Ontology reconciliation in terms of type refinement♣

© Leonid A. Kalinichenko, Nikolay A. Skvortsov

Institute for Problems of Informatics, Russian Academy of Sciences
{leonidk,scvora}@synth.ipi.ac.ru

Abstract

General principles of ontology integration and reconciliation independent on ontological representation models are described. Type refinement calculus over ontological concepts is used to detect correlation between them and reconcile cross-ontology relationships. The principles of ontology integration and reconciliation are applied for particular ontological model. Inference capabilities of description logics are used for ontology integration in frame of ontological models reduced to DAML+OIL model. Type refinement reasoning for more general ontological models requires full power of predicate logics.

1 Introduction

Interoperation of information resources requires detection of semantic relationships between them. Resource structure, behavior and extensions are defined by specification of classes and types. Semantics of specifications are provided by definition of a subject domain and relationships of resources and domain concepts. Nowadays domain ontologies are used for such purposes providing ontological concepts and their relationships. Various ontological models can be used for ontological modeling [6].

Ontological definitions of subject domains have become an essential part of information specifications. Semantic Web intends to use ontological definitions for interoperation of resources in the open information space. Subject mediators describe their semantic scope by ontological definitions of the subject domain. Resources registered in subject mediators define their ontological semantics in terms of the mediator for retrieving information semantically relevant to mediated queries. Data integration systems detect semantic correlation between specifications of resources being integrated. In component based design, semantically explicit specifications are required to correctly reuse pre-existing components in composed system.

Semantics of various information resources may be defined with different ontological definitions. Resources with similar semantics may use different ontologies. To make such resources semantically interoperable in the situations listed above correlation of their ontologies is to be determined.

Therefore ontology integration and reconciliation is a key issue to make independent information resources interoperable. Often expert decisions and discussions on common ontological commitments are required. Such discussions consist in detection of commonalities of ontological definition semantics and discovering of difference between definitions. Sometimes the difference of concepts is not so evident and may be discovered comparing classes of objects or finding out that the same object instances semantically belong to both compared ontological concepts. Technical support for ontology integration applies different approaches. Empirical approaches use linguistic methods over description of ontological concepts or evaluate structural commonalities for detection of correlation between ontological concepts and their relationships. Formal integration methods use inference in logical theories.

This paper considers an approach for ontology integration investigating compositional methods for information system development and information mediation. Methods for ontological mapping and evaluation of ontological relevance between resource specifications based on the verbal ontological model were considered previously. Here formal methods for more accurate integration of ontologies and their reconciliation are considered. In contrast to other works concerning ontology integration (for example informal advising system PROMPT [1] or formal basis for ontology integration in description logics [11]), the paper presents the method for formal integration process applicable for arbitrary ontological models.

Section 2 briefly describes ontological model construction principles applying the unified canonical model of information resource specification and shows cross-ontology mapping process. Section 3 explains ontology reconciliation process independently of ontological models of resources. This process is defined in terms of type refinement used in formal specification development. Section 4 describes an application of the method for a limited ontological model like DAML+OIL [13].

2 Ontologies in the canonical model

The ontological model used in the SYNTHESIS group projects was described in details in [7]. The common canonical model is used for ontological as well as for structural and behavioral resource specifications. The canonical model is intended for homogeneous

Proceedings of the 6th Russian Conference on Digital Libraries RCDL2004, Pushchino, Russia, 2004

representation of specifications of heterogeneous information resources applying specific model mapping technique [3].

2.1 Applying the canonical model for ontologies

Specifications of information resources are associated with ontological contexts containing concepts of the respective subject domain. Ontological concepts are described with their verbal definitions similar to definitions of words in an explanatory dictionary. Verbal definitions are required for human readability and also used for establishing semantic relationships between concepts. Four kinds of semantic relationships (hypernym/hyponym, positive, whole/part and related relationships which are usually applied in thesauri) can be discovered between ontological concepts as fuzzy ones.

Elements of ontological specifications mentioned above are special features for linguistic definition of concepts. But more expressive and formal definition of ontological concepts is achieved by specifying them in frame of the canonical model as abstract data types. Thus a concept definition may be constructed using attributes, associations, and invariants.

Depending on ontological models of resources and practical objectives of tasks solved with ontologies, ontological specifications may be defined in different styles. Features used for ontological concept specifications define mediator ontology language (MOL) that has a simple core and an extension as a subset of canonical model formed during mapping of certain ontological model of resources into the canonical model. The extensible framework principles for ontology language definition were described in [8]. For instance it was shown how the DAML+OIL model can be reversibly mapped into the canonical model and how the mediator ontology language equivalent to this model can be generated.

