Ontology Correspondence via Theory Interpretation

Immanuel Normann[1] and **Oliver Kutz**[2]

Abstract. We report on ongoing work to apply techniques of automated theory morphism search in first-order logic to ontology matching and alignment problems. Such techniques are able to discover 'structural similarities' across different ontologies by providing theory interpretations of one ontology into another.

We sketch the techniques currently available for automating the task of finding theory interpretations in first-order logic and discuss possible extensions and modifications for other ontology languages such as description logics and modular ontology languages such as $\mathcal{E}$-connections.

1 Introduction and Motivation

The problem of finding semantically well-founded correspondences between ontologies, possibly formulated in different logical languages, is a pressing and challenging problem. Ontologies may be about the same domain of interest, but may use different terms; one ontology might go into greater detail than another, or they might be formulated in different logics, whilst mostly formalising the same conceptualisation of a domain, etc. To allow re-use of existing ontologies and to find overlapping 'content', we need means of identifying these 'overlapping parts'.

Often, ontologies are mediated on an ad-hoc basis. Clearly, any approach relying exclusively on lexical heuristics or manual alignment is too error prone and unreliable, or does not scale. As noted for instance by [16], even if a first matching is realised automatically using heuristics, a manual revision of such candidate alignments is still rather difficult as the semantics of the ontologies generally interacts with the semantics given to alignment mappings.

A lot of research has already been carried out in the area of ontology matching [6]. However, most work is based on approximate matching of the graph structures of taxonomies and statistical or heuristic approaches, see e.g. [10, 9].

A new approach, that we currently explore, is to apply methods of automated theory interpretation search to the realm of ontologies. Such methods have been mainly developed for the application to formalised mathematics (and some of the techniques currently are specialised for theories formulated in first-order logic). Whilst theory interpretations are rather flexible in that they are not restricted to exact formulation and phrasing of ontology terms, in contrast to the above mentioned approaches to ontology matching, they do establish a logically rather strict relationship across two ontologies, namely that all axioms of one ontology are provable in the other along a translation, essentially embedding one ontology into another.

Such embeddings can give guidance in ontology development, and can be applied for searching and structuring of 'design patterns' for ontologies.

[1] Department of Linguistics and Literature, University of Bremen, Germany, email: normann@uni-bremen.de

[2] SFB/TR 8 Spatial Cognition, University of Bremen, Germany, email: okutz@informatik.uni-bremen.de

2 Theory Interpretations and Refinements

Theory interpretations have a long history in mathematics generally, and are probably employed by any 'working mathematician' on a daily basis; the basic idea is the following: given two theories T_1 and T_2 (which we here assume to be first-order theories), find a mapping of terms of T_1 to terms of T_2 (a signature morphism, typically expected to respect typing) such that all translations of axioms of T_1 become provable from T_2. If such a theory interpretation is successfully provided, all the knowledge that has already been collected w.r.t. T_1 can be re-used from the perspective of T_2, using the translation (see [7] for some examples from the history of mathematics). In this case, in mathematical jargon, we might say that T_2 **carries the structure** of T_1.

Certain, very basic structures, are found everywhere in mathematics. The most obvious example might be group theory. The basic abstract structure of a group can be re-interpreted in a more concrete setting, giving the group in question additional structure (think of the natural numbers, rings, vector-spaces, etc.). Re-using the metaphor mentioned above, we say that an ontology O_2 *carries the structure of* O_1, if the latter can be re-interpreted, by an appropriate translation σ, into the language of O_2 such that all of its axioms are entailed by O_2. In this case, informally, we consider the pair $\langle O_2, \sigma \rangle$ a **context** for O_1.

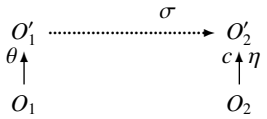

Figure 1. A heterogeneous refinement/theory interpretation.

The notion of theory interpretation is also closely related to the notion of refinement from software engineering. A heterogeneous refinement is depicted in Fig. 1. Here, given ontologies O_1 and O_2, possibly formulated in different logics, say a DL and a variant of first-order logic, we want to show that O_2 specialises, or refines, the information contained in O_1. To do this, we first need to translate

both O_1 and O_2 into a common logic, say first-order logic, by means of suitable translations θ and η. Here, the translation η additionally needs to be conservative in order not to 'distort' the information contained in O_2. In a final step, a theory interpretation σ from O_1' to O_2' is provided, showing that all translations of axioms of O_1 hold in O_2' along $\theta \circ \sigma$. The notion of a heterogeneous refinement also leads to a general definition of heterogeneous sub-ontology, compare [13].

