Non-destructive Integration of Form-based Views

Jan Hidders[1], Jan Paredaens[1], Philippe Thiran[2], Geert-Jan Houben[2], and Kees van Hee[2]

[1] University of Antwerp, Belgium
[2] Eindhoven University of Technology, The Netherlands

Abstract. Form documents or screen forms bring essential information on the data manipulated by an organization. They can be considered as different but often overlapping views of its whole data. This paper presents a non-destructive approach of their integration. The main idea of our approach is to keep the original views intact and to specify constraints between overlapping structures. For reasoning over constraints, we provide a set of inference rules that allows not only to infer implied constraints but also to detect conflicts. These reasoning rules are proved to be sound and complete. Although the form-based views are hierarchical structures, our constraints and reasoning rules can also be used in non-hierarchical data models.

1 Introduction

In the design process for data-intensive applications the design of the global data model is a crucial step. Often this step involves the integration of different data models that each describe the information need of different groups of end users. In the case of workflow and case management systems these data models or views are usually defined as a form, hence *form-based views* and the tasks that are managed by the system are typically manipulations of these forms. For large and complex workflows the task of modeling the forms is often split according to the different case types. The consequence is often that we obtain a set of different data models that contain synonyms (different class names that refer to the same class) and homonyms (the same class name is used in different models with a different meaning).

A classical solution for resolving this problem is to integrate the different views into a single global schema [1]. However, in this paper we will integrate the different views by taking a disjoint union of them and adding constraints that express semantical relationships between the classes and relations in the different data models. Since the original views remain part of the global data model we call this *non-destructive integration.* The fact that the original views remain part of the global data model is important in the case of workflow systems because the the views are part of the description of the execution of the workflow. However, even for other types of data-intensive information systems such a design has benefits. Since the original class and relation names from the views are kept in the global data model this will make communication easier with the end-users for which the views described an information need. Moreover, since the relationships between the different views are made explicit it will be easy to see how changes to the global data model affect the different views and vice versa.

The main contribution of this paper is the presentation of a small but powerful set of semantical relationships between classes and relations in different views, and a sound and complete set of inference rules that allows us to derive implied relationships and in particular whether there is a conflict in the resulting global schema.

The paper is organized as follows. Section 2 develops a small example that allows us to informally present our non-destructive approach of form-based view integration. In Section 3, we formally specify the problem. The schema and their instances are defined as well as the different constraints we consider. Sections 4 and 5 present different sets of inference rules for deriving constraints and detecting conflicts. For each set, we prove their soundness and their completeness. In Section 6, we discuss related works. We give our concluding remarks in Section 7.

2 Informal problem definition

We assume that the process of data integration starts with so-called *form-based views* which are essentially hierarchical data structures that describe complex values which can be roughly thought of as tree-shaped graphs. Three examples of such views are given in Figure 1. Each view has the form of a tree which defines all the data that is shown in the view. The nodes of these graphs can be interpreted as classes that contain sets of objects, and the edges can be interpreted as binary relationships between these classes. The root node indicates for which class the view is defined as well as the name of the view. The nodes directly below a node define the attributes of this class. For example, in the Patient view we see that for a patient we have the patient's names, diseases, rooms and treating doctors. At the next level in the view we see that for a disease of a patient we have its types and its names. For the purpose of this paper we will assume that all attributes are set-valued, i.e., they can contain zero, one or more objects.

In the three views in the example we see that there is an overlap in the sense that some objects such as those in the Doctor class in the Patient view and those in the Doctor class in the Doctor view are in fact the same object. In a similar fashion it holds that some of the pairs of the Department-Doctor-Manager relationship in the Doctor view will also be pairs in the Department-Manager relationship. This type of redundancy can be solved by integrating the views into a single new schema, but we propose to leave the original views intact and explicitly specifies such constraint between the different views as illustrated in Figure 2.

There are 8 types of constraints that we will consider:

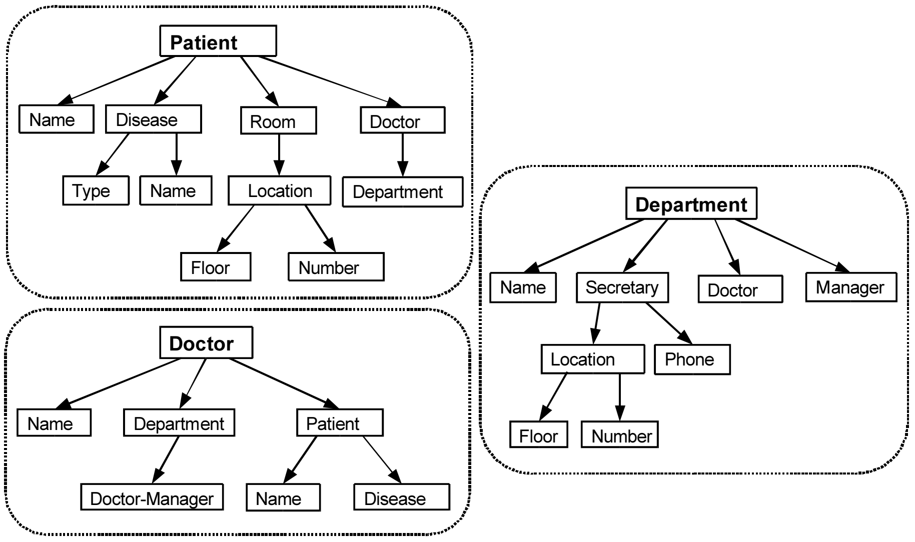

Fig. 1. Three form-based views

ISA The ISA constraint between classes is indicated by an edge that is labeled with $\Rightarrow$. An example is the edge between the Department class in the Doctor view and the Department class in the Department view. The ISA constraint indicates that the objects in one class must also be objects of the other class.

