What's not in a name: Some Properties of a Purely Structural Approach to Integrating DL Terminologies.

Alex Borgida
Dept. of Computer Science
Rutgers University
borgida@cs.rutgers.edu

Ralf Küsters
Theoretical Computer Science
RWTH Aachen
kuesters@cs.rwth-aachen.de

Abstract

One approach to integrating knowledge bases is based on finding assertions that relate the expressions in the constituent terminologies. For knowledge bases with many terms this task requires computer support. We set up a formal framework for merging Description Logic TBoxes, and then explore the limits of a purely structural approach to the problem of finding inter-relationships between knowledge bases. Some theoretical notions are empirically examined in a real medical ontology (GALEN).

1 Integration — An Overview

Suppose we are given two terminologies (also called ontologies, conceptual models) — collections of terms and their meanings as used in some universe of discourse (UofD). Our task is to *integrate/align* them into a new terminology that captures the meaning of the original terms *and* their inter-relationships.

This problem is motivated by several application scenarios. First, terminologies have been developed independently by multiple research groups in a variety of application areas, including medicine and natural language processing; for example, quite recently, several ontologies about logistics were used as part of the HPKB project. In each of these cases there have been efforts at merging these terminologies [16, 20, 17].

Second, there are many examples of *heterogeneous information sources*, such as federated databases or semi-structured data on the internet, which keep information about aspects of some single domain of discourse (e.g., movies), and for which one desires uniform access through a "conceptual schema", mostly for the purpose of seamless querying. Note that in these circumstances the original information sources are meant to be kept around, and at least the data in them is supposed to be updated independently.

Predating these, a traditional step in database design has been so-called "view integration": taking the descriptions of the database needs of different parts of an organization ("views"), and coming up with a unified central schema, which reconciles the differences [4]. In this case, the data is supposed to be stored in the unified logical schema, and each view must be derivable from it.

In the more general context of software requirements engineering, one encounters the notion of *view reconciliation*, which deals once again with the differing conceptual/generic aspects of a UofD, as seen by different stake holders (but unlike the database problem, does not have to deal with data aspects).

As we shall see later, a basic assumption made in this paper is that the original terminologies continue to *exist independently* of the final merged result, and are themselves *accurate* models of the UofD.

What makes integration complicated? Naively, one could ask why it is not possible to just take the union of the original schemas/knowledge bases in order to merge them. The first obvious problem is that the terminologies could be expressed using *different languages* (a.k.a. data models). Usually, this problem is addressed by either providing so-called "wrapper software" to support a uniform procedural/API interface, or by translating the original sources into the same (often "semantically richer") language, such as the Entity-Relationship data model or KR languages, like CYC, OKBC, or some description logic.

Even if the schemas are expressed in the same language, one cannot simply union them. First, duplication caused by identically named terms requires collecting such identical identifiers, and unifying them into a single reference. Merging also involves detecting homonyms — identical terms that refer to different concepts — which need to be renamed to avoid inappropriate unification. Although these problems can be dealt with relatively easily, finding and resolving some of the following "mismatches" is more difficult:

1. The designers of the original schemas may choose different identifiers for the same concept in the UofD; this might range from simple morphological variant (e.g. PATIENT vs. PATIENTS), to *synonymous* (e.g., DOCTOR vs. PHYSICIAN), to hypernyms (e.g., SURGEON vs. DOCTOR), and even partly related, but not synonymous terms (OFFICE-VISIT vs. APPOINTMENT).
2. More significantly, designers may model the same ideas at different level of detail: an address might be modeled as a single string or, in greater detail, as an object with attributes street, city, state etc.

3. Finally, more substantial modeling differences arise out of phenomena such as *reification*, illustrated in the following example: the relationship between a patient and her doctor might be modeled by the role $\mathsf{treatedBy}$, or there might be a new concept, TREAT, with functional attributes who (a doctor) and whom (a patient).

Approaches to integration. The problem of synonymous *identifiers* is usually attacked first, using heuristics such as string manipulation and thesauri (e.g., WordNet) to find potentially co-incidental terms.

The remaining *"structural conflicts"* have been addressed using a variety of different approaches:

- *"Wrappers" and "mediators"* provide procedural solutions. As pointed out in [7] these techniques suffer from the standard defects of procedural approaches in general, including potential lack of consistency between multiple procedures.