An example of ontological definitions in the canonical model subset equivalent to the DAML+OIL model follows. Ontological concept `Person` is defined exploiting parent relationships regarding parenting of a human being. Two subconcepts of `Person` are disjoint concepts `Male` and `Female`. The definitions that follow these subconcepts are association metatypes describing relationships of `Person`. These definitions will also be used below as a common ontology into which a local ontology will be integrated.

```
{ Person; in: type;
  hasParent: Person;
    metaslot in: HasParent end
  hasFather: Male;
    metaslot in: HasFather end
  hasMother: Female;
    metaslot in: HasMother end
  hasMom: Female;
    metaslot in: HasMother end
  hasChild: Person;
    metaslot in: HasChild end
  sameAs:
  { in: invariant, samePropertyAs,
    {{all a/Person(hasMother(a)=hasMom(a))}}
  }
};

{ Male; in: type;
  supertype: Person
  disjoint:
  { in: invariant, disjointWith,
    {{Male(a/Person)&Female(a/Person)={}}}
  }
};

{ Female; in: type;
  supertype: Person;
  disjoint:
  { in: invariant, disjointWith,
    {{Male(a/Person)&Female(a/Person)={}}}
  }
};

{ HasChild; in: metatype, association;
  instance_section:
  { domain: Person;
    range: Person
  }
};

{ HasParent; in: metatype, association;
  inverse: HasChild;
  instance_section:
  { domain: Person;
    range: Person
  }
};

{ HasFather; in: metatype, association;
  superclass: HasParent;
  instance_section:
  { domain: Person;
    range: Male
  }
};

{ HasMother; in: metatype, association;
  superclass: HasParent;
  instance_section:
  { domain: Person;
    range: Female
  }
};
```

2.2 Ontology integration approaches

Two approaches for interrelation of ontologies are applied:

- weak ontology integration and
- tight ontology integration.

Weak integration uses names of ontological concepts, their verbal definitions and semantic relationships and ignores their definitions as abstract data types. Verbal (linguistic) techniques of weak integration are described in [7]. Such ontology integration process results in relationships between ontological concepts of integrated contexts. Relationships are established by calculating the correlation coefficients between concepts on the basis of name and verbal definitions. Fuzzy relationships are deduced assuming their transitivity. Weak integration provides for discovery of interconcept correlations based on their verbal definitions. At the same time such integration may not be sufficient for more precise form of ontological integration exploiting formally proved relations between ontological concepts. The result of weak integration of ontologies may become a prelude for tight integration, a formal approach to integration of ontologies.

Tight integration uses specifications of ontological concepts as abstract data types. Semantic relationships deduced from the weak integration of the concepts provide an intuition to look for their more sound interpretation. For instance, a positive verbal concept relationship assumes that equivalence of respective concept type specifications is expected. A hyponym/hypernym relationship assumes that subconcept relation of respective concepts is expected. Such assumptions are to be proved formally with inference tools available for given ontological model. Verifying these relations of ontological concepts helps to reconcile them. For instance, if concepts are related but their equivalence or subconcept relationship is not confirmed, then such relationship probably does not hold.

3 Tight integration of ontologies in terms of type refinement

Specifications of ontological concepts are defined as abstract data types. Relations between concepts are specified as structural constituents of types. Ontological classes correspond to concept types, their extents constitute all objects that are semantically compliant with a respective concept. Mediator ontology model is extensible during mapping of ontological models of information resources into the canonical one and is synthesized during information resource registering. Common principles of ontology integration are applied for further development of an integration algorithm for specific mediator ontology language. To express ontological concept correlation, type refinement notion is used. It allows to reason on ontology integration in terms of abstract data types.

3.1 Type refinement

The term refinement [9] is used applied in formal specification development. An abstract specification of an information resource may be refined step by step to its implementation. Specifications may have different implementations such that each of them is a probable refinement of specifications. Formally any specification of an information resource function *p* may be represented in terms of preconditions *pre(p)* (admissible initial state) and postconditions *post(p)* (final state). *p1* is refined by a *p2* if exactly for all postconditions the weakest precondition of *p1* implies the weakest precondition of *p2*. Laws of refinement define this relation for constituents of type specifications [10]. Two abstract data types may be in a refinement order.