Relational ISA The relational ISA constraint between relationships is indicated by an edge that is labeled with $\Rightarrow^{\downarrow}$. An example is the edge between the Department-Doctor-Manager relationship in the Doctor view and the Department-Manager relationship in the Department view. This constraint indicates that all pairs of the first relationship are also pairs of the second relationship.

Inverse Relational ISA The inverse relational ISA constraint between relationships is indicated by an edge that is labeled with $\Rightarrow^{\uparrow}$. An example is the edge between the Patient-Doctor relationship in the Patient view and the Doctor-Patient relationship in the Doctor view. This constraint indicates that all the inverse pairs of the first relationship are also pairs of the second relationship.

Disjointness The disjointness constraint between classes is indicated by an edge that is labeled with $\not\sim$. An example is the edge between the Location class in the Patient view and the Location class in the Department view. This constraint indicates that the two classes cannot have common objects.

Relational Disjointness The relational disjointness constraint between relationships is indicated by an edge that is labeled with $\not\sim^{\downarrow}$. This constraint indicates that the two relationships cannot have common pairs.

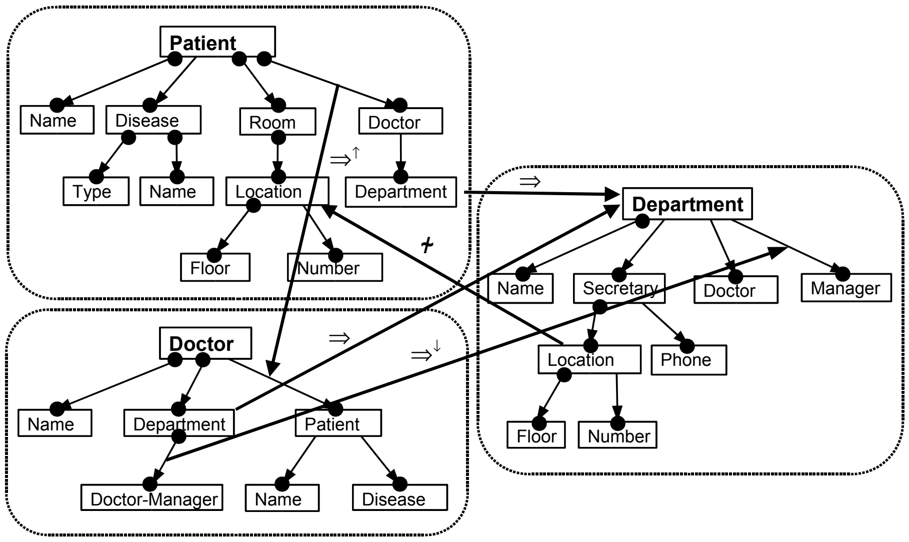

Fig. 2. Three integrated form-based views

Inverse Relational Disjointness The inverse relational disjointness constraint between relationships is indicated by an edge that is labeled with $\not\sim^{\uparrow}$. This constraint indicates that there cannot be a pair in one relationship such that the inverse pair is in the other relationship.

Totalness The totalness constraint of a relationship r is indicated by a solid dot at the beginning of the edge of r. It indicates that the relationship r is total, i.e.,. that for every object o in the source class of r there is a pair of r whose first component is o. For example, every patient has a name, a room and a doctor, but probably has no disease.

Surjectivity constraint The surjectivity constraint of a relationship r is indicated by a solid dot at the end of the edge of r. It indicates that the relationship r is surjective, i.e., for every object o in the target class of r there is a pair of r whose second component is o.

The schema of Figure 2 has a straightforward interpretation that is similar to that of FDM [2], binary ORM [3] and the data models that are used in descriptive logics. Note that in all these models the instances of a schema are essentially graphs that somehow match the schema. Since the original views are still present in the schema it is also clear that for each view we can define a projection on the instances of this schema. Although this projection will usually define a graph it can always be transformed into a forest by splitting nodes with two incoming edges. This means that the general approach is here that of the *local as view* (LAV) approach as defined in [4].

When the constraints are added to the views it is possible that conflicts appear. For example, if there is an ISA constraint between the classes A and B

and at the same time a disjointness constraint between them then the class A nor B can never be populated.

In the remainder of this paper, we discuss the problem of reasoning over schemas with such constraints in order to infer implied such constraints and to detect conflicts.

