Practical Context Transformation for Information System Interoperability

Holger Wache, Heiner Stuckenschmidt

Center for Computing Technologies
University of Bremen, Germany
{wache, heiner}@tzi.de

Abstract. This paper discusses the use of contextual reasoning, i.e. context transformation for achieving semantic interoperability in heterogeneous information systems. We introduce terminological contexts and their explication in terms of formal ontologies. Using a real-world example, we compare two practical approaches for context transformation one based on transformation rule, the other of re-classification of information entities in a different terminological context. We argue that both approaches supplement each other and develop a unifying theory of context transformation. A sound and complete context transformation calculus is presented that covers both transformation approaches.

1 Introduction

Mediators [5] are middleware components that provide a flexible integration of several information systems such as database management systems, geographical information systems, or the World Wide Web. A mediator combines, integrates, and abstracts the information provided by the sources [24] tackling the same problems which are discussed in the federated database research area, i.e. *structural heterogeneity* (schematic heterogeneity) and *semantic heterogeneity* (data heterogeneity) [15]. Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content and its semantics of an information item. In rule–based mediators [6], rules are mainly designed in order to reconcile structural heterogeneity. Discovering semantic heterogeneity problems and their reconciliation play a subordinate role. But for the reconciliation of the semantic heterogeneity problems, the semantical level also has to be considered [11, 4, 14]. *Contexts* are one possibility to capture this semantical level. A context [13] contains "meta data relating to its meaning, properties (such as its source, quality, and precision), and organization" [19]. A value has to be considered in its context and may be transformed into another context (so–called *context transformation*).

In this paper, we review two approaches to the implementation of context transformation in mediators, namely functional context transformation and context transformation by re-classification. We discuss their use for providing semantic interoperability among heterogeneous information systems. We propose a unifying theory of practical context transformation that covers both approaches and present a sound and complete context transformation calculus. The paper is structured as follows: Section 2 introduces the problem of semantic heterogeneity and motivates the use of contextual knowledge. In section 3 we illustrate an integration process using an example from a real application. The use of the different transformation approaches is discussed in section 4. In section 5 we present the unifying theory of context transformation and the transformation calculus including sketches of the soundness and completeness proofs.

2 Context, Ontologies and Information Systems

In principle there are two possible solutions to achieving semantic interoperability between heterogeneous information systems [7]: the tight coupling strategy that creates a new information system with a unified semantics and the loose coupling approach that does not touch the individual semantics and instead provides transformations on a semantic level. There are strong arguments in favor of the loose coupling approach [12]. First of all the use of individual semantics allows small representations and efficient reasoning within the individual system. Second, the semantics in a multi-context system is much more flexible and can be used to handle inconsistencies that would become threatening when trying to create a single context with a global semantics.

2.1 Contexts and Semantic Heterogeneity

In order to achieve semantic interoperability in a heterogeneous information system, the *meaning* of the information that is interchanged has to be understood across the systems. Semantic conflicts occur, whenever two contexts do not use the same interpretation of the information. Goh identifies three main causes for semantic heterogeneity [7].

- *Confounding conflicts* occur when information items seem to have the same meaning, but differ in reality, e.g. due to different temporal contexts.
- *Scaling conflicts* occur when different reference systems are used to measure a value. Examples are different currencies or marks.
- *Naming conflicts* occurs when naming schemes of information differ significantly. A frequent phenomenon is the presence of homonyms and synonyms.

It has been argued that semantic heterogeinity can be resolved by transforming information from one context into another. In [18] and [7] context transformation methods are developed. The scope of these approaches in mainly on the conversion of different scaling conflicts. In our work we address the problem of providing practical solutions for the context transformation problem that is not only capable of converting between different scales, but also covers the transformation of application-specific vocabularies. We therefore argue for a semantic interoperability approach that is based on transformations between individual terminological contexts.

2.2 Ontologies as Contextual Information

Ontologies have set out to overcome the problem of implicit and hidden knowledge by making the conceptualization of a domain explicit. This corresponds to one of the definitions of the term ontology most popular in computer science [8]: *"An ontology is an explicit specification of a conceptualization."* An ontology is used to make assumptions about the meaning of a term available. It can also be seen an an explication of the context a term is normally used in.