It should be clear that whenever either the logics or the signatures of the ontologies involved do not directly fit, there are a number of possible solutions to choose from (we can just extend the logic in question, we can extend definitionally the signature, or both).[3]

Here is an illustrative example from mathematics:

Example 1 (Lattices and Partial Orders) *Consider P as the theory of partial-orders with $\mathbf{Sig}(P) = \{\leq\}$ and let L be the theory of lattices with $\mathbf{Sig}(L) = \{\sqcap, \sqcup\}$. These are both first-order theories, so the logical languages are directly compatible and we only need to translate the non-logical terms. However, the signatures obviously do not fit as L has only binary functions (rather than relations). This can be remedied by extending the signature of L by a binary relation symbol $\sqsubseteq$ (which makes the signatures fit by the mapping $\sigma := \leq \mapsto \sqsubseteq$), and define $\mathfrak{S} = \{\forall a, b. a \sqsubseteq b \leftrightarrow a \sqcup b = a\}$. This is a definitional axiom. It can now be seen that $L \cup \mathfrak{S} \models \sigma(P)$, i.e. σ is a theory interpretation embedding the theory of partial orders into the theory of lattices, using the definitional axiom in $\mathfrak{S}$.*

Thus, we may say that lattices carry the structure of partial orders. It should be obvious that both these theories also define central structures for ontology design.

3 Automated Discovery of Theory Interpretations

The goal of discovering ontology interpretations may be rephrased as the problem of finding all those ontologies in a large repository $\mathfrak{R}$ that could serve as a context (in the above sense) for a given ontology O_1. I.e. given O_1, we are looking for the set

$$\{O_2 \in \mathfrak{R} \quad | \quad O_1 \text{ is interpretable into } O_2\}.$$

Conversely, given O_2, we can look for the set

$$\{O_1 \in \mathfrak{R} \quad | \quad O_2 \text{ is interpretable into } O_1\},$$

i.e. the set of all ontologies into which O_2 can be interpreted.

In case of ontologies formalised in **FOL**, this task is undecidable, whereas for ontologies formalised in DL it is generally decidable. I.e., given the ontologies O_1, O_2, and a signature morphism σ from O_1 to O_2, it is decidable whether the σ-translated axioms of O_1 are entailed by O_2. However, the combinatorial explosion yielded by trying to find all possible symbol mappings between two given ontologies makes such a brute force approach unpractical.

To obtain one of the answer sets above in reasonable time (i.e seconds or minutes), we necessarily have to relax our initial goal towards an approximation of the set of all possible contexts for a given ontology. In summary, our approach for the first-order caseis based on formula matching modulo an equational theory—elaborated in detail in [20]. We want to outline this in the following.

Suppose we are given a source ontology O_1 and a target ontology O_2, which we assume have been translated to first-order via the standard translations. In the first step, we normalise each sentence of these ontologies according to a fixed equational theory. The underlying technique basically stems from term-rewriting: rewrite rules represent an equational theory such that all sentence transformations obtained through these rules are in fact equivalence transformations, e.g. such as $\neg A \sqcap \neg B \mapsto \neg(A \sqcup B)$. A normal form of a convergent rewrite system is then the unique representative of a whole equivalence class of sentences. The goal of normalisation is thus to identify (equivalent) expressions such as $\neg(\exists R.A \sqcap B)$ and $\neg B \sqcup \forall R.\neg A$.

In the next step, we try to translate each normalised axiom φ from O_1 into O_2, i.e. we seek a sentence ψ in O_2 and a translation σ such that $\sigma(\varphi) = \psi$. Note that potentially each axiom can be translated to several target sentences via different signature morphisms. To translate all axioms of O_1 into O_2, there must be a combination of *compatible* signature morphisms[4] determined from the previous, single sentence matchings. This task is also known as (consistent) many-to-many formula matching. In fact many-to-many formula matching modulo some equational theory is already applied in automated theorem proving (ATP) [11]. However, our approach is different in a crucial aspect: it allows for significant search speed up. We are normalising all ontologies as soon as they are inserted into the repository, i.e. not at cost of query time. Only the normalisation of the query ontology is at query time. Moreover, the normal forms not just allow for matching modulo some equational theory, but also enable a very efficient matching pre-filter based on skeleton comparison. A sentence skeleton is an expression where all (non-logical) symbols are replaced by placeholders. E.g., $\square \sqsubseteq \square \sqcup \square$ is the skeleton of $A \sqsubseteq B \sqcup C$. Obviously, two sentences can only match if they have an identical skeleton. Since syntactic identity can be checked in constant time, a skeleton comparison is a very efficient pre-filter for sentence matching.

Concerning sentence normalisation, some further improvements in comparison to traditional normalisation in ATP should be mentioned. In ATP, formulae are typically normalised to CNF for resolution, or DNF for tableaux reasoning. Both are not unique normal forms (even not modulo associativity and commutativity (AC)). Our approach uses a Boolean ring normal form which is unique modulo AC. Moreover, we developed an AC standardisation that computes a unique skeleton for given AC-equivalence classes of sentences.