- *Structure transformation approaches* rely on catalogues of transformations (e.g., reification/unreification) that can be applied to massage schema portions in order to bring to evidence common terms, which can then be unified. These depend on an in-depth analysis of the "data model" being used (e.g., [13]).

- *Assertional approaches* take the union of the schemas (after removing homonyms) and augment it by *integration assertions* (IAs) in order to relate terms from the original schemas [5, 9].

Later on in this paper, we will focus on this third kind of integration. In the literature several (more or less restricted) types of IAs have been proposed, which are listed next:

Name equivalence: The simplest way to relate concepts in two schemas is to indicate those that are equivalent: $A \equiv B$, where A and B are concept identifiers in S_1 and S_2, respectively. This approach is applied implicitly in [16], and it can be used to inter-relate synonyms (see 1. above).

Identifier mappings: A more expressive form of IAs allows statements of the form $A \equiv C$, where A is an identifier in one terminology, while C can be a more complex description/formula in the other. Such assertions have been used, among others, by Johannesson et al. [12], and are useful when concepts are described on different levels of abstraction (see 2. above)

Identifier subsumption: These resemble name equivalence, except that subsumption is used instead of equality: $A \sqsubseteq B$. This is useful to describe more refined relationships between hierarchies of primitive concepts, such as synonymy, hypernomy, and hyponymy. These kinds of relationships between concepts are used in [15, 18] for information integration

General constraints: These permit subsumption relationships (like $C \sqsubseteq D$) to be expressed between arbitrary descriptions in the two terminologies. Such powerful constraints appear in the work of Lenzerini et al. [7, 9].

What is a good integration? Although the various structure transformation and assertion approaches describe a space of merged schemas, they do not state what are "good" integrated schemas nor how to find them. Yet it is crucial to evaluate the result of merging schemas, and to support the integration process (semi-)automatically. A list of desirable features of merged schemas certainly includes (informal) notions such as completeness, minimality, and understandability (see [4]). Completeness means that the merged schema allows to recover *all and only* the information stored under the constituent schemas. An important initial contribution towards formalizing this notion was made by Hull [14], who defined the notion of *information capacity.*

Definition 1 The schema S *is dominated by* S' if there is a total 1-1 function f from the models of S to those of S'; S and S' have the same information capacity if they dominate each other.

Intuitively, one expects the result S of merging the two schemas S_1 and S_2 to preserve the information capacity of each. Thus, S must dominate S_1 and S_2. On its own, information capacity turns out to allow too many unintuitive ways of merging schemas, because the function f can perform arbitrarily complex computations. As an answer to this shortcoming, several refinements of this approach have been proposed in the literature.

For the transformation-based approach, Miller et al. [19] have examined and improved a variety of transformations by making them preserve information capacity.

For the assertion-based approach, Biskup and Convent [5] introduced the notion of *conflict-freeness* to formalize the integration of relational databases; this was later extended by Ekenberg and Johannesson [12] to deal with schemas (and IAs) formulated as theories in first order predicate logic. The main idea is that for given schemas S_1 and S_2, only those IAs are allowed that are conflict-free w.r.t. the schemas. If $I(S)$ denotes the set of all models of a schema S, Σ' denotes a subset of the signature Σ of S, and $I(S)[\Sigma']$ refers to the projection of the models in $I(S)$ to Σ', then conflict-freeness is defined as follows.

Definition 2 Let $\mathcal{A}$ be IAs between S_1 and S_2 with $\Sigma_1 \cap \Sigma_2 = \emptyset$. Then, S_1 and S_2 are *conflict-free* w.r.t. $\mathcal{A}$ iff $I(S_1 \cup S_2 \cup \mathcal{A})[\Sigma_i] = I(S_i)$ for $i = 1, 2$.

Since $I(S_1 \cup S_2 \cup \mathcal{A})[\Sigma_i] \subseteq I(S_i)$ is always true, the interesting inclusion is in the other direction, which says that every model $\mathcal{I}$ of S_i can be extended to a model $\mathcal{I}'$ of $S_1 \cup S_2 \cup \mathcal{A}$, where *extension* means that $\mathcal{I}'$ and $\mathcal{I}$ coincide on the signature of S_i, say S_1, but $\mathcal{I}'$ might have a bigger domain and in addition interprets the identifiers of S_2. This observation guarantees that the merged schema $S_1 \cup S_2 \cup \mathcal{A}$ dominates S_1 and S_2 (cf. Definition 1). Thus, the result of merging two schemas in a conflict-free manner dominates the original schemas. In this sense, the work by Biskup and Convent in fact refines the notion of domination introduced by Hull.

