1 ei:pub)$ | $ei:Book \sqsubseteq ei:Product$ $ei:Music \sqsubseteq ei:Product$ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------|

Fig. 3(b). Constraints of the import schema for the eBay (export) schema

In what follows, we formalize and generalize the arguments outlined in Example 3, indicating how to translate subset constraints and cardinality constraints separately.

Definition 1: Let EC_k be the set of constraints of E_k . The *translation* of the subset constraints in EC_k for γ_k is the set Σ_k of all subset constraints $C \sqsubseteq D$ such that γ_k has definitions for C and D of the form $C \equiv \rho_C$ and $D \equiv \rho_D$ and $EC_k \models \rho_C \sqsubseteq \rho_D$.

The translation of cardinality constraints, in Definition 2, follows from Proposition 1, which captures simple facts about such constraints. For example, if $a:\text{pub}$ has minCardinality 2 with respect to $a:\text{Book}$, then trivially $a:\text{pub}$ has minCardinality 1 with respect to $a:\text{Book}$; likewise, if $a:\text{isbn}$ has maxCardinality 1 with respect to $a:\text{Book}$, then trivially $a:\text{isbn}$ has maxCardinality 2 with respect to $a:\text{Book}$.

To improve readability, we use $\text{min}[k,P]$ to abbreviate the minCardinality constraint $D\beta (\geq k P)$, and $\text{max}[k,P]$ for the maxCardinality constraint $D\beta (\leq k P)$, where D is implicit from the domain constraint for P .

Proposition 1: For any property P of E_k , we have:

- (i) $EC_k \models \text{min}[h,P]$ iff there is $\text{min}[g,P]$ in EC_k such that $h \leq g$.
- (ii) $EC_k \models \text{max}[h,P]$ iff there is $\text{max}[g,P]$ in EC_k such that $h \geq g$.

Definition 2: Let EC_k be the set of constraints of E_k . The *translation* of the cardinality constraints in EC_k for γ_k is the set κ_k defined as follows. For each property P in I_k , if the definition of P in γ_k is $P \equiv P_1 \circ P_2 \circ \dots \circ P_r$, then

- (i) If $\text{min}[g_i^j, P_i]$ are all the minCardinality constraints in EC_k for property P_i , for $i \in [1, r]$ and $j \in [1, s_i]$, then κ_k contains a minCardinality constraint of the form $\text{min}[g, P]$, with $g = \prod_{i=1}^r \text{max}(\{g_i^j / j = 1, \dots, s_i\})$. If EC_k has no minCardinality constraints for P_i , then so does κ_k .
- (ii) If $\text{max}[h_i^j, P_i]$ are all the maxCardinality constraints in EC_k for property P_i , for $i \in [1, r]$ and $j \in [1, t_i]$, then κ_k contains a maxCardinality constraint of the form $\text{max}[h, P]$, with $h = \prod_{i=1}^r \text{min}(\{h_i^j / j = 1, \dots, t_i\})$. If EC_k has no maxCardinality constraints for P_i , then so does κ_k .

We are now ready to combine Definitions 1 and 2 to indicate how to translate the subset and cardinality constraints of E_k into constraints of I_k .

Definition 3: Let EC_k be the set of constraints of E_k . The *translation* of the subset and cardinality constraints in EC_k for γ_k is the set $IC_k = \Sigma_k \cup \kappa_k$.

We establish, in Proposition 2, that IC_k is a correct translation of the subset and cardinality constraints of E_k with respect to γ_k . Then, we state, in Proposition 3, that IC_k is the largest theory of subset and cardinality constraints such that γ_k induces a mapping from consistent states of E_k into states of I_k that satisfy the theory. Thus, Definition 3 indicates the best possible translation for the subset and cardinality constraints of E_k to the vocabulary of I_k .

Proposition 2: γ_k induces a mapping from consistent states of E_k into states of I_k that satisfy IC_k .

Proposition 3: Let Φ be any set of subset and cardinality constraints such that γ_k has definitions for their classes and properties. Suppose that γ_k induces a mapping from consistent states of E_k into states of I_k that satisfy Φ . Then, $\Phi \subseteq IC_k$.

In summary, Definitions 1, 2 and 3 indicate, for the families of conceptual schemas and schema mappings introduced in Section 2, how to translate subset and cardinality

constraint without computing inverse mappings. Propositions 2 and 3 assert that the translation is correct and the best possible. We refer the reader to [6] for the translation of domain and range constraints.

4 Constraint Revision

Consider again a mediated environment with a mediated schema M , a mediated mapping γ and, for each $k=1, \dots, n$, an export schema E_k , an import schema I_k and a local mapping γ_k . Assume that V and MC are the vocabulary and the set of constraints of M .