3 Formal problem definition

For formally specifying the form-based views, we use a graph representation instead of a tree representation as presented in the previous section. We use this simplified data representation since our aim is to provide some constraints that can be used in a context broader than the forms. As such, we are now considering these two types of data models:

- *Frame:* graph structure related to an original form-based view,
- *Schema:* graph structure related to the union of disjoint form-based views, with constraints among them.

In the following paragraphs, we present these types by giving their schema definition and their instance definition.

A *frame* is a multigraph where the nodes represent classes and the edges relationships. More formally:

Definition 1 (Frame). *A* frame *is a tuple $F = (C, R, s, t)$ with C a set of classes, R a set of relationships, $s : R \to C$ a function that indicates the source class of a relationship, and $t : R \to C$ a function that indicates the target class of a relationship.*

Definition 2 (Instance). *An* instance *of a frame $F = (C, R, s, t)$ is a tuple $I = (O, [\![\cdot]\!])$ with O a set of objects, and $[\![\cdot]\!]$ the interpretation function that maps classes $c \in C$ to a subset of O, denoted as $[\![c]\!]$, and relationships $r \in R$ to subsets of $O \times O$, denoted as $[\![r]\!]$, such that, for all relationships $r \in R$, it holds that:*

$$[\![r]\!] \subseteq [\![s(r)]\!] \times [\![t(r)]\!] \tag{1}$$

A *schema* is a frame over which some constraints are specified.

Definition 3 (Constraint). *Given a frame $F = (C, R, s, t)$ a* constraint *is one of the following:*

subclass $c_1 \Rightarrow c_2$, $r_1 \Rightarrow^{\downarrow} r_2$, $r_1 \Rightarrow^{\uparrow} r_2$
cardinality $r\cdot$, $\cdot r$
disjointness $c_1 \not\sim c_2$, $r_1 \not\sim^{\downarrow} r_2$, $r_1 \not\sim^{\uparrow} r_2$

with $r, r_1, r_2 \in R$ and $c_1, c_2 \in C$. We let $I \vdash k$ denote that constraint k holds for instance $I = (O, [\![\cdot]\!])$. Then we have:

$$I \vdash c_1 \Rightarrow c_2 \textit{ iff } [\![c_1]\!] \subseteq [\![c_2]\!] \tag{2}$$
$$I \vdash r_1 \Rightarrow^{\downarrow} r_2 \textit{ iff } [\![r_1]\!] \subseteq [\![r_2]\!] \tag{3}$$
$$I \vdash r_1 \Rightarrow^{\uparrow} r_2 \textit{ iff } [\![r_1]\!]^{-1} \subseteq [\![r_2]\!] \tag{4}$$
$$I \vdash \cdot r \textit{ iff } [\![s(r)]\!] = \{o_1 \mid (o_1, o_2) \in [\![r]\!]\} \tag{5}$$
$$I \vdash r\cdot \textit{ iff } [\![t(r)]\!] = \{o_2 \mid (o_1, o_2) \in [\![r]\!]\} \tag{6}$$
$$I \vdash c_1 \not\sim c_2 \textit{ iff } [\![c_1]\!] \cap [\![c_2]\!] = \varnothing \tag{7}$$
$$I \vdash r_1 \not\sim^{\downarrow} r_2 \textit{ iff } [\![r_1]\!] \cap [\![r_2]\!] = \varnothing \tag{8}$$
$$I \vdash r_1 \not\sim^{\uparrow} r_2 \textit{ iff } [\![r_1]\!]^{-1} \cap [\![r_2]\!] = \varnothing \tag{9}$$

Definition 4 (Schema). *A* schema *is a tuple $S = (F, K)$ with F a frame and K a finite set of constraints over F.*

Definition 5 (Instance). *An* instance of schema S *is an instance I of frame F such that $I \vdash k$ for all constraints $k \in K$.*

4 Inference rules for subclass and cardinality constraints

In this section, we present the sets of inference rules M_1 and M_2 that only derive constraints of the forms $c_1 \Rightarrow c_2$, $r_1 \Rightarrow^{\downarrow} r_2$, $r_1 \Rightarrow^{\uparrow} r_2$ and of the forms $\cdot r$ and $r\cdot$, respectively. In Figure 3, we give the set of rules $M1$ and in Figure 4, we give the set of rules M_2. We assume that the inference rules are defined given a frame $F = (C, R, s, t)$ and variables $c, c_1, c_2, \ldots$ range over C and $r, r_1, r_2, \ldots$ range over R. If for a relationship $r \in R$, it holds that $s(r) = c_1$ and $t(r) = c_2$ then this is denoted as $c_1 \xrightarrow{r} c_2$. We will also assume that K^* is the closure of K under the rules in $M_1 \cup M_2$.

4.1 Instance construction

For proving the completeness of inference rules in $M_1 \cup M_2$, we construct the instances I^{tot}, I^{surj} given a schema $S = (F, K)$ with $\cdot r \notin K^*$ and $r\cdot \notin K^*$, respectively. Informally, the instance I^{tot} is constructed as follows. It is assumed that $\cdot r \notin K^*$. We introduce two objects, o_1 and o_2 where o_1 is in only the super-classes of $s(r)$ and o_2 is simply in all classes of the schema. Then we fill the relations with pairs that contain o_1 to satisfy the surjectivity and totalness constraints. The construction of I^{surj} is similar except we assume that $r\cdot \notin K^*$ and replace $s(r)$ with $t(r)$. This leads to the following formal definition.