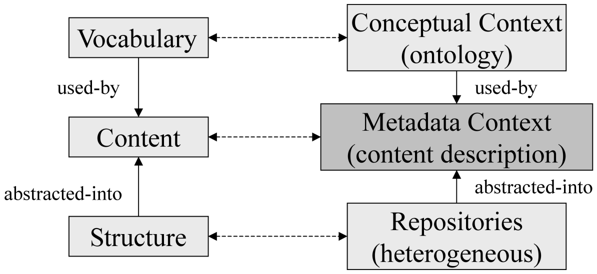

Fig. 1. The Role of Context in Information Systems Interoperability. (adapted from [12])

Kashyap and Shet [12] discuss the role of contexts and ontologies for semantic interoperability (compare figure 1). According to their view, contexts are used to abstract from the content of an information repository. So-called *metadata contexts* describe the information content of a repository and therefore allow to decide whether a repository contains relevant in formation. Additionally *conceptual contexts* are introduced. A conceptual context is an ontology that defines the meaning of terms used in the metadata context and the repository. While Kashyap and Shet define relationships between the ontologies, our approach relies on the use of shared basic vocabulary that is used to derive inter-ontology relationships.

We propose to use formal ontologies in order to capture and explicate the assumptions made by each context, because they can be used as a basis for automatic translations between vocabularies that preserve the intended meaning of the translated vocabulary.

3 Context-Based Semantic Integration

We illustrate the need for context modeling and transformation by a real–world example which also serves to illustrate our approach. Two sources — `CORINE` and `ATKIS` — provides geological information.

The first source `CORINE` [3] stores its data in two tables[1]. The first table is called `clc_ns2`. Every entry represents one geological item. `clc_ns2` contains the attributes `CLC_NS2_ID` (identifier), `AREA` (size in ha), and `NS` (classification). Especially the last attribute `NS` refers to catalog, wherein all items are classified. In `CORINE`, the catalog contains more than 64 concepts. The second table `clc_ns2_pol` stores polygons describing the area of an item. The attributes are `CLC_NS_ID` (reference to `clc_ns2`), `VERT_ID` (identifier of a vertices), and `NEXT_V_ID` (identifier of the following vertices).

In the second source `ATKIS` [1] a geological item is stored in one table `atkisf` with the attributes `id`, `fl` (size in m^2), and `folie` (classification). Analogously to `CORINE` the last attribute `folie` refers to a classification catalog containing more than 250 terms.

But the catalogs of `CORINE` and `ATKIS` are different. Further, both catalogs underly different conceptualizations. The task of this example is that the data of `CORINE` database has to be converted in the `ATKIS` database. Of course, this transformation can be viewed as a special case of an integration task demonstrating all the problems which can occur. Besides the obvious structural heterogeneity problems, the main problem relies on the reconciliation of the semantic heterogeneity: both geological information sources classify the common areas in different catalogs. A mediator system that tries to query information from one system in terms of the other will fail or return wrong results, because it will not be able to unify the land-use classifications and will not recognize that the returned size of an area refers to a different scale. Consequently, the classification of the `CORINE` has to be converted into the `ATKIS` catalog. Moreover, the size has to be converted according to their different currencies. Both conversions are the challenge for the semantic integration and are handled by the both kinds of context transformation.

3.1 A Minimal Modeling Language

In order to capture the semantics of the different land-use classifications used in the systems we want to integrate, we have to describe an ontologies of land-use classes. We use a description logic in order to build these ontologies. The features of this language are described below.

Description logics are a family of logic-based representation formalisms that cover a decidable subset of first-order logic. Description logics are mostly used to describe terminological knowledge in terms of concepts and binary relation (slots) between concepts that can be used to define a concept term by necessary and sufficient conditions that have to be fulfilled by all instances of the concept. We use a minimal description logic that consist of conjunction, disjunction, negation of as well as existentially and universally qualified range restrictions on slots. These language elements can be used to describe concept expressions with the following syntax and semantics:

---

[1] For readability reasons the tables of both sources are simplified. Some attributes are omitted.