Intuitively, the inclusion $I(S_1 \cup S_2 \cup \mathcal{A})[\Sigma_i] \supseteq I(S_i)$ means that the assertions in $\mathcal{A}$ *describe* a certain connection between S_1 and S_2 as opposed to imposing additional constraints on the schemas. If they restricted the models of S_1 or S_2, then these assertions could not just describe *connections* between these two schemas: they would be adding additional knowledge too! Additional knowledge is added, for example, when relating not necessarily disjoint concepts in S_1 to disjoint concepts in S_2. The underlying assumption of this approach is that the schemas contain the *complete* structural information expressible in the language of choice. To meet this requirement it might be necessary to revisit the original schemas and to add extra material. Then, if concepts in S_1 are not defined to be disjoint, it would not make sense to relate them to disjoint concepts in S_2.

Tools supporting the integration process. Especially in the presence of large numbers of concepts, one is faced with the problem of finding the relevant IAs and, in the case of the structure transformation approach, to recognize potentially appropriate places to apply transformations. Several papers offer tools supporting these tasks (see, e.g., [13, 8, 17, 16, 21]).

In general, the tools start from heuristics about the lexical similarity between names, but then often go on to take into account some structural information in the terminologies, such as the subsumption hierarchy and role restrictions.

2 The Framework

One significant disadvantage of the tools supporting the integration process is the lack of a formal framework. As a consequence, it is hard to estimate the quality of the computed similarities between elements of the schemas. Moreover, it is not possible to explore the theoretical limits of (semi-)automatic ontology integration. Since this paper aims at investigating these limits, we will now set up a formal framework for merging DL schemas/TBoxes, based on the notion of conflict-freeness as introduced by Biskup and Convent [5].

Our framework can be applied to an arbitrary description logic $\mathcal{L}$ (see, e.g., [11]). For our examples, it is however sufficient to only introduce the small description logic $\mathcal{FL}_\neg$. Let N_C and N_R be disjoint finite sets, the set of *concept names* and the set of *role names*. The set $\Sigma := N_C \cup N_R$ is called the set of *identifiers*. Given a concept name A and a role name r, *concept descriptions* C, D are inductively defined according to the following syntax rules:

$$
\begin{array}{rcll}
C, D & \longrightarrow & \top \mid & \text{(top-concept)} \\
 & & A \mid & \text{(concept name)} \\
 & & \neg A \mid & \text{(primitive negation)} \\
 & & C \sqcap D \mid & \text{(concept conjunction)} \\
 & & \forall r.C & \text{(value restriction)}
\end{array}
$$

As usual, the semantics of these descriptions is defined in terms of interpretations. An *interpretation* I consists of a nonempty set Δ^I, the *domain* of the interpretation, and an interpretation function that assigns to every concept name A a set $A^I \subseteq \Delta^I$, and to every role name r a binary relation $r^I \subseteq \Delta^I \times \Delta^I$. The interpretation function is extended to complex concept descriptions as follows:

$$
\begin{array}{rcl}
\top^I & := & \Delta^I, \\
(\neg A)^I & := & \Delta^I \setminus A^I, \\
(C \sqcap D)^I & := & C^I \cap D^I, \\
(\forall r.C)^I & := & \{d \in \Delta^I \mid \forall e \in \Delta^I : (d, e) \in r^I \rightarrow e \in C^I\}.
\end{array}
$$

A *schema* S (a.k.a. general TBox), is a finite set of axioms of the form

$$C \sqsubseteq D,$$

where both C and D (complex) concept descriptions. An interpretation I is a *model* of S ($I \models S$) iff I satisfies every axiom in S, i.e., $C^I \subseteq D^I$ for every axiom $C \sqsubseteq D$ in S. As before, the set of model of S is denoted by $I(S)$.

Given a schema S, the concept description C is *subsumed* by D modulo S ($S \models C \sqsubseteq D$) iff $I \models C \sqsubseteq D$, i.e., $C^I \subseteq D^I$, for all models I of S; C and D are *equivalent* modulo S ($S \models C \equiv D$) if $S \models C \sqsubseteq D$ and $S \models D \sqsubseteq C$. If S is the empty set, we simply write $C \sqsubseteq D$ and $C \equiv D$ instead of $\emptyset \models C \sqsubseteq D$ and $\emptyset \models C \equiv D$.