If we take import schemas into account, we may refine the steps required to add a new export schema E_0 to the mediated environment as follows:

1. (*Concept revision step*) Create the revised vocabulary V_r of the mediated schema, perhaps by including in V classes and properties originally defined in E_0 , and define the import schema I_0 for E_0 .
2. (*Mapping revision step*) Create the revised mediated mapping γ_r , and define the local mapping γ_0 between I_0 and E_0 .
3. (*Constraint revision step*) Create the revised set of constraints MC_r by computing the set IC_0 of constraints of I_0 , and applying a minimum set of changes to MC to account for IC_0 .

In this section, we assume that the first two steps have already been performed, resulting in the revised vocabulary V_r , the revised mediated mapping γ_r , and the definitions of the import schema I_0 for E_0 and the local mapping γ_0 between I_0 and E_0 . In particular, note that γ_r must be a set of definitions as in Equations (1) and (2). We also assume that the set IC_0 of constraints of I_0 have already been computed, as discussed in Section 3. We focus on how to create the revised set of constraints. The reader should bear in mind the notation just introduced, which will be used in what follows.

There are two questions here: (1) what it means to apply a minimum set of changes to a set of constraints; (2) how to maintain the correctness of the schema mappings. To address the first question, we introduce a lattice of sets of constraints. The second question then follows from a property of the lattice.

Recall from Section 2.1, that $Th(\Phi)$ denotes the theory induced by a set of formulas Φ . Let \mathcal{T} be the set of all sets of constraints. Then, (\mathcal{T}, \sim) is a lattice where, given any two sets of constraints, Φ_1 and Φ_2 , their greatest lower bound (g.l.b.) is $\Phi_1 \Delta \Phi_2 = Th(\Phi_1) \cup Th(\Phi_2)$ and their least upper bound (l.u.b.) is $\Phi_1 \nabla \Phi_2 = Th(\Phi_1) \cap Th(\Phi_2)$. Note that $\Phi_i \models \Phi_1 \nabla \Phi_2$ and $\Phi_1 \Delta \Phi_2 \models \Phi_i$ for $i=1,2$.

We argue that MC_r can be taken as the l.u.b. of MC and the translation of IC_0 to V_r .

Definition 4: The *translation* of IC_0 to V_r is the set of constraints C_0 defined as follows: for each β in IC_0 , the set C_0 contains β' constructed by replacing in β each class C_i^0 , of the vocabulary of I_0 , by C_i , the class of V_r that C_i^0 matches, and each property P_j^0 , of the vocabulary of I_0 , by P_j , the property of V_r that P_j^0 matches.

We now give a simple example that partially illustrates the constraint revision step.

Example 5: Consider the *Sales* mediated schema shown in Figure 2. Let *BN* be a new export schema (say, a fragment of the Barnes&Noble database), shown in Figures 4(a) and (b). To include *BN* in the mediated environment, we perform three steps:

(*Concept revision step*). Assume that the vocabulary of *Sales* is not changed and that the import schema for *BN* has classes *bi:Book*, *bi:Music* and *bi:Product*, and properties *bi:title* and *bi:pub*.

(*Mapping revision step*) Figures 4(c) and (d) show the revised mediated mapping and the local schema mapping from the *BN* export schema to its import schema.

(*Constraint revision step.*) According to the discussion in Section 3, the import schema for *BN* has only two subset constraints, σ_1 and σ_2 , where

$$\sigma_1: \text{bi:Book} \sqsubseteq \text{bi:Product} \qquad \sigma_2: \text{bi:Music} \sqsubseteq \text{bi:Product}$$

This follows from Definition 1, using the subset constraints of *BN* (Figure 4(b)) and the local mapping between *BN* and its import schema (Figure 4(d)). Also note that *b:CultProd* is not in the vocabulary of the import schema for *BN*.

Using Definition 4, we translate σ_1 and σ_2 to the vocabulary of *Sales*, replacing *bi:Book* by *s:Book*, *bi:Music* by *s:Music* and *bi:Product* by *s:Product*. This results in τ_1 and τ_2 , where

$$\tau_1: \text{s:Book} \sqsubseteq \text{s:Product} \qquad \tau_2: \text{s:Music} \sqsubseteq \text{s:Product}$$

Let *SC* be the set of constraints of *Sales* (Figure 2(b)). Let *C* be the set of constraints of the import schema of *BN*, after translation to the vocabulary of *Sales*. The revised set of constraints of the mediated schema, $SC_r = SC \sqcap C$, is such that: (1) SC_r contains τ_1 and τ_2 (just as *SC*), since τ_1 and τ_2 are in both *SC* and *C*; (2) SC_r has no cardinality constraints (unlike *SC*), since *C* has no cardinality constraints (by the middle column of Figure 4(b), *BN* has no cardinality constraints). Thus, adding *BN* to the mediated environment affects the constraints of *Sales*. \square

We are now ready to argue that MC_r can be taken as the l.u.b. of *MC* and C_0 .