Definition 6 (Instances I^{tot} and I^{surj}). *Given a schema S with $\cdot r \notin K^*$, we define $I^{tot} = (O^{tot}, [\![\cdot]\!]^{tot})$ such that $O^{tot} = \{o_1, o_2\}$ and $[\![\cdot]\!]^{tot}$ the smallest*

$$\textsc{Refl}\ \frac{}{c \Rightarrow c} \qquad \textsc{Trans}\ \frac{c_1 \Rightarrow c_2 \qquad c_2 \Rightarrow c_3}{c_1 \Rightarrow c_3} \qquad \textsc{RelRfl}\ \frac{}{r \Rightarrow^{\downarrow} r}$$

$$\textsc{RelTr1}\ \frac{r_1 \Rightarrow^{\downarrow} r_2 \qquad r_2 \Rightarrow^{\downarrow} r_3}{r_1 \Rightarrow^{\downarrow} r_3} \qquad \textsc{RelTr2}\ \frac{r_1 \Rightarrow^{\downarrow} r_2 \qquad r_2 \Rightarrow^{\uparrow} r_3}{r_1 \Rightarrow^{\uparrow} r_3}$$

$$\textsc{RelTr3}\ \frac{r_1 \Rightarrow^{\uparrow} r_2 \qquad r_2 \Rightarrow^{\downarrow} r_3}{r_1 \Rightarrow^{\uparrow} r_3} \qquad \textsc{RelTr4}\ \frac{r_1 \Rightarrow^{\uparrow} r_2 \qquad r_2 \Rightarrow^{\uparrow} r_3}{r_1 \Rightarrow^{\downarrow} r_3}$$

$$\textsc{IsaPr1}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ r_1\cdot \qquad r_1 \Rightarrow^{\downarrow} r_2 \end{array}}{c_2 \Rightarrow c_4} \qquad \textsc{IsaPr2}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ r_1\cdot \qquad r_1 \Rightarrow^{\uparrow} r_2 \end{array}}{c_2 \Rightarrow c_3}$$

$$\textsc{IsaPr3}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ \cdot r_1 \qquad r_1 \Rightarrow^{\downarrow} r_2 \end{array}}{c_1 \Rightarrow c_3} \qquad \textsc{IsaPr4}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ \cdot r_1 \qquad r_1 \Rightarrow^{\uparrow} r_2 \end{array}}{c_1 \Rightarrow c_4}$$

Fig. 3. Set of inference rules M_1

$$\textsc{RelPr1}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_3 \Rightarrow c_1 \qquad \cdot r_1 \\ r_1 \Rightarrow^{\downarrow} r_2 \end{array}}{\cdot r_2} \qquad \textsc{RelPr2}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_4 \Rightarrow c_1 \qquad \cdot r_1 \\ r_1 \Rightarrow^{\uparrow} r_2 \end{array}}{r_2\cdot}$$

$$\textsc{RelPr3}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_4 \Rightarrow c_2 \qquad r_1\cdot \\ r_1 \Rightarrow^{\downarrow} r_2 \end{array}}{r_2\cdot} \qquad \textsc{RelPr4}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_3 \Rightarrow c_2 \qquad r_1\cdot \\ r_1 \Rightarrow^{\uparrow} r_2 \end{array}}{\cdot r_2}$$

Fig. 4. Set of inference rules M_2

function[3] *that satisfies the following rules for all classes c:*

$$o_1 \in [\![c]\!] \ \textit{if}\ s(r) \Rightarrow c \in K^* \tag{10}$$

$$o_2 \in [\![c]\!] \tag{11}$$

[3] The ordering over set-valued functions over the same domain is defined such that f is smaller than g iff $f(x) \subseteq g(x)$ for all x in the domain.

and the following rules for all relationships r:

$$(o_1, o_2) \in [\![r]\!] \textit{ if } \cdot r_1 \in K^* \wedge s(r) \Rightarrow s(r_1) \in K^* \wedge r_1 \Rightarrow^{\downarrow} r \in K^* \tag{12}$$

$$(o_1, o_2) \in [\![r]\!] \textit{ if } r_1 \cdot \in K^* \wedge s(r) \Rightarrow t(r_1) \in K^* \wedge r_1 \Rightarrow^{\uparrow} r \in K^* \tag{13}$$

$$(o_2, o_1) \in [\![r]\!] \textit{ if } \cdot r_1 \in K^* \wedge s(r) \Rightarrow s(r_1) \in K^* \wedge r_1 \Rightarrow^{\uparrow} r \in K^* \tag{14}$$

$$(o_2, o_1) \in [\![r]\!] \textit{ if } r_1 \cdot \in K^* \wedge s(r) \Rightarrow t(r_1) \in K^* \wedge r_1 \Rightarrow^{\downarrow} r \in K^* \tag{15}$$

$$(o_2, o_2) \in [\![r]\!] \tag{16}$$

The construction of I^{surj} is identical except that $s(r)$ is replaced with $t(r)$.