syntax	**semantics**
concept-name	$\mathcal{I}(C) \subseteq \mathcal{D}$
top	$\mathcal{I}[\textbf{top}] = \mathcal{D}$
bottom	$\mathcal{I}[\textbf{bottom}] = \emptyset$
(**and** concept$^+$)	$\mathcal{I}[(\textbf{and}\, C_1\, C_2)] = \mathcal{I}[C_1] \cap \mathcal{I}[C_2]$
(**or** concept$^+$)	$\mathcal{I}[(\textbf{or}\, C_1\, C_2)] = \mathcal{I}[C_1] \cup \mathcal{I}[C_2]$
(**not** concept)	$\mathcal{I}[(\textbf{not}\, C)] = \mathcal{D} - \mathcal{I}[C]$
(**some** role concept)	$\mathcal{I}[(\textbf{all}\, r\, C)] = \{x \in \mathcal{D} \mid \forall y \in \mathcal{D} : \langle x, y\rangle \in \mathcal{I}[r] \Rightarrow y \in \mathcal{I}[C]\}$
(**all** role concept)	$\mathcal{I}[(\textbf{some}\, r\, C)] = \{x \in \mathcal{D} \mid \exists y \in \mathcal{I}[C] : \langle x, y\rangle \in \mathcal{I}[r]\}$

Class expressions are used to define concepts. A concept is defined using the keyword **concept** followed by the name of the concept and a concept expression that restricts the set of entities belonging to the concept to a subset of the whole domain of discourse. The meaning of a concept definition is defined by an interpretation. The Tuple $< \mathcal{D}, \mathcal{I} >$ is an interpretation, if $\mathcal{D}$ is a domain and $\mathcal{I}$ is an extension function that maps concept names into subsets of $\mathcal{D}$ and role names into $\mathcal{D} \times \mathcal{D}$. Using this interpretation, the semantics of the language constructs is given by the equations in the table above. This Tarskian style semantics offers a formal framework for the comparison of different terminologies.

3.2 The Integration Process

Step 1:Authoring of Shared Terminology Our approach relies on the use of a shared terminology in terms of properties used to define different concepts. This shared terminology has to be general enough to be used across all information sources to be integrated but specific enough to make meaningful definitions possible. The top-level concept *parcel* is defined below:

(**concept** parcel (**and**
    (**all** *ground* ground-type)
    (**all** *coverage* structure)
    (**all** *cultivation* plant)
    (**all** *vegetation* plant)
    (**all** *use* use-type)))

For the given integration task the shared terminology mainly consists of ontologies that define concepts a parcel can be related to, namely ground types, artificial structures built on a parcel, different kinds of plants that may grow on a parcel and general types of land use.

Step 2: Annotation of Information Sources Once a common vocabulary exists, it can be used to annotate different information sources. In this case annotation means that the inherent concept hierarchy of an information source is extracted and each concept is described by necessary and sufficient conditions using terms from the vocabulary defined in the share terminology. The result of this annotation process is an ontology that contains a definition of the terminological context. The meaning of land-use classes from both classifications is formally defined by further restricting the range of the slots attached to the general parcel concept. Here is an example:

(**concept** broad-leaved-forest (**and** parcel
    (**all** *coverage* no-stuctures)
    (**all** *ground* land)
    (**all** *vegetation* (**or** trees shrubs))
    (**some** *vegetation* broad-leaved-trees)))

The above example is taken form one of the entries in our CORINE data-set used in the case study. This entry is classified as 'broad-leaved-forst' which is a subclass of parcel that can be identified by the absence of water, a lack of artificial structures and a vegetation that may consist of trees and shrubs where some of the vegetation consists of broad-leaved trees.

We use so-called templates to assign a contextual description to data structures in a repository. A template is an fifth-ary predicate:

$$T = \langle name, context, type, value \rangle @source$$

A template has a *name*, a *context* addressing the semantics of the concept. The name of the context refers to the corresponding ontology that explicated the terminological context. Further elements of a template are: *type* determining the data type, the *value* referencing the information item itself, and the last identifier *source* denoting which source the template belongs to. The value can be a simple value, e.g. a number, or a string, or a list of attributes. An *attribute* consists of a name and a template the attribute refers to. In the last case the type is `complex`. In case of simple values the type slot contains the basic data type.

Templates with attributes can represent tables (relations) in a relational data structure model. The attributes of the template are the attributes of the relation. The value of the template attributes are templates encapsulating the basic data types of the relation attributes[2]. An example template for the `ATKIS` table is given below. The template contains variables and therefore describes a set of instances found in the database.

```
<atkisf,?LA,complex, {
  id ->     <id,?LI,string,?ID>,
  fl ->     <fl,?LS,real,?S>,
  folie ->  <folie,?LC,int,?C>
  }>@ATKIS
```

---

[2] For readability reasons the source is omitted in the nested templates

Step 3: Semantic Translation of Information Entities The purpose of the steps described above was to lay a base for the actual translation step. The existence of a terminological context model for all information sources to be integrated enables a translation method to work on the contextual knowledge. Two different types of translation by context transformation have been investigated:

- Rule-based functional transformation [23]
- Classification-based transformation [20].

