Ontology-based Verification of Core Model Conformity in Conceptual Modeling

Claudia HESS and Christoph SCHLIEDER, Germany

Key words: Conformity verification, schema evolution, ontological modeling

SUMMARY

Reference models, often called core models are developed in various application domains. Until now, no computational support exists for the task of verifying the conformity between such core models and their domain models. The approach developed at Bamberg University uses Semantic Web technologies to examine whether or not a domain model is a derivation of a core model. This ontology-based conformity verification supports an iterative modeling process in which core or domain models are modified. Inference services as provided by ontologies can be used to analyze the relationships between core and domain models. For example, it is possible to formally prove which specific relations hold between two types of models and compare the result with the intentions of the domain experts involved in the modeling. As a consequence, knowledge not explicitly represented is revealed. In case that the domain model does not conform to the core model, an interpretation of the inference results is provided in ordinary language giving the domain experts hints on how to modify either the core model, the domain model or both. We evaluated our approach by applying it to the core cadastral model proposed by Lemmen et al. (2003) and a national cadastral model, the Greek model (Tzani, 2003) which both are results of research activities within the European COST Action G9 "Modelling Real Property Transactions". Although our approach to conformity verification was only evaluated with the cadastral models, it can be used for conformity verification in various applications domains due to its generality.

Ontology-based Verification of Core Model Conformity in Conceptual Modeling

Claudia HESS and Christoph SCHLIEDER, Germany

1. INTRODUCTION

Core conceptual models that act as a reference or standard for modeling activities play an important role in the development and the use of information systems. Such core models facilitate not only the reuse of existing software components during the realization of new systems, but also permit the translation from one conceptual realm into another. Generally, the purpose of a core model is not to provide a standard to which everybody is forced to adhere, but rather to represent general modeling knowledge that can be reused for specific domains. This is to say that domain models will use the core model as a basis, and extend it to their own requirements. National models that conform to a core model like the core cadastral model will not result in a unification of existing legal and administrative but in interoperable cadastral systems which still reflect the particular demands of the different countries. This way of standardization was already successful in other contexts such as Enterprise Resource Planning (ERP). ERP systems establish de facto standards which are flexible enough to be parameterized to the specific demands of each company. ERP systems consider not only the requirements imposed by legislation but also best practices of companies. Such standardization is also conceivable in the cadastral domain. Best practices and common technologies of cadastral systems are discussed in projects like the COST Action G9 “Modelling Real Property Transactions” and could be integrated in cadastral software which is customizable for each country. Core models play a vital role in this context as they reflect the basic ideas implemented in cadastral systems. They support the comparison of processes and structures of the individual national cadastral systems and those offered by the standardized system. Parts to be customized are therefore easily identified. If we can show that national cadastral models conform to a core cadastral model, then the development of cadastral software included its customizable parts is facilitated. Up until now, there is no formal approach for verifying conformity between domain models and the core model. In the following, a formal, ontology-based approach to the verification of core/domain model conformity is presented which is illustrated by applying it to the field of cadastral standardization.

The approach proposed can be applied to two basic cases of use. Firstly, it helps domain experts in modifying the domain model to be a derivation of the core model when the core model is intended as a normative standard. Secondly, the approach supports the inductive development of a core model on the basis of several already existing domain models. In the cadastral domain, we find a mixture of the base cases. The cadastral core model described by Lemmen et al. (2003) was developed on the basis of several national models already available in UML. However, domain models which are models of national cadastral systems are modeled as extension of the core cadastral model. Thus, the verification process should be able to guide domain experts in modifying core and domain models.

The next section compares our approach with current research activities applying Semantic Web technology and inference services to quality improvement in conceptual modeling. Section 3 explains the notion of "conformity" and the steps proposed to verify it. This process is illustrated in the subsequent section 4 with examples of the conformity verification between the core and the Greek cadastral model. The approach is evaluated in section 5. Finally, the conclusion summarizes results and highlights areas for future research.

2. RELATED WORK

The approach presented in this paper is based on ontologies, a technology promoted in the context of the Semantic Web research activities. Knowledge representation and reasoning capabilities provided by ontology modeling languages are used. In conformity verification, relations between core and domain model are formalized in an ontology language and inference services check consistency, compute the type of identified relations, and make implicitly defined knowledge explicit. Inference services also support the conceptual modeling of information systems in other approaches, similar as in conformity verification.