4.2 Soundness and completeness of rules

Theorem 1. *Given a schema $S = (F, K)$ with K containing only subclass constraints and cardinality constraints and K^* the closure of K under the rules in $M_1 \cup M_2$ then*

1. *$c_1 \Rightarrow c_2 \in K^*$ iff $I \vdash c_1 \Rightarrow c_2$ for all instances I of S,*
2. *$r_1 \Rightarrow^{\downarrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\downarrow} r_2$ for all instances I of S,*
3. *$r_1 \Rightarrow^{\uparrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\uparrow} r_2$ for all instances I of S,*
4. *$\cdot r \in K^*$ iff $I \vdash \cdot r$ for all instances I of S, and*
5. *$r \cdot \in K^*$ iff $I \vdash r \cdot$ for all instances I of S.*

Proof. (Sketch) The *only-if* part of all the propositions is easily proved by verifying that all the inference rules in $M_1 \cup M_2$ are sound which follows from the semantics of the constraints as defined in Definition 3.

The *if* part proceeds by showing that for all constraints it holds that if it is not in K^*, then it does not hold in at least one of I^{tot} and I^{surj}. It can also be shown that I^{tot} and I^{surj} are instances of S. Finally, it can be shown that if $\cdot r \notin K^*$ ($r \cdot \notin K^*$) then this constraint is not satisfied by I^{tot} (I^{surj}). □

5 Inference rules for deriving disjointness constraints

In this section we will also consider constraints of the forms $c_1 \not\Leftrightarrow c_2$, $r_1 \not\Leftrightarrow^{\downarrow} r_2$ and $r_1 \not\Leftrightarrow^{\uparrow} r_3$. We give three sets of these rules, namely M_3 (Figure 5) and M_4 (Figure 6). We will assume from now on that K^* is the closure of K under the rules in $M_1, \ldots, M_4$.

With disjointness constraints it is possible to define schemas in which certain classes and relations cannot be populated. To find such conflicts we introduce the following syntactical notion of conflict.

Definition 7 (conflict). *A* conflict *is a constraint of the form $c \not\Leftrightarrow c$ or $r \not\Leftrightarrow^{\downarrow} r$. If a set of constraints K does not contain such a conflict then it is said to be* conflict-free*.*

Note that $r \not\Leftrightarrow^{\uparrow} r$ is not a conflict since there are non-empty relations for which it holds.

$$\textsc{DsjSym}\ \frac{c_1 \nsim c_2}{c_2 \nsim c_1} \qquad \textsc{DsjDnSym}\ \frac{r_1 \nsim^{\downarrow} r_2}{r_2 \nsim^{\downarrow} r_1}$$

$$\textsc{DsjUpSym}\ \frac{r_1 \nsim^{\uparrow} r_2}{r_2 \nsim^{\uparrow} r_1} \qquad \textsc{DsjInh}\ \frac{c_1 \nsim c_2 \qquad c_3 \Rightarrow c_2}{c_1 \nsim c_3}$$

$$\textsc{DsjInh1}\ \frac{r_1 \nsim^{\downarrow} r_2 \qquad r_3 \Rightarrow^{\downarrow} r_2}{r_1 \nsim^{\downarrow} r_3} \qquad \textsc{DsjInh2}\ \frac{r_1 \nsim^{\downarrow} r_2 \qquad r_3 \Rightarrow^{\uparrow} r_2}{r_1 \nsim^{\uparrow} r_3}$$

$$\textsc{DsjInh3}\ \frac{r_1 \nsim^{\uparrow} r_2 \qquad r_3 \Rightarrow^{\downarrow} r_2}{r_1 \nsim^{\uparrow} r_3} \qquad \textsc{DsjInh4}\ \frac{r_1 \nsim^{\uparrow} r_2 \qquad r_3 \Rightarrow^{\uparrow} r_2}{r_1 \nsim^{\downarrow} r_3}$$

$$\textsc{DsjPr1}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_2 \nsim c_4 \end{array}}{r_1 \nsim^{\downarrow} r_2} \qquad \textsc{DsjPr2}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_2 \nsim c_3 \end{array}}{r_1 \nsim^{\uparrow} r_2}$$

$$\textsc{DsjPr3}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_1 \nsim c_3 \end{array}}{r_1 \nsim^{\downarrow} r_2} \qquad \textsc{DsjPr4}\ \frac{\begin{array}{c} c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \\ c_1 \nsim c_4 \end{array}}{r_1 \nsim^{\uparrow} r_2}$$

Fig. 5. Set of inference rules M_3

5.1 Instance construction

For proving the completeness of the inference rules in $M_1, \ldots, M_4$, we construct the instances I^{base}, $I^{\nsim}$, $I^{\nsim^{\uparrow}}$ and $I^{\nsim^{\downarrow}}$ given a schema $S = (F, K)$ with $F = (C, R, s, t)$.