A signature Σ_T of a type specification $T = <V_T, O_T, I_T>$ includes a set of operation symbols O_T indicating operations argument and result types and a set of predicate symbols I_T (for the type invariants) indicating predicate argument types. Conjunction of all invariants in I_T constitutes the type invariant. V_T is an extension of type *T* (a carrier of the type).

Type *U* is a refinement of type *T* ($U \sqsubseteq T$) iff

- there exists a one-to-one correspondence between operation symbols of types:

  $Ops: O_T \leftrightarrow O_U$

- there exists an abstraction function that maps each admissible state of *U* into the respective state of *T*:

  $Abs: V_U \rightarrow V_T$

- Type invariant of type *U* is stronger than type invariant of type *T*:

  $\forall x \in V_U \quad I_U(x) \rightarrow I_T(Abs(x))$

- Each operation of type *U* is a refinement of operation of type *T*:

  $\forall o \in O_T, Ops(o)=o' \in O_U \quad o' \sqsubseteq o$

To establish an operation refinement it is required that for any state the precondition of refined operation *o* should imply the precondition of refining operation *o'* and operation postcondition of *o'* should imply postcondition of *o*:

$$(pre(o) \rightarrow pre(o'))\&(post(o') \rightarrow post(o))$$

Note that type attributes are represented as a couple of operation symbols getting and setting an attribute value. Subtyping is defined similarly to the refinement, but *Ops* becomes an injective mapping. Based on the notion of type refinement, a measure of common information between types in the type lattice can be established.

Since specification of an ontological concept is an abstract data type we can use the notion of type refinement for formal definition of concept correlation. For well-defined ontological concepts, if an ontological concept *c1* refines a concept *c2* then ontological class of *c1* is a subclass of ontological class of *c2*, in other words any object semantically corresponding to *c1* corresponds to *c2* too. This criterion does not depend on an ontological model and may be generally applied for ontology integration in the extensible framework of ontological models.

3.2 Ontology integration process

Development of an algorithm for ontology integration for particular mediator ontology language exploits common principles described below.

An ontology integration task differs in different architectures or environments. E.g., in mediators a number of local ontologies are to be integrated into federated ontology, for interoperation of information resources several local ontologies are to be interrelated.

There are many cases where more than two ontologies are to be integrated. Such complex task of ontology integration may be subdivided into subtasks of a local one integrated into the common one. This is the first common principle of ontology integration. We assume that the common ontology is represented in the canonical model corresponding to the chosen mediator ontology language (MOL).

To avoid technical heterogeneity of different ontologies, the second principle is applied stating that all manipulations with ontologies are performed in the canonical model. For this purpose an ontological model, in which a local ontology is represented, is mapped into the model of common ontology. The method for such mapping is described in [4] and [8]. After local ontological models having been mapped into the canonical one, the ontologies being integrated are represented uniformly in the canonical model.

The third common principle of ontology integration is that ontology integration starts with verbal methods of ontological concept correlation and continues with a formal methodology of verification and reconciliation of integrated ontologies. Weak integration [7] of the ontologies is performed for establishing preliminary assumptions regarding cross-ontology correspondence of concepts and concept relationships. These assumptions form the set of relationships requiring to be verified formally during tight integration. These relationships are used to form structural correspondences *Ops*. Also they help to assume correspondences between internal structural elements of ontological concepts during their integration. The result of tight ontology integration process is a set of relationships between concepts in different ontologies formally proved and confirmed by an expert.

The fourth principle of ontology integration is imposed by the use of canonical model subset for ontological modeling. Formal methods of ontology integration and reconciliation work with abstract data types. The criterion of correct correlation concepts is based on type refinement verification. The process of tight integration of two ontologies starts with conversion of the common ontology and the local ontology into a formal model in which verification of refinement for used ontological model becomes possible. It looks feasible to convert into a formal model all cross-ontology relationships found during weak integration too and apply refinement verification methods. But in some cases a better way might be a step-by-step process of adding relationships for iterative reconciliation of integrated ontologies. This approach requires a technique for verification of refinement of parts of concept specifications independent of unverified relationships. For this purpose abstract data type calculus may be applied [5]. A sound algorithm for such approach is an issue for further investigation.

Usually common ontology concepts are expected to be more general than local ones. If refinement hasn't been confirmed, ontological concepts must be corrected, the cross-ontology relationship must be removed, or kind of the cross-ontology relationship corrected. After that the procedure of refinement verification is repeated.