Let S_1 and S_2 be to schemas with disjoint sets of identifiers Σ_1 and Σ_2. Then, we take the IAs $\mathcal{A}$ between S_1 and S_2 to be finite sets of name equivalences, i.e., $\mathcal{A}$ is a finite set of equivalences of the form

$$A \equiv B,$$

where $A \in \Sigma_1$ and $B \in \Sigma_2$. Often $\mathcal{A}$ is represented by a (partial) mapping σ from Σ_1 into Σ_2, i.e., σ stands for the assertions

$$\mathcal{A}_\sigma := \{A \equiv \sigma(A) \mid A \in \mathsf{domain}(\sigma)\}.$$

Given some concept description C and a mapping σ, then $\sigma(C)$ denotes the concept description obtained from C by replacing every occurrence of an identifier in C, say A, by $\sigma(A)$ in case A belongs to the domain $\mathsf{domain}(\sigma)$ of σ; otherwise, A is not replaced.

Finally, following the line of Biskup and Convent [5], a *merged schema* is of the form

$$S_1 \cup S_2 \cup \mathcal{A},$$

where S_1 and S_2 are supposed to be *conflict-free* w.r.t. the IAs $\mathcal{A}$. In this case, if the mapping σ corresponds to $\mathcal{A}$, i.e., $\mathcal{A}_\sigma = \mathcal{A}$, σ is called a *conflict-free mapping* (CFM).

We should point out that many difficulties one encounters when integrating schemas, like reification and subsumption relationships between (complex) concepts (see Section 1), are not taken into account here. Nevertheless, we believe that the proposed framework is a good starting point to explore possibilities and limits of structural information in the integration process.

Our framework defines a space of "good" merged schemas. Given the constituent schemas, every merged schema is determined by its CFM. Finding such CFMs is what we will call *structural integration* in the following, motivated by the fact that conflict-freeness is defined in a purely model-theoretic way; lexical similarities are not taken into account.

In [6], it has been shown that computing CFMs is not an easy problem. More precisely, given two schemas, deciding the existence of a total CFM between these schemas is NP-hard. Moreover, when CFMs are allowed to map identifiers to complex terms (identifier mapping), then computing CFMs can be carried out by first computing candidate CFMs and then deciding whether these candidates are in fact conflict-free. The former step (computing candidates) can be reduced to the unification of concept descriptions modulo schemas or matching under side conditions modulo schemas. Although first results on unification and matching in description logics are available (see [3, 2, 1]), schemas have not yet taken into account. Also, it is an open problem whether conflict-freeness is decidable in the context of descriptions logics — in the settings considered in [5, 12], conflict-freeness has been shown to be undecidable [10, 12]. To sum up, until now only little is known about the computation of CFMs (considered as identifier mappings).

3 Limits of purely structural DL integration

In general, given two schemas S_1 and S_2, there exist several CFMs relating their identifiers. Still, only one of the CFMs is the intuitive mapping, i.e., the one a human integrator would come up with. The key question tackled in this paper is to what extent CFMs coincide with the intuitive mapping, and thus, how much of the meaning of an identifier is encoded in the schema and how much is merely intuition of the user. If much of the meaning of an identifier is captured in the schema rather than being the intuition of the user, then one can hope that a program can make use of the structural information to significantly support the integration process.

We will first approach these questions theoretically and then evaluate the problems empirically by looking at a sizeable ontology (GALEN).

Let us start with a very simple example in order to give a feel for CFMs.

Example 3 Let the schemas S_1 and S_2 be defined as follows:

S_1:
$$\begin{aligned} \mathsf{HeartClinic} &\sqsubseteq \mathsf{Clinic} \\ \mathsf{PsychiatricClinic} &\sqsubseteq \mathsf{Clinic} \\ \mathsf{PsychiatricClinic} &\sqsubseteq \neg\mathsf{HeartClinic} \end{aligned}$$

S_2:
$$\begin{aligned} \mathsf{CardiacH} &\sqsubseteq \mathsf{Hospital} \\ \mathsf{PsychiatricH} &\sqsubseteq \mathsf{Hospital} \\ \mathsf{CardiacH} &\sqsubseteq \neg\mathsf{PsychiatricH} \end{aligned}$$

If σ maps HeartClinic and Clinic on the same concept in S_2, say CardiacH, then σ is not conflict-free. This is a direct consequence of the simple fact that every CFM must be injective.