We argued that these two kinds of context transformation supplement each other in the sense that functional transformation is well suited to resolve scaling conflicts while classification based transformation can be used to resolve non-trivial naming conflicts [21]. The transformation step is discussed in more detail in the next section.

4 Context Transformation

A conceptual model of the context of each information source builds a basis for an integration on the semantic level. We call this process context transformation, because we take the information about the context of the source (in our case `CORINE`) and re-interpret this information in the terms of a target (`ATKIS`) providing a new context description for that entity within the new information source. We compare two different approaches for context transformation namely rule-based context transformation and context transformation based on classification and show how these two approaches can be used to integrate the example data.

4.1 Context Transformation with Rules

Context Transformation Rules (CTR's) define a context transformation between two templates. Operationally they has to *exchange* information. More precisely CTR's replace one template by another. An important aspect of CTR's is that they can be applied to templates which are nested in the structure of a top–level template i.e. to an template in an attribute. This aspect simplifies the formulation of CTR's and improves the scalability and the flexibility of context transformation. A CTR is represented as follows:

$$a \rightsquigarrow \overline{b} \leftarrow b_1, ..., b_n$$

The head of the rule defines the relation $\rightsquigarrow$ — the so called context transformation relation. The relation describes which template a can be transformed into template b. The other terms in the body $b_1, ..., b_n$ are required to support or to restrict the context transformation. Normally the body terms are expressions but can also be further templates, e.g. if further information for the context transformation is needed.

We illustrate the use of CTR's in our example. The surfaces in `ATKIS` and `CORINE` are stored with different measures of size, namely square-meters and hectares. Therefore the surface value of `CORINE` can not be copied but has to be converted dividing the number of square-meters by the factor 10000 . The conversion is done during the context transformation. The appropriate CTR looks like:

```
<?N,surface,?T,?V_HA>@CORINE -> <?N,surface,?T, ?V_M2>@ATKIS
     :- ?V_M2 := ?V_HA / 10000.
```

In the templates the contexts and the sources are specified but the names, types, and values are replaced by variables. Such a CTR is applicable to all templates with the specified context, where an appropriate substitution for the variables exists, including those templates which are hidden in the structure, e.g. to the template `AREA` stored in the attribute `AREA` of template `clc_ns2`.

4.2 Context Transformation by Classification

Context transformation rules cover a lot of semantic heterogeneity problems, i.e. we were able to integrate the diverging measures and units e.g. used to determine the size of area. However, the integration of the different kinds of type information is more difficult. Representing this by CTR's would lead to very large set of trivial CTR's. For maintainability and scalability reasons, this demands for more sophisticated mechanisms for context transformation. We solve this problem by automatically deriving type transformation rules using the inference capabilities of the description logics we use to represent conceptual contexts.

The main inference mechanism used in description logics is subsumption checking. A concept is said to subsume another concept, if the membership of the latter implies membership in the former. Following the semantics defined above the subsumption relation between two concepts is equivalent to a subset relation between the extensions of the concept definition. Given two concept definition A and B subsumption is tested by checking if $A \sqsubseteq B \iff \mathcal{I}[(\mathbf{and}\, B\, (\mathbf{not}\, A))] = \emptyset$. Subsumption checking can be seen as a special classification method, because it returns a list of classes B_i (concepts) an member of a given concept A belongs to. In terms of subsumption reasoning a context transformation task can be defined as follows:

Let $\mathcal{S}$ and $\mathcal{T}$ be two terminological contexts represented by sets of concept definitions with subsumption relations $\sqsubseteq_{\mathcal{S}}$, $\sqsubseteq_{\mathcal{T}}$ and concept membership relations $\in_{\mathcal{S}}$, $\in_{\mathcal{T}}$. Let further S be a concept from one terminological context $\mathcal{S}$ ($S \in \mathcal{S}$). Then the transformation of a data set s from context $\mathcal{S}$ into the context $\mathcal{T}$ is described by

$$(s \in_{\mathcal{S}} S \implies s \in_{\mathcal{T}} T) \iff (S \sqsubseteq_{\mathcal{T}} T) \quad (1)$$