Proposition 4: Let $MC_r = MC \sqcap C_0$. Assume that:

- (i) The mediated mapping γ and the local mapping $\gamma_1, \dots, \gamma_n$ induce a mapping from consistent states of E_1, \dots, E_n into consistent states of *M*.
- (ii) The local mapping γ_0 induces a mapping from consistent states of E_0 into consistent states of I_0 .

Then, the revised mediated mapping γ_r and the local mappings $\gamma_0, \gamma_1, \dots, \gamma_n$ induce a mapping from consistent states of EC_0, EC_1, \dots, EC_n into states of the revised mediated schema that satisfy MC_r .

The proof of Proposition 4 depends on assuming that γ_r defines the classes and properties of V_r as unions of classes and properties from the vocabularies of the import schemas. Since $MC_r = MC \sqcap C_0$, with respect to (\mathcal{T}, \models) , we may consider that MC_r is the least way to revise *MC* and yet retain correctness of the mappings, in view of Proposition 4.

The solution to the least constraint revision problem outlined up to this point gives no indication on how to select a finite set of constraints that generates $MC \sqcap C_0$. In the rest of this section, we therefore show how to compute the subset and cardinality constraints in the l.u.b. of two sets of constraints.

| | |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------|
| b:Product b:title range string b:Book b:pub range string | b:CultProd is-a b:Product b:Music is-a b:CultProd b:Book is-a b:CultProd |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------|

Fig. 4(a) The new export schema BN to be added to the mediated environment

| | | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------|
| $\exists b:\text{title}.\top \sqsubseteq b:\text{Product}$ $\sqsubseteq \forall b:\text{title}.\text{string}$ $\exists b:\text{pub}.\top \sqsubseteq b:\text{Book}$ $\sqsubseteq \sqsubseteq \forall b:\text{pub}.\text{string}$ | (no cardinality constraints) | b:Book \sqsubseteq b:CultProd b:Music \sqsubseteq b:CultProd b:CultProd \sqsubseteq b:Product |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------|

Fig. 4(b). Constraints of the new export schema BN

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| s:Product \equiv bi:Product \sqcup ai:Product \sqcup ei:Product s:Music \equiv bi:Music \sqcup ai:Music \sqcup ei:Music s:Book \equiv bi:Book \sqcup ai:Book \sqcup ei:Book | s:title \equiv bi:title \sqcup ai:title \sqcup ei:title s:pub \equiv bi:pub \sqcup ai:pub \sqcup ei:pub |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|

Fig. 4(c). Revised Mediated mapping of the mediated environment

| | |
|---------------------------------------------------------------------------------------|----------------------------------------------------|
| bi:Product \equiv b:Product bi:Music \equiv b:Music bi:Book \equiv b:Book | bi:title \equiv b:title bi:pub \equiv b:pub |
|---------------------------------------------------------------------------------------|----------------------------------------------------|

Fig. 4(d). Local schema mappings from the BN export schema to its import schema

Proposition 5 provides a simple way to compute the subset constraints that are logical consequences of a set of such constraint.

Proposition 5 (Subset Constraint Chaining): Let $\sigma_1, \dots, \sigma_n$ be a sequence of subset constraints. Suppose that, for each $i \in [1, n]$, σ_i is of the form $A_i \sqsubseteq A_{i+1}$. Then, we have that $\sigma_1, \dots, \sigma_n \models \sigma$, where σ is the subset constraint $A_1 \sqsubseteq A_{n+1}$.

We say that $\sigma_1, \dots, \sigma_n$, with the characteristics listed in Proposition 5, is a *chain of subset constraints connecting A_1 to A_{n+1}* , and that σ is the result of *chaining $\sigma_1, \dots, \sigma_n$* .

The final result shows how to compute the subset and cardinality constraints in the l.u.b. of two sets of constraints.