Lemma 4 Let S_1 and S_2 be two schemas with disjoint sets of identifiers Σ_1 and Σ_2. Let σ be a CFM from Σ_1 into Σ_2. Then, σ is injective, i.e., for all identifiers A, B in the domain of σ, $S_2 \models \sigma(A) \equiv \sigma(B)$ implies $S_1 \models A \equiv B$.

PROOF. Assume that σ is not injective. In this case, there are identifiers A, B such that $S_2 \models \sigma(A) \equiv \sigma(B)$ but $S_1 \models A \not\equiv B$. Thus, there exists a model I of S_1 such that $A^I \neq B^I$. In particular, $I \in I(S_1)$. Since σ is a CFM, we know $I(S_1) = I(S_1 \cup S_2 \cup \mathcal{A}_\sigma)[\Sigma_1]$. Thus, there exists a model I' of $S_1 \cup S_2 \cup \mathcal{A}_\sigma$ that coincides with I on Σ_1. Then, since $A \equiv \sigma(A)$ and $B \equiv \sigma(B)$ are assertions in $\mathcal{A}_\sigma$, and $A^I \neq B^I$, we know $\sigma(A)^{I'} \neq \sigma(B)^{I'}$, a contradiction to the fact that I' is a model of S_2 and $S_2 \models \sigma(A) \equiv \sigma(B)$. ∎

It is also easy to see that CFMs must preserve subsumption relationships. For example, if σ maps Clinic on CardiacH and HeartClinic on Hospital, then σ is not conflict-free.

Lemma 5 Let S_1 and S_2 be two schemas with disjoint sets of identifiers Σ_1 and Σ_2. Let σ be a CFM from Σ_1 into Σ_2. Then, σ preserves subsumption relationships, i.e., for all concept descriptions C, D over $\mathsf{domain}(\sigma)$: $S_1 \models C \sqsubseteq D$ iff $S_2 \cup \mathcal{A}_\sigma \models C \sqsubseteq D$ (which is equivalent to $S_2 \models \sigma(C) \sqsubseteq \sigma(D)$).

PROOF. Assume, there exist concept descriptions C, D over $\mathsf{domain}(\sigma)$ with $S_1 \models C \sqsubseteq D$ but $S_2 \cup \mathcal{A}_\sigma \not\models C \sqsubseteq D$. Then, there exists $I \models S_2 \cup \mathcal{A}_\sigma$ with $C^I \not\subseteq D^I$. This means that I cannot be extended to a model I' of S_1 since otherwise $C^I = C^{I'} \subseteq D^{I'} = D^I$ in contradiction to the assumption. Thus, $I \in I(S_2) \setminus I(S_1 \cup S_2 \cup \mathcal{A}_\sigma)[\Sigma_2]$. However, since σ is conflict-free, we know $I(S_2) = I(S_1 \cup S_2 \cup \mathcal{A}_\sigma)[\Sigma_2]$, in contradiction to the previous statement.

Conversely, assume that there are concept descriptions C, D over $\mathsf{domain}(\sigma)$ with $S_2 \cup \mathcal{A}_\sigma \models C \sqsubseteq D$ but $S_1 \not\models C \sqsubseteq D$. Then, analogously to the previous case one can derive a contradiction to the fact that σ is conflict-free. ∎

These lemmas show that conflict-freeness rules out some unintuitive mappings. Still, due to the restricted structural information provided by schemas, it is in general not possible to get rid of all undesirable mappings. In Example 3 the mapping

$$\begin{aligned} \mathsf{Clinic} &\mapsto \mathsf{Hospital} \\ \mathsf{HeartClinic} &\mapsto \mathsf{CardiacH} \\ \mathsf{PsychiatricClinic} &\mapsto \mathsf{PsychiatricH} \end{aligned}$$

cannot be distinguished from the one that switches the images for $\mathsf{HeartClinic}$ and $\mathsf{PsychiatricClinic}$. Both mappings are conflict-free, but only the former is the intended one. This is due to the fact that $\mathsf{CardiacH}$ and $\mathsf{PsychiatricH}$ are structurally indistinguishable. In order to formalize this, we need the following notation: Given identifiers A, B and a schema S, define $S[A \leftrightarrow B]$ to be a schema that is obtained from S by simultaneously substituting A with B and B with A, i.e., A and B are switched in S.