Franconi and Ng (2000) assist with their tool i•com the conceptual modeling of integration information systems such as data warehouses. The modeling of single and multiple schemas for databases with inter-schema constraints are supported. I•com therefore facilitates the integration of different data sources into a data warehouse. The conceptual models are extended Entity-Relationship (EER) models offering a wider range of modeling primitives than standard Entity-Relationship models. For example, is-a hierarchies and additional constraints, such as disjointness, can be expressed. As Literate UML models are used in the conformity verification, it is not the standard models, i.e. ER models or UML models that are used for conceptual modeling, but their extended version in which additional constraints can be encoded. Inconsistencies are not likely to occur without these supplementary modeling primitives. Inference services based on the representation of the conceptual models in an ontology modeling language, such as the Literate UML models in the conformity verification, or a Description Logic, such as the EER models, would not infer "interesting" facts. These results would not help in the i•com tool to improve the design phase of information systems, and in the case of conformity verification, to substantiate the decision on conformity. Although both approaches use inference services, the conceptual models are represented in a different way. The i•com tool transforms the EER models in the Description Logic *SHIQ*. The conformity verification does not use a particular Description Logic, but technology being developed for the Semantic Web, namely the ontology modeling language DAML+OIL (World Wide Web Consortium, 2001).

Berardi et al. (2003) use Description Logic for reasoning on UML class diagrams. The aim is to provide automated reasoning support to make implicit facts explicit and to detect inconsistencies in the models. The UML class diagrams without arbitrary OCL constraints are encoded in the Description Logic *ALCQI* which provides the capability to reason about UML class diagrams. Current Description Logic-based systems implement this Description Logic and may be used as core reasoning engines in the future implementation of sophisticated CASE tools (Berardi et al., 2003, Berardi, 2002).

These CASE tools would be a great help during the modeling of core and domain models. In this work, models serving as input for the conformity verification are not necessarily correct. Inconsistencies and implicit facts are detected during the conformity verification, but it would be sensible to use correct models for the verification process. Such CASE tools offering inference services would be a great help for the initial models, but conformity verification could not be provided because only reasoning about one model is permitted. There is no possibility of identifying corresponding elements in both models and continuing with reasoning.

3. CONFORMITY VERIFICATION

In the cadastral domain, most national administrations have – at least semi-formally – described a cadastral domain model, which reflects their legislation and special demands (Lemmen et al., 2003). Our computational approach supports the task of analyzing whether or not heterogeneous domain models are, in spite of all their differences, conform to a core model. In other words, we present a way to formally define and then examine with a software tool the conformity between national models and the core cadastral model.

Intuitively, we could say that models conform to a core model if they extend it to a particular domain without altering its essential properties. But how can we check our intuition about the conformity between two models? Formal criteria and a formal verification process are required. Figure 1 shows the complete verification process.

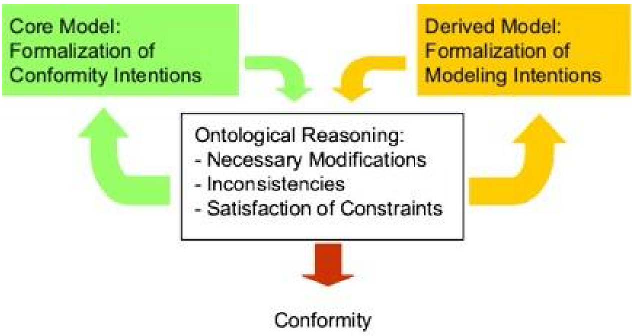


Figure 1- Iterative Process of Conformity Verification

The formalization of the problem is achieved by two parallel processes. On the one hand, the experts that author the core model specify their intentions about the kind of conformity domain models should satisfy in terms of constraints. These constraints describe which classes of the core model must have a corresponding class in the domain model and constitute a formalization of the *conformity intentions*. On the other hand, domain experts formulate their *modeling intentions*, by stating for classes of the domain model to which classes of the core model they should correspond.

The core model with its conformity intentions and the domain model with its modeling intentions serve as input for the ontology-based conformity verification. Both are formalized in an ontology modeling language into which core and domain models are transformed. Identified relations are integrated in one single ontological model consisting of core and domain model. This permits to compute the "similarity" of the classes with identified correspondences by ontological reasoning. A set of queries is sent to a reasoner. Thus, “conformity” means that the resulting relations meet the conformity constraints, or more concretely, that all conformity constraints are satisfied by having classes being as similar to each other as required by the constraints. The individual steps are described in the next chapters in more detail.

Core and domain models are adapted in an iterative process during which conformity is established. Ideally, the core model is already fixed and the interpreted output of the conformity verification is used as basis for changes in the domain model in the next iteration. Modifications in the core model or its conformity constraints are normally more tedious because their impact on other models, already declared as “conforming”, must be considered.