Informally we can describe the construction of I^{base} as follows. For each class c we introduce a distinct object o_c that is in c and all its super-classes. For each relation r we introduce the objects o^1_r (and o^2_r) that are in the source (target) class of r and all its (implied) super-classes. Next we add the pair (o^1_r, o^2_r) to relation r and all its super-relations, and the inverse to all its inverse super-relations. Finally, to satisfy the totalness and surjectivity constraints we add for relations q with such a constraint and each object o in a class c a pair with o and either o^1_q or o^2_q to q and its inverse and normal sub-relations. This leads to the following formal definition:

Definition 8 (Instance I^{base}). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ we define $I^{base} = (O^{base}, [\![\cdot]\!]^{base})$ such that $O^{base} = \{o_c \mid c \in C\} \cup \{o^1_r, o^2_r \mid r \in R\}$ and the interpretation function is defined as the smallest interpretation function*

$$\textsc{CnflPr1}\ \frac{\begin{array}{c} c_1 \xrightarrow{r} c_2 \\ r \nsim^{\downarrow} r \qquad \cdot r \end{array}}{c_1 \nsim c_1} \qquad\qquad \textsc{CnflPr2}\ \frac{\begin{array}{c} c_1 \xrightarrow{r} c_2 \\ r \nsim^{\downarrow} r \qquad r\cdot \end{array}}{c_2 \nsim c_2}$$

$$\textsc{IsaCnfl}\ \frac{c_1 \nsim c_1}{c_1 \Rightarrow c_2} \qquad\qquad \textsc{IsaDnCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \Rightarrow^{\downarrow} r_2}$$

$$\textsc{IsaUpCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \Rightarrow^{\uparrow} r_2} \qquad\qquad \textsc{TotCnfl}\ \frac{c_1 \xrightarrow{r} c_2 \qquad c_1 \nsim c_1}{\cdot r}$$

$$\textsc{SurjCnfl}\ \frac{c_1 \xrightarrow{r} c_2 \qquad c_2 \nsim c_2}{r\cdot} \qquad\qquad \textsc{DisjCnfl}\ \frac{c_1 \nsim c_1}{c_1 \nsim c_2}$$

$$\textsc{DisjDnCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \nsim^{\downarrow} r_2} \qquad\qquad \textsc{DisjUpCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \nsim^{\uparrow} r_2}$$

Fig. 6. Set of inference rules M_4

that satisfies the following rules for all classes c:

$$o_d \in [\![c]\!] \textit{ if } d \Rightarrow c \in K^* \tag{17}$$
$$o_r^1 \in [\![c]\!] \textit{ if } r \Rightarrow^{\downarrow} q \in K^* \wedge s(q) \Rightarrow c \in K^* \tag{18}$$
$$o_r^1 \in [\![c]\!] \textit{ if } r \Rightarrow^{\uparrow} q \in K^* \wedge t(q) \Rightarrow c \in K^* \tag{19}$$
$$o_r^2 \in [\![c]\!] \textit{ if } r \Rightarrow^{\downarrow} q \in K^* \wedge t(q) \Rightarrow c \in K^* \tag{20}$$
$$o_r^2 \in [\![c]\!] \textit{ if } r \Rightarrow^{\uparrow} q \in K^* \wedge s(q) \Rightarrow c \in K^* \tag{21}$$

and the following rules for all relationships r:

$$(o_q^1, o_q^2) \in [\![r]\!] \textit{ if } q \Rightarrow^{\downarrow} r \in K^* \tag{22}$$
$$(o_q^2, o_q^1) \in [\![r]\!] \textit{ if } q \Rightarrow^{\uparrow} r \in K^* \tag{23}$$
$$(o, o_q^2) \in [\![r]\!] \textit{ if } o \in [\![s(q)]\!] \wedge q \Rightarrow^{\downarrow} r \in K^* \wedge \cdot q \in K^* \tag{24}$$
$$(o_q^1, o) \in [\![r]\!] \textit{ if } o \in [\![t(q)]\!] \wedge q \Rightarrow^{\downarrow} r \in K^* \wedge q\cdot \in K^* \tag{25}$$
$$(o, o_q^1) \in [\![r]\!] \textit{ if } o \in [\![t(q)]\!] \wedge q \Rightarrow^{\uparrow} r \in K^* \wedge q\cdot \in K^* \tag{26}$$
$$(o_q^2, o) \in [\![r]\!] \textit{ if } o \in [\![s(q)]\!] \wedge q \Rightarrow^{\uparrow} r \in K^* \wedge \cdot q \in K^* \tag{27}$$

Lemma 1. *Given a schema $S = (F, K)$ such that K^* is conflict-free, then the corresponding I^{base} is an instance of S.*

Proof. (Sketch) We first show that I^{base} is an instance of F (i.e., the proposition (1) holds). We then can show that all constraints in K will also hold for I^{base}. Finally we verify that the constraints from (2) to (9)) are satisfied. $\square$

Informally we can describe the construction of $I^{\nsim}$ as follows. We assume that $a \nsim b \notin K^*$. Then we construct the instance as for I^{base} except that we introduce a special object o_{ab} that is placed both in class a and in class b and in all their super-classes. This leads to the following formal definition:

Definition 9 (Instance $I^{\nsim}$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $a \nsim b \notin K^*$ we define $I^{\nsim} = (O^{\nsim}, [\![\cdot]\!]^{\nsim})$ such that $O^{\nsim} = O^{base} \cup \{o_{ab}\}$ and the interpretation as for I^{base} but with the following additional rule:*

$$o_{ab} \in [\![c]\!] \textit{ if } a \Rightarrow c \in K^* \vee b \Rightarrow c \in K^* \tag{28}$$

Lemma 2. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $a \nsim b \notin K^*$, then the corresponding $I^{\nsim}$ is an instance of S.*

Proof. (Sketch) The proof proceeds similar to that of Lemma 1 except that for some propositions we need to consider extra cases. □

Informally we can describe the construction of $I^{\nsim\downarrow}$ as follows. We assume that $p \nsim^{\downarrow} q \notin K^*$. Then we construct the instance as for I^{base} except that we introduce a special pair (o^1_{pq}, o^2_{pq}) that is placed both in the relation p and in the relation q and in all their super-relations, and the inverse is placed in all the inverse super-relations. This leads to the following formal definition:

Definition 10 (Instance $I^{\nsim\downarrow}$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $p \nsim^{\downarrow} q \notin K^*$ we define $I^{\nsim\downarrow} = (O^{\nsim\downarrow}, [\![\cdot]\!]^{\nsim\downarrow})$ such that $O^{\nsim\downarrow} = O^{base} \cup \{o^1_{pq}, o^2_{pq}\}$ and the interpretation function as for I^{base} but with the following additional rules for all classes c:*

$$o^1_{pq} \in [\![c]\!] \textit{ if } (p \Rightarrow^{\downarrow} r \in K^* \vee q \Rightarrow^{\downarrow} r \in K^*) \wedge s(r) \Rightarrow c \in K^* \tag{29}$$
$$o^1_{pq} \in [\![c]\!] \textit{ if } (p \Rightarrow^{\uparrow} r \in K^* \vee q \Rightarrow^{\uparrow} r \in K^*) \wedge t(r) \Rightarrow c \in K^* \tag{30}$$
$$o^2_{pq} \in [\![c]\!] \textit{ if } (p \Rightarrow^{\downarrow} r \in K^* \vee q \Rightarrow^{\downarrow} r \in K^*) \wedge t(r) \Rightarrow c \in K^* \tag{31}$$
$$o^2_{pq} \in [\![c]\!] \textit{ if } (p \Rightarrow^{\uparrow} r \in K^* \vee q \Rightarrow^{\uparrow} r \in K^*) \wedge s(r) \Rightarrow c \in K^* \tag{32}$$

and for all relationships r:

$$(o^1_{pq}, o^2_{pq}) \in [\![r]\!] \textit{ if } p \Rightarrow^{\downarrow} r \in K^* \vee q \Rightarrow^{\downarrow} r \in K^* \tag{33}$$
$$(o^2_{pq}, o^1_{pq}) \in [\![r]\!] \textit{ if } p \Rightarrow^{\uparrow} r \in K^* \vee q \Rightarrow^{\uparrow} r \in K^* \tag{34}$$

Informally we can describe (o^1_{pq}, o^2_{pq}) as the typical pair that is both in the relationship p and q.

Lemma 3. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $p \nsim^{\downarrow} q \notin K^*$, then the corresponding $I^{\nsim\downarrow}$ is an instance of S.*

Proof. The proof proceeds similar to that of Lemma 2 and considers the extra cases for Prop. (1), Constr. (7), Constr. (8) and Constr. (9) using the assumption that $p \nsim^{\downarrow} q \notin K^*$. □

Definition 11 (Instance $I^{\not\sim^{\uparrow}}$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $p \not\sim^{\uparrow} q \notin K^*$ we define $I^{\not\sim^{\uparrow}}$ similar to $I^{\not\sim^{\downarrow}}$ but here we add a pair $(o^{12}_{pq}, o^{21}_{pq})$ such that it is in $[\![p]\!]^{\not\sim^{\uparrow}}$ and its inverse, $(o^{21}_{pq}, o^{12}_{pq})$, is in $[\![q]\!]^{\not\sim^{\uparrow}}$.*

Lemma 4. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $p \not\sim^{\uparrow} q \notin K^*$, then the corresponding $I^{\not\sim^{\uparrow}}$ is an instance of S.*

Proof. The proof proceeds similar to that of Lemma 3. □

5.2 Soundness and completeness of rules

Theorem 2. *Given a schema $S = (F, K)$ with K^* the closure of K under the rules in $M_1, \ldots, M_4$ then*

1. *$c_1 \Rightarrow c_2 \in K^*$ iff $I \vdash c_1 \Rightarrow c_2$ for all instances I of S,*
2. *$r_1 \Rightarrow^{\downarrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\downarrow} r_2$ for all instances I of S,*
3. *$r_1 \Rightarrow^{\uparrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\uparrow} r_2$ for all instances I of S,*
4. *$\cdot r \in K^*$ iff $I \vdash \cdot r$ for all instances I of S,*
5. *$r\cdot \in K^*$ iff $I \vdash r\cdot$ for all instances I of S,*
6. *$c_1 \not\sim c_2 \in K^*$ iff $I \vdash r_1 \not\sim r_2$ for all instances I of S,*
7. *$r_1 \not\sim^{\downarrow} r_2 \in K^*$ iff $I \vdash r_1 \not\sim^{\downarrow} r_2$ for all instances I of S, and*
8. *$r_1 \not\sim^{\uparrow} r_2 \in K^*$ iff $I \vdash r_1 \not\sim^{\uparrow} r_2$ for all instances I of S.*

Proof. (Sketch) The *only-if* part of all the propositions is easily proved by verifying that all the inference rules in $M_1, \ldots, M_4$ are sound, which follows straightforwardly from the semantics of the constraints as defined in Definition 3.