Definition 6 Let S be a schema and let A, B be identifiers in S of the same type, i.e., both are concept names or both are roles. We call A and B *indistinguishable* in S ($A \approx_S B$ for short) iff $S \equiv S[A \leftrightarrow B]$, i.e., $I(S) = I(S[A \leftrightarrow B])$.

According to this definition, $\mathsf{CardiacH}$ and $\mathsf{PsychiatricH}$ are indistinguishable in S_2. We note that $\approx_S$ induces an equivalence relation on the identifiers of S. The following theorem shows that CFMs cannot differentiate between indistinguishable identifiers. To state the theorem we need some notation: Let S_1 and S_2 be two schemas with disjoint sets of identifiers Σ_1 and Σ_2. Let σ be a CFM from Σ_1 into Σ_2, and let A, B be indistinguishable identifiers in S_2 such that A, B belong to the image of σ. Then, define $\sigma' := \sigma[A \leftrightarrow B]$ to coincide with σ except that $\sigma'(A') := B$ for all A' with $S_2 \models \sigma(A') \equiv A$ and $\sigma'(B') := A$ for all B' with $S_2 \models \sigma(B') \equiv B$. Thus, roughly speaking, σ' is obtained from σ by switching A and B (modulo equivalence classes).

Theorem 7 [6] In the situation just described, $\sigma[A \leftrightarrow B]$ is a CFM from Σ_1 into Σ_2.

The theorem shows that in the presence of indistinguishable identifiers it is not possible for a tool which uses purely structural information to come up with *the* intended CFM. Instead, one needs to resort to heuristics based on identifiers, or to ask the user in order to choose between indistinguishable identifiers. Thus, as a first answer to the questions raised at the beginning of this section, one can say that indistinguishable identifiers place a limit on the purely structural approach to integration.

One can go a step further and not only relate single identifiers to each other but sets of identifiers that structurally cannot be distinguished from other sets of identifiers.

Before formally defining this generalized relationship, we illustrate the new notion by the following example:

Example 8 Let the schema S be defined as follows:

$$\begin{array}{rcl} \mathsf{Patient} & \sqsubseteq & \forall \mathsf{disease}.\mathsf{Disease} \\ \mathsf{Heart\text{-}Patient} & \sqsubseteq & \mathsf{Patient} \\ \mathsf{Employee} & \sqsubseteq & \forall \mathsf{address}.\mathsf{Address} \\ \mathsf{Boss} & \sqsubseteq & \mathsf{Employee} \\ \mathsf{Patient} & \sqsubseteq & \neg \mathsf{Employee} \end{array}$$

Obviously, switching the identifiers describing the patient and the ones for employees, i.e., switching Patient and Employee, disease and address, Disease and Address, Heart-Patient and Boss leads to an equivalent schema. We shall call these two sets indistinguishable.

For the formal definition we need some notation: Let S be a schema with signature Σ and let φ be a partial, injective mapping from Σ into Σ, where the domain $\mathsf{domain}(\varphi)$ of φ and the image $\mathsf{image}(\varphi)$ of φ are disjoint. Then, define $S[\varphi]$ to be the schema obtained from S by simultaneously switching A and $\varphi(A)$ in S for all $A \in \mathsf{domain}(\varphi)$. In particular, if the domain of φ is $\{A\}$ and $\varphi(A) = B$, then $S[\varphi] = S[A \leftrightarrow B]$.

Definition 9 Let S be a schema with signature Σ. Furthermore, let φ be a partial, injective mapping from Σ into Σ such that the domain $\mathsf{domain}(\varphi)$ and the image $\mathsf{image}(\varphi)$ of φ are disjoint. Then, φ is called *indistinguishable* in S iff $S \equiv S[\varphi]$.

Note that as long as subsumption modulo a schema is decidable, one can decide whether φ is indistinguishable in S, since in order to check $S \equiv S[\varphi]$ one only has to check whether every axiom in S is implied by $S[\varphi]$ and vice versa. Consequently, the set of all indistinguishable φ's can be computed.