3.1 Transformation UML to DAML+OIL

The formal language for expressing both, conformity intentions and modeling intentions is the ontology modeling language DAML+OIL. Since the cadastral core model as well as many national cadastral models have been described using UML or literate UML, there is the need to transform from UML into DAML+OIL. Literate UML is founded on the idea of “Literate Modeling” proposed by Arlow, Emmerich, and Quinn (1999). It means that constraints or further relationships between elements are described in the natural language text in which the UML models are embedded. This technique is also used for the core cadastral model (Lemmen et al., 2003).
The transformation of the Literate UML models into ontology models therefore requires two steps. Firstly, the models themselves are translated into the ontology language. Such a transformation and its rules are proposed by Falkovych et al. (2003). Secondly, the information provided in the text surrounding the models has to be added to the class or attribute definitions in the ontology language.

Figure 2 illustrates the transformation of a part of the UML core cadastral model, serialized in XMI (Object Management Group, 2002) and the Literate UML belonging to it, into an ontology model in DAML+OIL. It becomes clear that ontological modeling provides an enhanced expressiveness compared with UML in the sense that not only the UML models but also the additional textual constraints can be expressed in the ontology model.

UML:

```
<UML:Class xmi.id = 'a15'
      name = 'Person'
      visibility = 'public'
      isSpecification = 'false'
      isRoot = 'false'
      isLeaf = 'false'
      isAbstract  = 'false'
      isActive =  'false'>
...
   <UML:Attribute xmi.id = 'a373'
      name = 'tmin'
      visibility = 'private'
      isSpecification = 'false'
      ownerScope = 'instance'>
   ...
   </UML:Attribute>
   ...
</UML:Class>
```

Literate UML:

```
"Each Person is either a
NaturalPerson or a
NonNaturalPerson. No Person can
be a NaturalPerson and a
NonNaturalPerson."
```

DAML+OIL:

```
<daml:Class rdf:about="#Person"
      rdfs:label="Person">
  ...
  <daml:Restriction>
    <daml:onProperty>
      <daml:DatatypeProperty
      rdf:about="#Person_tmin"/>
    </daml:onProperty>
    <daml:hasClass rdf:resource=
      "http://www.w3.org/2000/10/
      XMLSchema #date"/>
  </daml:Restriction>

  <daml:disjointUnionOf rdf:parse
      Type="daml:collection">
      <daml:Class rdf:about=
        "#NaturalPerson"/>
      <daml:Class rdf:about=
        "#NonNaturalPerson"/>
  </daml:disjointUnionOf>

</daml:Class>
```

Figure 2 - Transformation UML to DAML+OIL

3.2 Identification of Correspondences and Update of the Ontology Model

Correspondences between core and domain models are identified by the domain experts responsible for the domain model. As the domain model was designed as extension of the core model, relations between classes and attributes of both models can be identified by domain experts. An important question is as to whether or not the results of the conformity verification give hints for a manual identification of correspondences. Taking into account the effort invested in the modeling of core and domain models, it is justified to use a manual identification guaranteeing to preserve their high quality. A completely automated approach would fail to yield satisfactory results. State-of-the-art solutions to automated matching, like lexical analysis of class names, are not easily applicable to the cadastral models as names are not necessarily provided in the same language and even if they were, they are often quite different due to the historical development of national cadastral systems. Considering the high quality and the small size of the models, it would currently only be sensible to assist the user in identifying correspondences by a semi-automated process suggesting relations, but not to establish an automated matching.

In the following, we propose a workflow for the identification of correspondences and the update of the ontology models with these correspondences. During the stepwise refinement of correspondences in the workflow, it is often not possible to formulate relations directly, but heterogeneity problems must be considered. Heterogeneity problems occur because models reflect the specific requirements of their application domain, in the case of the cadastral models the different legislation and administration of the respective country. They can be divided in two groups (Wache, 2003). On the one hand, *structural heterogeneity* can be observed. That means that semantically equivalent elements are stored in different data structures, e.g. one model uses the attribute ***name***, another model has two attributes, namely ***first_name*** and ***last_name***. On the other hand, ***semantic heterogeneity*** can be found because of the different interpretation of information which is syntactically the same. For example, the attribute ***price: double*** may describe a price in euros or in dollars. This differentiation cannot be made based on the UML class diagrams. Domain experts should be aware of it and avoid it during modeling. In the course of the workflow description, structural heterogeneity problems are discussed in the steps in which they can occur and how to resolve them.