In general, it is not decidable, whether the condition ($S \sqsubseteq_{\mathcal{T}} T$) holds, because the subsumption relation is only defined for the context $\mathcal{T}$ while the concept definition S is taken from context $\mathcal{S}$ and is therefore used by a different subsumption relation. At this point, the shared vocabulary comes in to play. Provided, that the concepts from both contexts are defined using the same basic vocabulary, we get a unified subsumption relation defined as:

$$\sqsubseteq \, = \, \sqsubseteq_{\mathcal{S}} \, \cup \, \sqsubseteq_{\mathcal{T}} \tag{2}$$

We can compute $\sqsubseteq$ using available subsumption reasoner that support the language. The result of the context transformation is now given by

$$class_{\mathcal{T}}(S) = \{T_i | S \sqsubseteq T_i \wedge T_i \in \mathcal{T} \wedge \, \nexists V \in \mathcal{T} : V \sqsubset T_i\}.$$

Having determined the result set $class_{\mathcal{T}}(S)$ we can use its elements for defining a set of new context transformation rules in the following way:

$$s \rightsquigarrow t \leftarrow s \in_{\mathcal{S}} S, t \in_{\mathcal{T}} T, T \in class_{\mathcal{T}}(S)$$

These new rules supplement the rule base used for context transformation and integrate classification-based and rule-base transformation.

In the case study we used the FaCT reasoner [10] in order to compute the unified subsumption relation for the ATKIS an the CORINE context. The derived direct subsumers of 'broad-leaved-forest' from the CORINE context are FORESTS-AND-SEMI-NATURAL-AREAS and FORESTS. In the case of 'broad-leaved-forest' we also get the correct result for the ATKIS context. The direct subsumers from the target hierarchy are: VEGETATION-AREA and FOREST-AREA.

5 A Unifying Model of Context Transformation

In the previous section we describe the two kinds of context transformations from an informal point of view and give an impression how context transformation rules are represented and context classification is defined by example. In this section we present a formal context transformation calculus which explains how context transformation rule and context classification can be implemented. Moreover the calculus also shows how the two kinds of context transformation can be combined.

5.1 A Theory of Context Transformation

As shown in the examples above the context transformation is indicated by the binary relation $\rightsquigarrow$. The expression $s \rightsquigarrow t$ means that term s is context transformed into the term t. We give am axiomatic specifying the meaning of $\rightsquigarrow$. The set of axioms establish a unifying theory $\mathcal{T}_C$ of context transformation. They require that for all values x_i, y_i and z the following implications hold:

$$\begin{aligned}
& x \rightsquigarrow x && \text{(CT1)}\\
& x \rightsquigarrow y \wedge y \rightsquigarrow z \implies x \rightsquigarrow z && \text{(CT2)}\\
\forall f: \; & x \rightsquigarrow y \implies f(...,x,...) \rightsquigarrow f(...,y,...) && \text{(CT3)}\\
\forall P: \; & x_1 \rightsquigarrow y_1 \wedge ... \wedge x_n \rightsquigarrow y_n \wedge P(x_1,...,x_n) \implies P(y_1,...,y_n) && \text{(CT4)}\\
& \quad \text{where } P \neq \rightsquigarrow &&
\end{aligned}$$

The first two axioms show the reflexivity and the transitivity of context transformation relations. The next two axioms are known as the substitution axioms and allows the application of context transformation on sub terms. The first axiom allows the exchange of x by y in the functional term f. The second substitution axioms substitutes all arguments $x_1, ..., x_n$ by $y_1, ..., y_n$ in the n-ary predicate P.

If the theory of context transformation is compared to other axiomatization the similarity to the equality theory is conspicuous. Only the symmetry is missed in the context transformation theory. The $\rightsquigarrow$ is obviously not symmetric because a term s can be transformed into a term t but not vice versa (e.g. if during the transformation s is abstracted to t). On the other hand, in general $\rightsquigarrow$ is not asymmetric because sometimes a term t can be retransformed in s (e.g. if there is a bijective functional dependency).

The substitution axioms are formulated in second order predicate logic. It is possible to reformulate these axioms in first order predicate logic by give one axiom for every functional symbol f and every predicate symbol P. But this reformulation would lead to an infinite number of axioms which practical is not be possible to handle.

5.2 A Transformation Calculus

In principle the theory of context transformation can be implemented using standard logical inference, i.e. deduction. However this turns out to be extremely inefficient. We therefore develop a more efficient calculus and show that it follows the general theory.