Theorem 7 can now be generalized in the obvious way as follows. First, we extend the definition of $\sigma[A \leftrightarrow B]$ to $\sigma[\varphi]$. Let σ be a CFM from Σ_1 into Σ_2 for the schemas S_1 and S_2. Moreover, let φ be an indistinguishable mapping for S_2 and assume that both the domain and the image of φ belong to the image of σ. Then, define $\sigma' := \sigma[\varphi]$ to coincide with σ except that if $\varphi(A) = B$, then $\sigma'(A') := B$ for all A' with $S_2 \models \sigma(A') \equiv A$ and $\sigma'(B') := A$ for all B' with $S_2 \models \sigma(B') \equiv B$.

Theorem 10 [6] In the situation just described, $\sigma[\varphi]$ is a CFM from Σ_1 into Σ_2.

Empirical results. In order to get an idea of how much structural information is contained in a real knowledge base, and thus, how far a structural approach to integration can get, we have applied the notion of indistinguishable identifiers (Definition 6) and CFMs to the GALEN medical knowledge base [22], which contains about 2700 concept names, 400 role names, general inclusion axioms ($C \sqsubseteq D$), and a role hierarchy ($r \sqsubseteq r'$).

It turned out that about 30% (25%) of the 2700 concept names (400 role names) belong to non-trivial $\approx$-equivalence classes, i.e., classes containing more than one concept name (role name). The biggest of these classes contains 29 concept names (9 role names). This means that about two thirds of the concept names in GALEN can be structurally distinguished from all other concepts in the ontology. Thus, on the one hand, the phenomenon of indistinguishability cannot be neglected; on the other hand the empirical results reveal GALEN's rich structure.

This is also confirmed when looking for CFMs, for which we have considered the following idealized situation: Suppose one is given two copies of the GALEN ontology (say, one with English identifiers and a copy with Chinese identifiers). For those unfamiliar with one of these languages, the task of finding the "identity mapping", which maps the English identifiers to the corresponding Chinese ones, must rely solely on structural information. The question is to what degree CFMs resemble the identity mapping. It has turned out that every total CFM coincides with the identity mapping on about 50% of the concept names. Hence, a considerable part of the meaning of GALEN terms is encoded as structural information.

4 Conclusion

We considered the problem of merging two DL terminologies by finding a set of assertions that relate them. We were particularly concerned with obtaining some formal criteria for judging the appropriateness of the IAs, as a foundation for tools that would find such candidate assertions by processing large knowledge bases, where "manual" integration is not an option.

We found that the notion of CFM, introduced by Biskup and Convent [5] in the context of database view merging, captures some of the intuition concerning the role of integration assertions. It is entirely possible that in practical situations one might encounter IAs that are not conflict-free, which means that new knowledge is being added in the process of integration. Assuming that the original terminologies are accurate views of overlapping parts of the same universe of discourse[1], the only possibility is that one or both of them are incomplete. Therefore, we suggest that part of the integration process may involve revisiting the original terminologies, and adding extra material, resulting in the IAs being conflict-free. This update is motivated by our assumption that the original terminologies persist independently even after the alignment process.

We have observed however that the task of integration does not reduce to that of finding CFMs, because almost always there are multiple mappings, only one of which is intuitively preferable. Certainly, this observation is not surprising on its own. The distinguishing feature of our work is rather that, unlike other contributions to (semi-)automatic integration, we tried to get a better understanding of this phenomenon in that we explored the limits of structural integration both theoretically and empirically. Our theoretical investigations revealed that part of the problem are indistinguishable identifiers: identifiers whose specifications are not sufficiently complete, since only their lexical token distinguishes them from other identifiers. Empirical analysis of the GALEN knowledge base showed the extent to which such identifiers occur even in the best of knowledge bases. Also, we used the merging of GALEN with a copy of itself (but with identifiers renamed) as a test of the effect of indistinguishable identifiers. In both cases, the effect was not negligible.

This makes clear that outside help will be needed to make the right choices, which in the end will involve human decisions. However, in order to make the list of choices small enough, heuristics will be needed. Although heuristics based on purely structural information can support this task, the applicability of the structural approach has provable limits as shown here.

References

[1] F. Baader and R. Küsters. Matching in description logics with existential restrictions. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000)*, pages 261–272, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[2] F. Baader, R. Küsters, A. Borgida, and D. McGuinness. Matching in description logics. *Journal of Logic and Computation*, 9(3):411–447, 1999.