Two separate strategies for common and local ontologies manipulation are distinguished. A way of manipulation is chosen depending on completeness of the common ontology for its scope. During consolidation strategy a subject domain is defined and common ontology is formed. Experts choose local ontologies that are representative for the subject domain of common ontology. The common ontology may be completed with definitions from these representative ontologies. This is not trivial because any change in common ontology requires verification and correcting of all relationships with changed concepts. During the operational strategy the common ontology is considered to be complete enough and local ontologies should be mapped to the domain of common ontology so that a part of local ontology may be chosen which complies with the common ontology definitions. When possible, a local ontology is changed to comply with the common ontology.

Depending on ontological models of representative sources and on the style of their usage, the mediator ontology language formed on consolidation phase may acquire different complexity w.r.t. the task of refinement verification. Generally this task may become intractable. In such cases more comprehensive ontological models and semi-automatic tools for interactive proof of refinement should be used [12]. This makes important a choice of representative local ontologies and formation of the mediator ontology language to be controllable.

4 Implementation for a particular model

In [8] it was shown how the DAML+OIL ontological model [13] can be reversibly mapped into the subset of the mediator canonical model used for the mediator ontology language. Let representative local ontologies had a model equivalent or reducible to DAML+OIL. For such models as DAML+OIL it is possible to verify refinement of concepts automatically. This model is based on the description logics [2]. The logics was identified as expressive for concept and role subsumption inference. In DAML+OIL model the proof of subsumption is equivalent to the task of concept refinement verification. For simplicity it possible we try to apply ontology integration strategy applicable to the mediator ontology models reducible to DAML+OIL. In particular, the language OWL [15] that is close to the DAML+OIL model and OWL ontologies mapped to the canonical model fall into this class of ontological models.

Assume that the example of ontology above in the paper is a common ontology, and there is a need to integrate a local ontology into the common ontology. The following specifications in the canonical model define the local ontology. `Human` concept defines a human being, having available relationships `ParentOf`, `ChildOf`, `ToMother`, `ToFather`. These relationships have restricted cardinality. Two subconcepts of this concept are Mother and Father. They are disjoint and defined as having at least one child.

```
{ Human; in: type;
  parentOf: Human;
    metaslot in: ParentOf end
  childOf: Human;
    metaslot in: ChildOf end
  toMother: Mother;
    metaslot in: ToMother end
  toFather: Father;
    metaslot in: ToFather end
};

{ Father; in: type;
  supertype: Human;
  minCard:
  { in: invariant, onProperty, minCardinality,
    {{all x(Father(x)&count(ParentOf(x))>0)}}
  }
};

{ Mother; in: type;
  supertype: Human;
  disjoint:
  { in: invariant, disjointWith,
    {{Father(a/Human)&Mother(a/Human)={}}}
  }
  minCard:
  { in: invariant, onProperty, minCardinality,
    {{all x(Mother(x)&count(ParentOf(x))>0)}}
  }
};

{ ParentOf; in: metatype, association;
  instance_section:
  { domain: Human;
    range: Human
  }
};

{ ChildOf; in: metatype, association;
  inverse: ParentOf;
  instance_section:
  { domain: Human;
    range: Human;
    association_type: {{0,2},{0,inf}}
      metaslot
        in: onProperty, maxCardinality
      end
  }
};

{ ToMother; in: metatype, association;
  superclass: ChildOf;
  instance_section:
  { domain: Human;
    range: Mother;
    association_type: {{0,1},{0,inf}}
      metaslot
        in: onProperty, maxCardinality
      end
  }
};

{ ToFather; in: metatype, association;
  superclass: ChildOf;
  instance_section:
  { domain: Human;
    range: Father;
    association_type: {{0,1},{0,inf}}
      metaslot
        in: onProperty, maxCardinality
      end
  }
};
```

To integrate the example into the common ontology it is required to apply weak ontology integration methods to reveal preliminary cross-ontology relationships. Using concept and relationship names with thesaurus-like linguistic relations, or applying analysis of verbal descriptions of concepts (which are not given here in specifications) we assume the following relationships:

- concepts Human and Person are equivalent;
- the concept Father is a subconcept of Male;
- the concept Mother is a subconcept of Female;
- Relationships ParentOf and HasChild are equivalent;
- Relationships ChildOf and HasParent are equivalent;
- Relationships ToFather and HasFather are equivalent;
- Relationships ToMother and HasMother are equivalent.