The *if* part is proven in two steps. We first show that for each type of constraint that if K^* is conflict-free then it holds, and then we show that from this it follows that it holds for any K. □

Corollary 1. *The rules in $M_1, \ldots, M_3$ are sufficient to detect if K^* is conflict-free.*

Proof. From the preceding corollary it follows that if the closure of K under $M_1, \ldots, M_3$ does not contain a conflict then the closure under $M_1, \ldots, M_4$ will also not contain a conflict. It is also clear that if the first closure contains a conflict then so does the second closure. Therefore the first closure contains a conflict iff the second closure does. □

6 Related work

There has already been a large amount of research on the topic of data integration [4] and reasoning about taxonomies in general [5] and database schemas in particular [6]. As is argued in [7] the two subjects are closely linked together since the ability to reason over the views can be used to check the representation for inconsistencies and redundancies, and to maintain the system in response to changes in the data needs. In particular, [8] presents a reasoning approach for automating a significant part of the schema integration process and [9] relies on a reasoning support to improve the quality of data.

Description Logics (DL) are a well-known family of knowledge representation formalisms that descend from KL-ONE [10]. Long since, they have been applied to data management [11] and information integration [12]. The basic idea is to express database schemas as DL knowledge bases so that DL reasoning techniques can be used to reason about the schema. Although this approach can be restricted to useful fragments where reasoning is still tractable, e.g. [13] and [14], it often already becomes intractable for relatively small fragments [15] and even more so when the concept of *inverse role* is added [16]. It was to the best of our knowledge not yet known that the fragment that is proposed in this paper, which can express such inverse roles, has a relatively simple set of inference rules that is sound and complete and allows tractable reasoning.

7 Conclusion

In this paper we have proposed a view integration method that leaves the original views intact and allows their relationships to be defined by constraints that explicitly express semantical relationships between the components of the different views. Although the motivation of this approach comes from workflow and case management systems where the original views are important for the description of the workflow, this approach can also be beneficial for data integration in more general settings. To support the integration process we have proposed a set of inference rules that allows us to derive implied semantical relationships and especially whether there are conflicts in the integrated schema. We have shown that these sets of rules are sound and complete for all proposed types of constraints, and that subsets of these rules can be already complete for certain subsets of the constraints. Finally it was shown that the inference rules provide in all cases a tractable inference mechanism.

References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. **18** (1986) 323–364
2. Shipman, D.W.: The functional data model and the data language DAPLEX. ACM Trans. Database Syst. **6** (1981) 140–173
3. Halpin, T.: Information modeling and relational databases: from conceptual analysis to logical design. Morgan Kaufmann Publishers (2001)
4. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS. (2002) 233–246
5. Bergamaschi, S., Sartori, C.: On taxonomic reasoning in conceptual design. ACM Trans. Database Syst. **17** (1992) 385–422
6. Formica, A., Missikoff, M.: Inheritance processing and conflicts in structural generalization hierarchies. ACM Comput. Surv. **36** (2004) 263–290

7. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Information integration: Conceptual modeling and reasoning support. In: Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIS'98). (1998) 280–291
8. Kashyap, V., Sheth, A.P.: Semantic and schematic similarities between database objects: A context-based approach. VLDB J. **5** (1996) 276–304
9. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Source integration in data warehousing. In: Proc. of the 9th Int. Workshop on Database and Expert Systems Applications (DEXA'98), IEEE Computer Society Press (1998) 192–197
10. Brachman, R., Schmolze, J.: An overview of the KL-ONE knowledge representation system. Cognitive Science (1985) 171–216
11. Kirk, T., Levy, A.Y., Sagiv, Y., Srivastava, D.: The Information Manifold. In Knoblock, C., Levy, A., eds.: Information Gathering from Heterogeneous, Distributed Environments, Stanford University, Stanford, California (1995)
12. Calvanese, D., Giacomo, G.D., Lenzerini, M., Nardi, D., Rosati, R.: Description logic framework for information integration. In: KR. (1998) 2–13
13. Brachman, R., Levesque, H.: The tractability of subsumption in frame-based description languages. In: AAAI-84, Austin, Texas (1984) 34–37
14. Borgida, A., Brachman, R.J., McGuinness, D.L., Resnick, L.A.: CLASSIC: a structural data model for objects. In: Proc. of the ACM SIGMOD International Conference on Management of Data, Portland, Oregon (1989) 58–67
15. Nutt, W., Donini, F.M., Lenzerini, M., Nardi, D.: The complexity of concept languages. Inf. Comput. **134** (1997) 1–58
16. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. Journal of Logic and Computation **9** (1999) 385–410