[1] If this assumption does not hold, than we must be prepared to find inconsistencies between the terminologies, which leads to an entirely new set of formal problems related to "theory merging/amalgamation" (e.g., [23]).

[3] F. Baader and P. Narendran. Unification of concept terms in description logics. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 331–335. John Wiley & Sons Ltd, 1998.

[4] C. Batini, M. Lenzerini, and S.B. Navathe. A comparative analysis of methodologies for database schema integration. *Computing Surveys*, 18(4):323–364, 1986.

[5] J. Biskup and B. Convent. A Formal View Integration Method. In Carlo Zaniolo, editor, *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, pages 398–407, Washington, D.C., USA, 1986. ACM Press.

[6] A. Borgida and R. Küsters. What's not in a name? Initial Explorations of a Structural Approach to Integrating Large Concept Knowledge-Bases. Technical Report DCS-TR-391, Rutgers University, USA, 1999. Available via ftp://ftp.cs.rutgers.edu/pub/technical-reports/.

[7] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description Logic Framework for Information Integration. In A.G. Cohn, L.K. Schubert, and S.C. Shapiro, editors, *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR-98)*, pages 2–13. Morgan Kaufmann, 1998.

[8] S. Castano and V. De Antonellis. A Schema Analysis and Reconciliation Tool Environment for Heterogeneous Databases. In *1999 International Database Engineering and Applications Symposium (IDEAS 1999)*, pages 53–62, Montreal, Canada, 1999. IEEE Computer Society.

[9] T. Catarci and M. Lenzerini. Representing and Using Interschema Knowledge in Cooperative Information Systems. *Journal of Intelligent and Cooperative Information Systems*, 2(4):375–398, 1993.

[10] B. Convent. Unsolvable problems related to the view integration approach. In G. Ausiello and P. Atzeni, editors, *Lecture Notes in Computer Science*, volume 243, pages 141–156, Rome, Italy, September 1986. Springer.

[11] F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The Complexity of Concept Languages. *Information and Computation*, 134:1–58, 1997.

[12] L. Ekenberg and P. Johannesson. Conflictfreeness as a Basis for Schema Integration. In S. Bhalla, editor, *6th International Conference on Information Systems and Data Management*, pages 1–13, Bombay, 1995. Springer-Verlag.

[13] W. Gotthard, P.C. Lockemann, and A. Neufeld. System-Guided View Integration for Object-Oriented Databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(1):1–22, 1992.

[14] R. Hull. Relative Information Capacity of Simple Relational Database Schemata. *SIAM Journal on Computing*, 15(3):856–886, 1986.

[15] V. Kashyap. *Information Brokering over Heterogeneous Digital Data - A Metadata-based Approach*. PhD thesis, Rutgers University, New Brunswick, USA, 1997.

[16] K. Knight and S.K. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)*, volume 1, pages 773–778, Seattle, WA, USA, 1994. AAAI Press.

[17] D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000)*, pages 483–493, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[18] E. Mena, V. Kashyap, A.P. Sheth, and A. Illarramendi. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. In *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, pages 14–25, Brussels, Belgium, 1996. IEEE-CS Press.

[19] R.J. Miller, Y.E. Ioannidis, and R. Ramakrishnan. The Use of Information Capacity in Schema Integration and Translation. In R. Agrawal, S. Baker, and D.A. Bell, editors, *19th International Conference on Very Large Data Bases (VLDB'93)*, pages 120–133, Dublin, Ireland, 1993. Morgan Kaufmann.

[20] F.N. Noy and M.A. Musen. SMART: Automated Support for Ontology Merging and Alignment. In *Proceedings of the Twelfth Workshop on Knowledge Acquisition (KAW'99)*, Banff, Canada, 1999.

[21] L. Palopoli, D. Sacca, and D. Ursino. Semi-Automatic Technique for Deriving Interscheme Properties from Database Schemes. *Data and Knowledge Engineering*, 30(3):239–273, 1999.

[22] The GALEN Project. See http://www.cs.man.ac.uk/mig/galen/.

[23] V.S. Subrahmanian. Amalgamating Knowledge Bases. *ACM Transactions on Database Systems (TODS)*, 19(2):291–331, 1994.