The workflow follows a set of generic mapping relations that supports the user in identifying correspondences and in dealing with heterogeneities. The set of generic mapping relations consists of relations between elements which can directly be added to the ontology model of core and domain model. A translation for each of the relations into an ontology modeling language is provided. The mapping relations are based on Wache's classification of data schema integration conflicts (Wache, 2003), but they are adapted to the needs of the verification process. Wache's classification aims at a translation of data from one application to another. In contrast to this classification, the conformity verification does not consider the instance-level but only the schema-level. The grouping is based on the different modeling primitives, i.e. classes and attributes and not on the different kinds of heterogeneity problems, as proposed by Wache. Domain experts who identify correspondences are more familiar with the elements they already use for modeling than with possible heterogeneity problems.
In the set of generic mapping relations, we distinguish mapping between:

(a) Classes
(b) Attributes
(c) Classes and attributes

Each of these relations can be bilateral, i.e. 1:1, or multilateral, i.e. 1:n, m:1 or m:n.

The workflow for identifying correspondence between the elements of two models is divided into several steps, each of which will be illustrated in the following by an example from the cadastral domain. It is designed for one pair of corresponding classes (or groups of classes) and must be repeated for every new pair.

1. Domain experts identify semantically equivalent parts in core and domain model:

Conformity between two classes could only be claimed if a class of the domain model contains the same information as a class of the core model, i.e. if they are semantically equivalent.
In our example, we start with the knowledge that a concept describing the owner of land can be found in every cadastral system (Lemmen et al., 2003). In the core model, the ***Person-classes*** describe the owner of land and in the Greek cadastral model, the ***BENEFICIARY-classes***.

2. Refinement of the relation on the class level:

The relation between a pair or group of classes, identified in the previous step, is considered by analyzing its cardinality.

(a) Bilateral relations between classes:
There are two directly corresponding classes in the core and domain model.
(b) Multilateral relations between classes:

One class corresponds to several classes due to a different distribution of the attributes among the set of classes. Before continuing with the next step, this structural heterogeneity problem is resolved. The set of classes is merged into one single class, i.e. the multilateral relation between classes is transferred to bilateral.

(c) Relation between attribute and class:

In some cases, an attribute corresponds to a class. This results from the reification of an attribute to a class. Such discrepancy at the meta-level can be reduced to a bilateral relation between classes and bilateral relations between the attributes of these classes.

Continuing with the example, we concentrate on the relation between the classes *Person* and *BENEFICIARY*, which correspond directly to each other, i.e. there is a bilateral relation between both.

3. Refinement of the relation on the attribute level:

In the third step, the relations between attributes are considered, i.e. semantically equivalent attributes are identified. Only bilaterally corresponding classes need to be considered as all other relations can be reduced to bilateral ones. Attention has to be paid to structural and semantic data heterogeneity between attributes.

(a) Bilateral correspondence between attributes:

Two attributes with the same, or convertible datatype, correspond to each other.

(b) Multilateral correspondence between attributes:

One attribute corresponds with several attributes of a class of the other model. By merging the set of attributes, if the datatypes permit it, bilateral correspondence between attributes can be established and the structural heterogeneity problem are resolved.

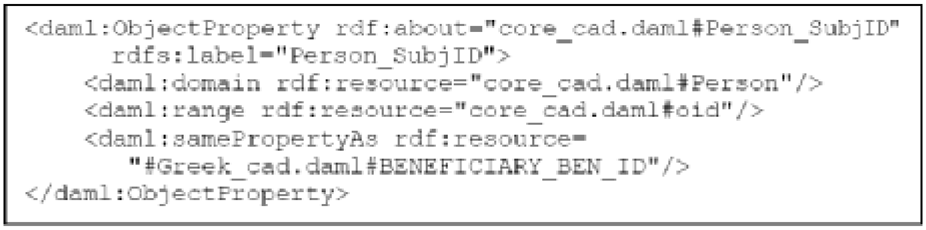
In the example, a correspondence can be established between the attribute *SubjID* of the class *Person* and the attribute *BEN_ID* of the class *BENEFICIARY*. The third step will be repeated as long as correspondences between the attributes of the selected classes are found.

The model consisting of the ontological representation of core and domain model is updated with the identified correspondences. Ontology modeling languages offer modeling primitives to express the equivalence between attributes and between classes. Table 1 lists these modeling primitives. The updated merged model serves as input for the computations described in the next chapter.

Relations	**DAML+OIL**	**OWL**
Bilateral relation between attributes	samePropertyAs	equivalentProperty
Bilateral relation between classes	sameClassAs	equivalentClass

Table 1 - Ontology Modeling Primitives for the Mapping Relations

In our example, the resulting part of the ontology model would look like in Figure 3.



```
<daml:ObjectProperty rdf:about="core_cad.daml#Person_SubjID"
      rdfs:label="Person_SubjID">
    <daml:domain rdf:resource="core_cad.daml#Person"/>
    <daml:range rdf:resource="core_cad.daml#oid"/>
    <daml:samePropertyAs rdf:resource=
       "#Greek_cad.daml#BENEFICIARY_BEN_ID"/>
</daml:ObjectProperty>
```

Figure 3 - Updated Ontology Model

3.3 Inference Services for the Conformity Verification

An identified correspondence between a core and a domain model class does not mean that these two classes are absolutely identical, – divergence is still possible. This difference, called *semantic domain heterogeneity*, arises from the different conceptualizations of objects in information systems (Wache, 2003). The results of the inference services on the models show to the user which classes of the input models are equivalent, which class of the domain model is a specialization of a class of the core model, or whether two classes correspond merely approximately.