Revealed cross-ontology relationships are applied to ontological specifications and refinement relationships are verified for related concepts and concept relationships. Attribute definitions of the concept Person use metatype associations HasChild, HasParent, HasFather and HasMother. Attribute definitions of the concept Human use ParentOf, ChildOf, ToFather and ToMother. So the relationship between Human and Person depends on relationships between these metatypes. The relationship of equivalence between Human and Person holds if relationships of equivalence between association metatypes are correct. Concepts Male and Female correctly refined by Father and Mother respectively because minimal cardinality restrictions in Father and Mother make type invariants of them stronger than type invariants of Male and Female concepts. Equivalence of ParentOf and HasChild association metatypes holds, they have no any restrictions and their domains and ranges are equivalent. ChildOf, ToFather and ToMother have maximal cardinality restrictions that are stronger than restrictions of refined associations HasParent, HasFather and HasMother respectively. Kind of relationships between these associations must be corrected from equivalence to subconcept. Finally, relationship between concepts Human and Person must be verified again. Since attributes of Human refine attributes of Person, then the local ontology concept Human refines common ontology concept Person. The kind of relationship between these types must be corrected. Now all cross-ontology relationships established during weak integration are verified formally.

For the common ontology chosen, inference of concept subsumption and equivalence may be performed in the tool FaCT [14] that is based on the description logic equivalent to DAML+OIL model and has features to automatically verify subsumption in it. For this purpose both ontologies must be loaded into the system. Cross-ontology relationships must be added too. In the DAML+OIL model, relationship of equivalence of classes (concepts) is declared as `sameClassAs`, subconcept/superconcept relationship is declared as `subClassOf`. Analogously equivalent properties (relationships) are defined by `samePropertyAs` relationship and subproperty/superproperty relationship is defined by `subPropertyOf`. Each of these cross-ontology relationships must be verified using inference capability of FaCT. Application of FaCT system for tight ontology integration extends weak ontology integration methods.

5 Conclusion

Tight ontology integration approach is defined in terms of the abstract data type refinement and its common principles of integration are provided. This approach is applicable for compositional information systems development, heterogeneous information sources mediation as well as for any activity related to the ontology manipulation.

References

[1] D. Calvanese, G. Giacomo, M. Lenzerini. A Framework for Ontology Integration. In Proc. of the First Semantic Web Working Symposium, pp. 303-316, 2001

[2] I. Horrocks, U. Sattler, S. Tobies. Practical reasoning for very expressive description logics. Logic Journal of IGPL, 8(3), 2000

[3] L. A. Kalinichenko. Integration of Heterogeneous Semistructured Data Models in the Canonical One. In Proceedings of the First All-Russian Conference on Digital Libraries, St. Petersburg, 1999

[4] L. A. Kalinichenko. Methods and tools for equivalent data model mapping construction. In Proceedings of the International Conference on Extending Database Technology (EDBT'90), Venice, 1990

[5] L. A. Kalinichenko. Compositional Specification Calculus for Information Systems Development. In Proceedings of the East-European Symposium on Advances in Databases and Information Systems (ADBIS'99), Springer-Verlag, LNCS, 1999.

[6] L. A. Kalinichenko, M. Missikoff, F. Schiappelli, N. A. Skvortsov. Ontological Modeling Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003, St.-Petersburg, Russia, 2003

[7] L. A. Kalinichenko, N. A. Skvortsov. An Approach to Ontological Modeling and Establishing Intercontext Correlation in the Semistructured Environment. 2nd Russian Conference "Digital Libraries", Protvino. Sep 26-28, 2000

[8] L. A. Kalinichenko, N. A. Skvortsov. Extensible Ontological Modeling Framework for Subject Mediation Proceedings of RCDL'2002 Conference, Dubna, 2002

[9] C. Morgan, K. Robinson. Specification Statements and Refinement, *IBM Journal of Research and Development*, 31(5):546-555, Sep 1987

[10] C. Morgan. Programming from Specifications (2nd Edition). Prentice-Hall, 1994.

[11] N. Noy, M. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, 2000

[12] B-Toolkit User's Manual - Release 3.4. Copyright B-Core (UK) Ltd., 1997

[13] DAML+OIL (March 2001) Reference Description. http://www.w3.org/TR/daml+oil-reference

[14] FaCT system (open source). http://www.cs.man.ac.uk/FaCT/

[15] OWL Web Ontology Language Reference (February 2004). http://www.w3.org/TR/owl-ref/

♣ Supported by the grant RFBR N 04 07 90083.