Due to the similarity to equality theory the calculus is similar to a term reduction system. The difference is that a context transformation is only applicable on the right side of the goal $\leftarrow s \rightsquigarrow t$ (reflecting the missing of the symmetry). The transformation eliminates the goal $\leftarrow s \rightsquigarrow t$ if the left side is unifiable with the right side of $\leftarrow s \rightsquigarrow t$ (cf. the reflexivity axiom). The two inference rules of the context transformation calculus are:

Definition 1. *Let $\mathcal{CT}$ be a set of context transformations and $\leftarrow s \rightsquigarrow t$ a goal state, then the following deduction rules apply:*

1. ***Context Transformation Inference Rule (CTIR)***
   Let $a \rightsquigarrow b \in \mathcal{CT}$ a context transformation and the term s contains at position p the sub term u ($s|_p = u$) then if there is a most general unifier σ exists in such a way that $\sigma u = \sigma a$ then

$$\frac{\begin{array}{c}\leftarrow s[u]_p \rightsquigarrow t \\ a \rightsquigarrow b\end{array}}{\leftarrow \sigma(s[b]_p \rightsquigarrow t)}$$

   The expression $s[b]_p$ means the replacement of the sub term at position p in term s by term b.

2. ***Elimination Inference Rule (EIR)***
   If there is a most general unifier σ in such a way that $\sigma s = \sigma t$ then the empty clause $\Box$ can be inferred. I.e. $\leftarrow s \rightsquigarrow t \vdash \Box$

The set of context transformation must be extended by the set of $f(s_1, ..., s_n) \rightsquigarrow f(s_1, ..., s_n)$ for each functional symbol f with arity n. These additional set of context transformation is needed for technical reasons, because without these context transformations the completeness (i.e. the lifting lemma) can not be proven. The addition of the set of tautologies is also known form the paramodulation principle where the set of equalities is extended by $f(s_1, ..., s_n) = f(s_1, ..., s_n)$. In practice these tautologies can be omitted.

5.3 Correctness and Completeness of the Calculus

For the calculus the correctness and the completeness can be proven but not the termination as the most logic-oriented calculus. For example one can easily define some rules with circles leading a never ending inference procedure.

The following theorems prove the correctness and the completeness with respect to the theory stretched by the axioms (see 5.1). The correctness ensures that for every (negated) goal for which a refutation exists the (positive) goal is a logical consequence of the set of context transformations.

Theorem 1. ***Correctness of the Context Transformation***
Let $\mathcal{CT}$ be a set of context transformations and $\leftarrow s \rightsquigarrow t$ a goal state. Assume that for $\leftarrow s \rightsquigarrow t$ there exists a refutation in n steps ($\mathcal{CT} \cup \{\leftarrow s \rightsquigarrow t\} \vdash \Box$). Then for every interpretation which satisfy $\mathcal{CT}$ and the context transformation axioms in $\mathcal{T}_C$, satisfy $s \rightsquigarrow t$.

$$\mathcal{CT} \cup \{\leftarrow s \rightsquigarrow t\} \vdash \Box \implies \mathcal{CT} \cup \mathcal{T}_C \models s \rightsquigarrow t$$

Due to the limit of paper length we only give a proof sketch of the theorem above:

Proof sketch. The prove is a induction over the length of the refutation. Suppose that the resolvent R of a deduction is a logical consequence. In the induction step ($n \rightarrow n + 1$) it is show that the goal G from which R is inferred is also a logical consequence depending on which inference rule of the calculus is applied. For example[3] if the CTIR is used the a context transformation $a \rightsquigarrow b$ is applied on $G = s[u]_p \rightsquigarrow t$ with $a = u$ resulting in $R = s[b]_p \rightsquigarrow t$. Every interpretation I which satisfies $a \rightsquigarrow b$ must also satisfy $s[a]_p \rightsquigarrow s[b]_p$ because I has to satisfy the substitution axioms (CT3). Applying the transitivity axiom (CT2) on $s[a]_p \rightsquigarrow s[b]_p$ and $R = s[b]_p \rightsquigarrow t$ leads to $s[a]_p \rightsquigarrow t$. Because $u = a$ it can be concluded that I also satisfies $s[u]_p \rightsquigarrow t = G$. The proves of the induction start for EIR are similar. ∎

---

[3] For brevity the substitution which normally is needed is omitted.

The completeness shows the reverse direction that for every goal which is a logical consequence of $\mathcal{CT}$ there exists a refutation for the (negated) goal. Here we only give the completeness result for terms without any variables (ground terms). In a lifting lemma this result can lifted to terms with variables.