The types of these exact and approximate correspondences are computed by a reasoner. Prerequisite for this computation is the identified relations on the attribute-level. In order to establish a correspondence, the user looks at the concrete definition of the attributes, i.e. at the intensional view of the concepts[1]. The reasoner however has an extensional view of the concepts in which a concept is defined as a set of individuals. This is a set-theoretical interpretation as used for defining the semantics of concepts in Description Logics. In other words, a concept denotes the set of all individuals that satisfy the properties specified in the concept definition (Baader et al., 2003).

Two concepts are determined by the reasoner as equivalent if both concepts have exactly the same extensions. Thus, according to the intensional view adopted by domain experts, all attributes of the core model class must have a corresponding attribute in the domain model class and inversely. The left part of Figure 4 shows two UML classes without any generalization classes. Correspondences are identified between the attributes *a1, b1* and *a2, b2*. The right part illustrates the extensional view. Concept *A* is the set of all individuals, which satisfy properties *a1* and *a2*. Concept *B* is, by analogy with *A*, the set of all individuals satisfying properties *b1* and *b2*. Concepts *A* and *B* are determined by a reasoner as equivalent.

Figure 4 - Equivalent Concepts

---

[1] In the context of object-oriented modeling, the terms “class” and “attribute” are used. In ontology modeling, the expressions “concept” and “property” are often used as synonyms. “Class” and “attribute” are favored in the context of UML diagrams, “concept” and “property” for ontologies.

Subsumption means that one concept is more general than a second. A subclass restricts possible extensions by adding further attributes to the class when compared with its superclass. Figure 5 shows on the left two UML classes, for which correspondence between the attributes *a1* and *b1* is identified. According to the extensional view demonstrated in the right part, concept *A* is the set of all individuals satisfying property *a1*. Concept *B* is the set of all individuals satisfying properties *b1* and *b2*. Due to the correspondence between *a1* and *b1*, all extensions of concept *B* are extensions of class *A*, but not inversely. Class *B* is therefore a specialization of class *A*.

Figure 5 - Subsuming Concepts

Overlapping is the weakest relation between exactly matching concepts. Transforming the first class into the second, there will always be a loss of information but required information is unavailable, too. Overlapping indicates only that there is some relation but that this relation is weak and will pose problems when mapping the models. Overlapping classes are pairs of classes where some, but not all of the extensions of the first class are also extensions of the second class. Inversely, the same applies. This is illustrated by Figure 6.

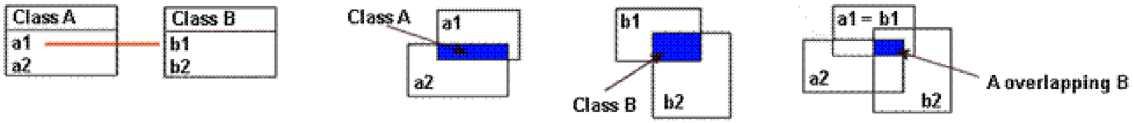


Figure 6 - Overlapping Concepts

Domain experts can identify a relation between concepts where the reasoner cannot determine a direct correspondence but nevertheless these two concepts are “similar” to each other. Figure 7 shows an example. Approximate mapping could be used, if two concepts do not overlap because of the disjointness of some attributes, such as *a2* and *b2* in the example.

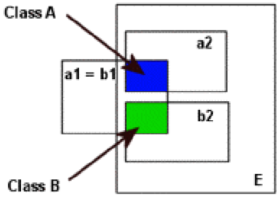


Figure 7 - Approximate Matching Concepts

A reasoner can prove the similarity of two classes in a formal way. The least upper bounds of a concept are determined, i.e. all minimal generalizations of a concept. They are computed by successively generalizing the datatypes of the properties. In the above example, the least upper bounds could be either computed for class *A* or *B*. The range of the properties *a2* or *b2* is generalized. Figure 8 shows the least upper bounds of class *B* which result of the generalization of the range of property *b2*. A reasoner could compare the original class *A* with the least upper bounds of class ***B*** by sending the standard queries. Having computed the least upper bounds only for one class, the resulting type is specialization.