Proposition 6: Let Γ_1 and Γ_2 be two sets of constraints. Construct set Γ as follows:

- (i) Let $A \sqsubseteq B$ be the result of chaining a sequence of subset constraints from Γ_1 , as well as the result of chaining a sequence of subset constraints from Γ_2 . Then, $A \sqsubseteq B$ is in Γ .
- (ii) For each property P , let $\min[g_i^j, P]$ be the minCardinality constraints in Γ_i for P , for $i=1,2$ and $j \in [1, s_i]$. Then, $\min[g, P]$ is in Γ , where $g = \min(\{g_1, g_2\})$ and $g_i = \max(\{g_i^j / j = 1, \dots, s_i\})$, for $i=1,2$. If either Γ_1 or Γ_2 have no min-Cardinality constraints for P , then so does Γ .
- (iii) For each property P , let $\max[h_i^j, P]$ be the maxCardinality constraints in Γ_i for P , for $i=1,2$ and $j \in [1, t_i]$. Then, $\max[h, P]$ is in Γ , where $h = \max(\{h_1, h_2\})$

and $h_i = \min(\{h_i^j / j = 1, \dots, t_i\})$, for $i=1,2$. If either Γ_1 or Γ_2 have no max-Cardinality constraints for P , then so does Γ .

Then, Γ is the set of all subset and cardinality constraints in $\Gamma_1 \nabla \Gamma_2$.

We refer the reader to [6] for a complete account of all families of constraints introduced in Section 2.2, including the domain and range constraints, which were omitted from the discussion for brevity.

In summary, computing the cardinality and subset constraints in the l.u.b. of two sets of constraints can be broken into computing the subset constraints in the l.u.b., which is straightforward by Chaining (Proposition 6 (i)), and computing the cardinality constraints in the l.u.b. (Proposition 6 (ii-iii)). This apparently simple fact is not necessarily true when other families of constraints are considered.

5 Conclusions

For the families of schemas and mappings defined in Section 2, we showed in Section 3 how to translate subset and cardinality constraints of the export schema to the import schema without computing inverse mappings. This problem reoccurs in other situations, such as how to express view constraints. The difficulty of the problem lies in that the definitions are in the inverse direction, as illustrated in Example 3.

To address the least constraint revision problem, we first introduced a lattice of sets of constraints. Then, again for the families of schemas and mappings defined in Section 2, we showed in Section 4 how to generate the subset and cardinality constraints of the revised set of constraints of the mediated schema.

Extending the results of this paper to domain and range constraints is fairly simple, but omitted here for brevity (see [6]). As future work, we are investigating families of constraints that include keys and disjointness constraints, which is a more difficult question, since disjointness and subset constraints may lead to inconsistencies.

References

- [1] Atzeni, P., Cappellari, P., Torlone, R., Bernstein, P.A., Gianforme, G.: Model-independent schema translation. *The VLDB Journal* 17(6), 1347–1370 (2008)
- [2] Bernstein, P., Melnik, S.: Model management 2.0: manipulating richer mappings. In: *Proc. 27th ACM SIGMOD Int'l. Conf. Management of Data, Beijing, China*, pp. 1–12 (2007)
- [3] Breitman, K., Casanova, M., Truszkowski, W.: *Semantic web: concepts, technologies, and applications*. Springer, London (2007)
- [4] Calvanese, D., Lenzerini, M., Nardi, D.: Description Logics for Conceptual data modeling. In: Chomicki, J., Saake, G. (eds.) *Logics for Databases and Information Systems*, Kluwer Academic Publishers, Dordrecht (1998)
- [5] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R., Ruzzi, M.: Data Integration through DL-Lite-A Ontologies. In: *Proc. 3rd Int'l. Workshop on Semantics in Data and Knowledge Bases*, pp. 26–47 (2008)

- [6] Casanova, M.A., Lauschner, T., Paes Leme, L.A., Breitman, K.K., Furtado, A.L.: A Strategy to Revise the Constraints of the Mediated Schema. Technical Report MCC34/09, Department of Informatics, PUC-Rio (April 2009)
- [7] Curino, C.A., Moon, H.J., Zaniolo, C.: Graceful database schema evolution: the PRISM workbench. *Proc. VLDB Endowment* 1(1), 761–772 (2008)
- [8] Euzenat, J., Shvaiko, P.: *Ontology matching*. Springer, Heidelberg (2007)
- [9] Fagin, R., Kolaitis, P.G., Popa, L., Tan, W.-C.: Quasi-inverses of schema mappings. In: *Proc. 26th ACM SIGMOD Symp. on Principles of Database Systems*, pp. 123–132.
- [10] Leme, L.A.P., Casanova, M.A., Breitman, K.K., Furtado, A.L.: Instance-based OWL Schema Matching. In: *Proc. 11th Int'l. Conf. on Enterprise Inf. Systems*, Milan, Italy (2009)