Theorem 2. ***Completeness of the ground Context Transformation***
Let $\mathcal{CT}$ be a set of ground context transformations and $\leftarrow s \rightsquigarrow t$ a ground goal. If there exists no interpretation which satisfies $\mathcal{CT} \cup \{s \rightsquigarrow t\}$ and the context transformation axioms in $\mathcal{T}_C$ then there exists a refutation for $\mathcal{CT} \cup \{\leftarrow s \rightsquigarrow t\}$.

$$\mathcal{CT} \cup \mathcal{T}_C \models s \rightsquigarrow t \implies \mathcal{CT} \cup \{\leftarrow s \rightsquigarrow t\} \vdash \square$$

Proof sketch. We show that every terms from the minimal Herbrand model $M_\mathcal{P}$ is refutable. It can be proven that $M_\mathcal{P}$ is the union of $M_0 = \mathcal{CT}$ and $M_{n+1} = M_n \cup \{s \rightsquigarrow t | s \rightsquigarrow t \leftarrow \overline{B} \in \mathcal{T}_C \wedge \overline{B} \subseteq M_n\}$ for $n > 0$. By induction of n we prove that for every $s \rightsquigarrow t \in M_{n+1}$ there exists a refutation. By induction hypothesis for every $s_i \rightsquigarrow t_i \in M_n$ there already exists a refutation.

The main idea of this prove is that the refutations for $s_i \rightsquigarrow t_i$ are combined into one refutation for $s \rightsquigarrow t$ according to $s \rightsquigarrow t \leftarrow s_1 \rightsquigarrow t_1, ..., s_k \rightsquigarrow t_k$ With respect to the theory $\mathcal{T}_C$ there are four possibilities how $s \rightsquigarrow t$ can be constructed. Suppose that $s \rightsquigarrow t$ is generated by the transitivity axiom (CT2) $s \rightsquigarrow t \leftarrow s \rightsquigarrow u, u \rightsquigarrow t$. The two refutations of $s \rightsquigarrow u$ with length $m+1$ and $u \rightsquigarrow t$ with length $l+1$ can be combined in a refutation of length $m+l+1$ as following:

$$\begin{array}{l}
\leftarrow s \rightsquigarrow u \\
\quad a_1 \rightsquigarrow b_1 \\ \hline
\leftarrow s_1 \rightsquigarrow u \\
\vdots \\
\leftarrow s_{m-1} \rightsquigarrow u \\
\quad a_m \rightsquigarrow b_m \\ \hline
\leftarrow s_m \rightsquigarrow u \\ \hline
\square \ \text{(with } s_m = u) \\
\text{(appl. of EIR)}
\end{array}
\quad
\begin{array}{l}
\leftarrow u \rightsquigarrow t \\
\quad a_{m+1} \rightsquigarrow b_{m+1} \\ \hline
\leftarrow u_1 \rightsquigarrow t \\
\vdots \\
\leftarrow u_{m-1} \rightsquigarrow t \\
\quad a_{m+l} \rightsquigarrow b_{m+l} \\ \hline
\leftarrow u_l \rightsquigarrow t \\ \hline
\square \ \text{(with } u_l = t)
\end{array}
\quad \Longrightarrow \quad
\begin{array}{l}
\leftarrow s \rightsquigarrow t \\
\quad a_1 \rightsquigarrow b_1 \\ \hline
\leftarrow s_1 \rightsquigarrow t \\
\vdots \\
\leftarrow s_{m-1} \rightsquigarrow t \\
\quad a_m \rightsquigarrow b_m \\ \hline
\leftarrow s_m \rightsquigarrow t \\
\equiv \quad \text{(because } s_m = u) \\
\leftarrow s_m \rightsquigarrow t \\
\quad a_{m+1} \rightsquigarrow b_{m+1} \\ \hline
\leftarrow s_{m+1} \rightsquigarrow t \\
\vdots \\
\leftarrow s_{m+l-1} \rightsquigarrow t \\
\quad a_{m+l} \rightsquigarrow b_{m+l} \\ \hline
\leftarrow s_{m+l} \rightsquigarrow t \\ \hline
\square \ \text{(with } s_{m+l} = u_l = t)
\end{array}$$

The other axioms can also be proven in the same way. ■

6 Discussion

We presented a context-based approach for the resolution of semantic conflicts among heterogeneous information sources. The approach preserves the individual contexts of the repositories involved and achieves interoperability by context transformations on the basis of explicit models of the terminological context of an information source.