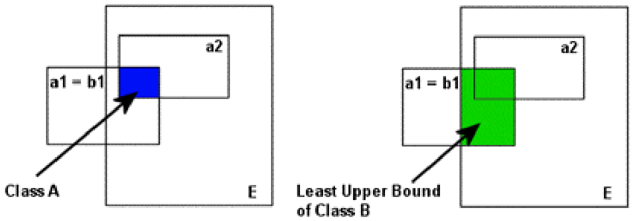


Figure 8 - Least Upper Bounds

Equivalence and subsumption can be directly computed by the reasoner. For example, a query checking the equivalence of two classes is formulated in the syntax of the reasoner RACER[2]: `(concept-equivalent? |file:/C:/CoreModel/CoreCad.daml#Person| |file:/C:/GreekModel/GreekCad.daml#BENEFICIARY|).`

The types overlapping and approximately matching can only be computed indirectly. Helper classes must be generated and used in the reasoner queries. Table 2 gives an overview of required actions and corresponding queries.

Type of Correspondence	Action	Query in RACER-Syntax
equivalence	-	concept-equivalent?
specialization	-	concept-subsumes?
overlapping	creation of the intersection class	concept-satisfiable?
approximate mapping	computation of the least upper bounds	concept-subsumes?

Table 2 - Actions and Reasoner Queries

Note that the type of a correspondence is not necessarily the intended type because classes are embedded in a hierarchical structure. Implicit knowledge is made explicit, i.e. knowledge encoded in the models which might be missed by human readers is determined by the reasoner. Even if in the first iterations, in which the models are perhaps incomplete, only relatively trivial relations can be inferred, information about the inferred knowledge becomes more and more important with the increasing complexity of the relations between the models. In a highly complex model, it is difficult to consider all the side-effects of a newly identified correspondence. Inconsistencies can occur in core or domain models but also across both models because of identified relations. They are detected by using inference mechanisms. Thus, complete knowledge of the effects of the formalized correspondences is available. These results are communicated to the user in the scope of an interpretation and reporting component. All results are edited in ordinary language, for example whether or not a conformity constraint is satisfied. Basic instructions are given in the case that conformity

---

[2] http://www.sts.tu-harburg.de/~r.f.moeller/racer/

constraints are violated, e.g. which relation has to be strengthened for the subsequent iteration.

4. TEST CASE

A prototype implementing all basic features of the theoretical approach was developed in order to evaluate the approach to conformity verification. It was tested with the core cadastral model and the Greek cadastral model. As both models were described as Literate UML models, they were translated from UML into the DAML+OIL ontology language. DAML+OIL was chosen because its successor OWL (World Wide Web Consortium, 2004) was not yet standardized when this work was started. Textual constraints in the Literate UML models were added to the ontology model of core and domain model. A first set of correspondence was integrated into the ontology model. Figure 9 shows a small part of the relations used for the first iteration. The reasoner RACER analyzed the relations between both models.

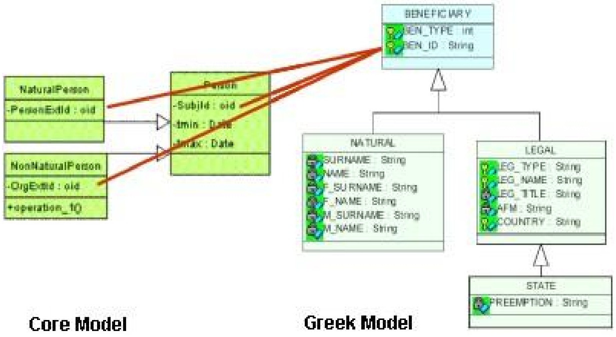


Figure 9 - Correspondences for Iteration 1

In the first iteration, the reasoner could only compute the overlapping type for most identified relations. Thus, conformity constraints of the specialization or equivalence type were not satisfied. On the basis of the results presented in the interpretation component, we strengthened the relations between core and Greek model in order to obtain relations of the specialization or even equivalence type. Exemplarily for all relations, the refinements of the relation between the person-classes are discussed. The following modifications show how such a refinement can be made. The decision whether or not these modifications should be realized is completely up to the Greek domain experts.

(a) If the attribute *Ben_Type* is added to the class *BENEFICIARY* in the Greek model only due to implementation issues, then this attribute could be removed.
(b) In the class *NaturalPerson* of the core model, the attribute *PersonExtID* specifies information related to the Person-Registry of a country. In contrast, the class *NATURAL* of the Greek model lists attributes which can be imported from the Person-Registry. Therefore, the attributes *Name*, *Surname*, *F_Name*, *F_Surname*, *M_Name*, *M_Surname* should be merged to an attribute "*AdditionalID*" corresponding to the attribute *PersonExtID*. The same applies for the class *LEGAL*.

(c) Additionally, we will remove for this second iteration the attributes *t_min* and *t_max*. We do not suggest this in general, but only for this example. It would be better to include a representation of temporal aspects in the Greek model.

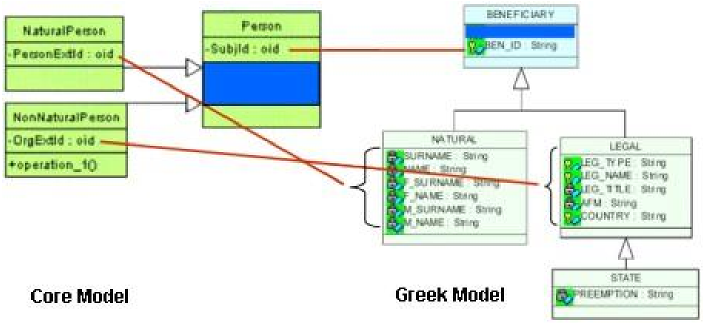


Figure 10 - Correspondences for Iteration 2

Figure 10 illustrates the proposed modifications. If these modifications are used for a second iteration, the following classes will be identified as equivalent by the reasoner: *Person* and *BENEFICIARY*, *NaturalPerson* and *NATURAL*, *NonNaturalPerson* and *LEGAL*.
The results of this second iteration in the conformity verification between core and Greek cadastral model must be reviewed by the Greek domain experts. They can decide whether this updated formalization reflects their modeling intentions in a better way than the correspondences of the first iteration resulting in relations of the overlapping type.

5. EVALUATION

Since the modeling work is still proceeding on both the core and the Greek cadastral model, we cannot expect the reasoner to come up with a result of the type "domain model conforms to the core model". However, an analysis of the reasoner's results can give indications on the modeling steps to take in the next iteration of the modeling process. For instance, a large number of overlapping concepts show that conformity constraints and intended correspondences need to be strengthened.
The experience obtained by the conformity verification between the cadastral models shows that the verification process can provide useful advice for future development of the models. The computations made by the reasoner seem to be a good basis for interpretation. In the current implementation, the interpretation component is rather simple as it gives only short explanations. It should be revised in future implementations of the prototype.
Applying the conformity verification to the core and Greek cadastral model, we noticed that it might be helpful to have more types of relations for the identification of corresponding elements. Apart from correspondence between classes, other types of relations such as "complement of" would be useful. Relationships between attributes could also be divided up. Modeling primitives are available in ontology modeling languages in order to declare a property as "subproperty" or as "inverse" of another.

In its current version, the prototype demonstrates an implementation of conformity verification with Semantic Web technologies, but it is not yet a product. A good understanding of ontologies is essential for the conceptualization and implementation of such ontology-based tool. For example, the exchange of ontology models between different Semantic Web tools is rather difficult because of differences in the serialization of ontology models. Provided that tools for preparatory parts such as the transformation from UML into an ontology language are available, it should be possible to implement a product for the conformity verification which is usable by domain experts without the help of knowledge engineers.

6. CONCLUSION

This work presented ontology-based conformity verification. Core and domain models were represented as ontology models and updated by the correspondences that domain experts had intended between the elements of both models. Domain experts obtained direct feedback because of the ability of the reasoner to formally prove the intended relationships. Reasoning permitted to detect inconsistencies in each model and across both models and revealed implicit facts. Thus, complete knowledge of the effects of the identified relations was provided. A consideration of this knowledge in subsequent versions of the models can increase their quality.

The results of our initial approach to conformity verification show that work in this area is incomplete. Future work should focus on the extension of the theoretical background. For example, this work has only touched on the inconsistencies in and across core and domain models. Examining the reasons for inconsistencies and providing solutions for resolving them would be an interesting research topic for conformity verification as well as for other ontology-based approaches such as information integration. Furthermore, an automated preliminary selection of corresponding elements could be realized so that domain experts would only need to confirm the identified relationships and this would save time during the verification process.

Verification of core model conformity can be useful in various application areas. It is claimed that the approach is not restricted to the cadastral domain although the approach was only evaluated with cadastral models. Great importance was attached to the generality of the approach and so subsequently no step has specialized on cadastral systems. Another example of use is that several business units in a company agree on a common data model which serves as a core model for the individual data models of each department and abstracts from their differences. Conformity verification could prove the relations between the department models and the core model.

Our approach reveals problems in the conformity verification with the core cadastral model as it actually is. The core cadastral model must be refined in close cooperation with experts for the national cadastral systems who in the other way round must be willing to modify their national model in order to achieve conformity. It is important to discuss core and national cadastral models on the same level of abstraction. There will always be problems in the conformity verification and the subsequent use of the models in various applications if some of the models are close to the implementation level representing directly the underlying databases and other models are more on the conceptual level abstracting from the concrete implementation.