The approach presented is a practical one, therefore it does not cover all aspects of context transformation investigated in theoretical work (e.g. [16],[9]). However it extends the capabilities of working system (e.g. [2, 17]) by providing a calculus for rule-based and classification-based context transformation that has been implemented in the MECOTA System [22] and is guaranteed to be correct and complete.

The integration allows us to cover two types of semantic conflicts mentioned by Goh, namely scaling and naming conflicts. The integration of methods for resolving confounding conflicts involving reasoning about space and time are a major topic for further research.

References

1. AdV. Amtliches Topographisch Kartographisches Informationssystem ATKIS. Technical report, Landesvermessungsamt NRW, Bonn, 1998.
2. S. Bressan, C.H. Goh, K. Fynn, M. Jakobisiak, K. Hussein, T. Lee, S. Madnick, T. Pena, J. Qu, A. Shum, and M. Siegel. Demonstration of the context interchange mediator prototype. Technical report, Sloan School of Management, MIT, 1997.
3. EEA. Corine land cover. Technical guide. Technical report, European Environmental Agency. ETC/LC, European Topic Centre on Land Cover, 1997-1999.
4. P. Frankhauser, M. Kracker, and E. Neuhold. Semantic vs. structural resemblance of classes. *SIGMOD Record*, 20(4), December 1991.
5. Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. Integrating and accessing heterogeneous information sources in tsimmis. In *Proceedings of the AAAI Symposium on Information Gathering*, pages 61–64, Stanford, California, March 1995.
6. Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, and Jennifer Widom. The tsimmis approach to mediation: Data models and languages. In *Next Generation Information Technologies and Systems (NGITS-95)*, Naharia, Israel, November 1995. Extended Abstract.
7. Cheng Hian Goh. *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources.* Phd, MIT, 1997.
8. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 1993.
9. R.V. Guha. *Contexts: A formalization and some applications.* PhD thesis, Stanford University, 1991.
10. I. Horrocks. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. Springer-Verlag, Berlin, May 1998.
11. V. Kashyap and A. Sheth. Schematic and semantic semilarities between database objects: A context-based approach. *The International Journal on Very Large Data Bases*, 5(4):276–304, 1996.

12. Vipul Kashyap and Amit Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Applications*. 1996.
13. Vipul Kashyap and Amit Sheth. *Cooperative Information Systems: Current Trends and Directions*, chapter Semantic Heterogeneity in Global Information Systems: The role of Metadata, Context and Ontologies. 1997, Academic Press.
14. Zoubida Kedad and Elisabeth Métais. Dealing with semantic heterogeneity during data integration. In Jacky Akoka, Mokrane Bouzeghoub, Isabelle Comyn-Wattiau, and Elisabeth Métais, editors, *Conceptual Modeling - ER '99, 18th International Conference on Conceptual Modeling*, volume 1728 of *Lecture Notes in Computer Science*, pages 325–339. Springer, November, 15-18 1999.
15. W. Kim and J. Seo. Classifying schematic and data heterogeinity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991.
16. J. McCarthy. Notes on formalizing context. In *Proceedings of the thirteenth international joint conference on artificial intelligence IJCAI'93*, 1993.
17. E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies. In *Proceedings 1st IFCIS International Conference on Cooperative Information Systems (CoopIS '96)*. Brussels, 1996.
18. E. Sciore, M. Siegel, and A. Rosenthal. Context interchange using meta-attributes. *ACM Transactions on Database Systems*, 19(2):254–290, 1994.
19. Edward Sciore, Michael Siegel, and Arnie Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Transactions on Database Systems*, pages 254–290, June 1994.
20. Heiner Stuckenschmidt and Ubbo Visser. Semantic translation based on approximate re-classification. In *Workshop on Semantic Approximation, Granularity and Vagueness*, Breckenridge, Colorado, 2000.
21. Heiner Stuckenschmidt and Holger Wache. Context modelling and transformation for semantic interoperability. In Mokrane Bouzeghoub, Matthias Klusch, Werner Nutt, and Ulrike Sattler, editors, *Knowledge Representation Meets Databases, Proceedings of the 7th Intl. Workshop KRDB'2000*. 2000.
22. Holger Wache. A rule-based mediator for the integration of heterogeneous sources (extended version). TZI–Berichte, University of Bremen, Bremen, 1999.
23. Holger Wache. Towards rule–based context transformation in mediators. *Proceedings of the International Workshop on Engineering Federated Information Systems (EFIS 99)*, 1999.
24. Gio Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, March 1992.