But even if core and the national cadastral models are in an early stage, the core model with national models which conformity was shown by the conformity verification represent a promising approach to standardization in the cadastral domain. Our results permit to expect concrete applications on the basis of conforming models. The core model can be the basis of a core software application which is only adapted to the local requirements expressed in the domain models. Furthermore, data could be exchanged between organizations and institutions of different countries with the help of the core model representing the minimum common data of all domain models. The next step would be to realize software in of these application areas.

This work concentrates on conceptual models, but we plan to extend our approach to the verification of core model conformity to process models. There would be for example standardized process models for transactions of land property and conforming process models in the various national cadastral systems.

REFERENCES

Arlow, J., Emmerich, W., and Quinn, J. (1999). Literate Modelling – Capturing Business Knowledge with the 'UML'. In J. Bézivin and Pierre-AlainMuller, editors, *'The Unified Modeling Language'*, volume 1618, pages 189–199. UML'98 - Beyond the Notation. First International Workshop, Springer-Verlag, Mulhouse, France. Selected Papers.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press.

Berardi, D. (2002). Using DLs to reason on UML class diagrams. In G. Görz, V. Haarslev, C. Lutz, and R. Möller, editors, *Proceedings of the KI-2002 Workshop on Applications of Description Logics.* CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-63/, Aachen, Germany.

Berardi, D., Calì, A., Calvanese, D., and De Giacomo, G. (2003). *Reasoning on UML Class Diagrams.* Technical Report 11-03, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza".

Falkovych, K., Sabou, M., and Stuckenschmidt, H. (2003). UML for the Semantic Web: Transformational Approaches. In B. Omelayenko and M. Klein, editors, *Knowledge Transformation for the Semantic Web.* IOS Press.

Franconi, E. and Ng, G. (2000). The i.com Tool for Intelligent Conceptual Modeling. In *Proceedings of Knowledge Representation Meets Databases (KRDB'00),* pages 45–53. Berlin, Germany.

Lemmen, C., van der Molen, P., van Oosterom, P., Ploeger, H., Quak, Q., Stoter, J., and Zevenbergen, J. (2003). A modular standard for the cadastral domain. In *Proceedings of*

Digital Earth 2003: Information resources for global sustainability: knowledge, networks, technology, economy, society, natural and human resources, policy and strategy, 21-25 September 2003, pages 108-117. Brno, Czech Republic.

Object Manamgement Group (2002). XML Metadata Interchange (XMI) Specification, version 2.0.

Tzani, A. (2003). *Object-Oriented Modelling of the Greek Cadastre.* Master's thesis, School of Rural and Surveying Engineering of Aristotle University of Thessaloniki.

Wache, H. (2003). *Semantische Mediation für heterogene Informationsquellen.* Ph.D. thesis, Technologie-Zentrum Informatik, Universität Bremen, Akademische Verlagsgesellschaft Aka GmbH, Dissertationen zur Künstlichen Intelligenz, Berlin.

World Wide Web Consortium (2001). DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001.

World Wide Web Consortium (2004). OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004.

BIOGRAPHICAL NOTES

Claudia Hess received her diploma in Information Systems form Bamberg University, Germany. Currently, she is research assistant at the Laboratory for Semantic Information Technology, led by Prof. Dr. Christoph Schlieder at Bamberg University. In her Ph.D. studies, she is interested in the application of Semantic Web technologies to questions from the cultural and geographical sciences.

Christoph Schlieder is Professor and Chair of Computing in the Cultural Sciences since 2002 at Bamberg University, Germany. He holds a Ph.D. and a Habilitation degree in computer science, both from the University of Hamburg. Before coming to Bamberg he was professor of computer science at the University of Bremen where he headed the Artificial Intelligence Research Group. His primary research interests lie in developing and applying methods from semantic information processing to problems from the cultural sciences. Applications areas of his research work include GIS and mobile assistance technologies as well as digital archives.

CONTACTS

Dipl. Wirtsch-Inf Claudia Hess
Laboratory for Semantic Information Processing
Bamberg University
Feldkirchenstr. 21
96045 Bamberg
Germany
claudia.hess@wiai.uni-bamberg.de
Phone: + 49-951-863-2844
Fax: +49-951-863-5841
http://www.kinf.wiai.uni-bamberg.de/

Prof. Dr. Christoph Schlieder
Laboratory for Semantic Information Processing
Bamberg University
Feldkirchenstr. 21
96045 Bamberg
Germany
christoph.schlieder@wiai.uni-bamberg.de
Phone: + 49-951-863-2840
Fax: +49-951-863-5841
http://www.kinf.wiai.uni-bamberg.de/