All the presented techniques were developed in the context of formalised mathematics and a tool for the automated discovery of theory interpretations in first-order logic has already been implemented [20]. This has been used for experiments on a **FOL** version of the Mizar library [18] that contains about 4.5 million formulae distributed in more than 45.000 theories, and thus is the world's largest corpus of formalised mathematics. Experiments where each theory was used as source theory for theory interpretation search in the rest of the library demonstrated the scalability of our approach. On average, a theory interpretation search takes about one second and yields 60 theory interpretations per source theory.

4 Discussion and Outlook

Because of the encouraging results in formalised mathematics, we are currently adopting and modifying these techniques for the application in the realm of ontologies. In principle, the methods for automated discovery of theory interpretations developed in [20] can

[3] E.g. the OneOf constructor found in many description logics allowing a finite enumeration of the elements of a concept is also expressible as a disjunction of nominals, and conversely. Such translations/simulations can be handled by a library of logic translations.

[4] Two signature morphisms are compatible if they translate all their common symbols equally.

be applied to any formalised content as long as the entailment relation obeys certain properties (as specified e.g. in entailment systems [17]).

Of course, there is no guarantee that what is successful for mathematical theories is equally successful for formal ontologies, and some of the characteristics and features regularly found in ontologies are problematic.

A central difference between formalised mathematics and ontologies is in the expressivity of the underlying formal languages: obviously, **FOL** (mostly used for formalised mathematics) is more expressive than typical DLs (used for ontologies). This is also reflected in the more complex grammar of **FOL**: DL typically completely lacks variables, often has no function symbols, and also no relations of arity greater than two. Hence, **FOL** formulae containing such constructs do not have a directly corresponding syntactical expression in DL. Intuitively, we may say that, compared to DL, there is a larger syntactic variety of **FOL** formulae. In practice, the majority of ontologies that can be found on the internet even make use of only a rather small fragment of the DL expressivity—for instance, ontologies which are just taxonomies have no other axioms than is-a hierarchies.

This difference in syntactic complexity between **FOL** and DL has most likely in many cases two (mutually dependent) unfavourable consequences for ontology morphism search: 1) a less effective skeleton filter and 2) lots of meaningless search results. Due to the lower structural variety of axioms in ontologies, many DL axioms share identical skeletons. Thus, on average, a given skeleton in DL does not reduce the search space for matching formulae in an ontology on the same scale as a skeleton in a **FOL** theory would. For the same reason, the chance to match a source formula to many target formulae is higher in DL ontologies than in **FOL** theories. In other words: it is generally likely that the number of interpretations between DL theories is much higher than between **FOL** theories. In many (if not most) cases, these DL interpretations may turn out to be meaningless, though. A typical example is an interpretation between taxonomies: if we consider the is-a hierarchy of a taxonomy as a tree, then ontology matching becomes essentially tree matching. Clearly, a 'small' tree can often be mapped into a 'large' tree in several ways. Since such a mapping does not at all depend on the node names of the involved trees (i.e. the terms of the ontologies), this means that there may be quite a few interpretations between taxonomies of completely unrelated domains. Such interpretations, however, are meaningless from a common sense perspective.

Initial experiments on DL ontologies already suggested some ideas on how to overcome these problems in future work:

- Interactive search space reduction: the user should be able to enforce some mappings of non-logical symbols—often some mappings are explicitly intended.
- Exploitation of the decidability of DLs for the morphism search.
- Specialised normal forms designed particularly for various DLs.

Many approaches to connecting, aligning, or linking ontologies, or to interpret the vocabulary (and thus re-use its axiomatisation) of one ontology in another, rely on notions of symbol mapping that are more complex than simple signature morphisms. Examples of such formalisms, which introduce additional semantic complexity, are distributed DLs [16, 3, 2] and $\mathcal{E}$-connections [15, 5]. The general semantic idea of these approaches is similar, and is illustrated in Fig. 2.

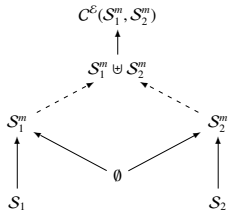

Figure 2. $\mathcal{E}$-connections or DDLs as structured heterogeneous theories

Here, given two ontologies $\mathcal{S}_1$ and $\mathcal{S}_2$, we first construct their disjoint union keeping the vocabulary completely disjoint. Given a 'link language' that allows to axiomatically connect the sorts of the component ontologies, we can in a second step provide a theory extension $C^{\mathcal{E}}(\mathcal{S}_1^m, \mathcal{S}_2^m)$, see [13] for technical detail. The nature of the 'link language' is here left open intentionally, as this is the main point of divergence between DDLs, $\mathcal{E}$-connections, and similar approaches.

The necessity of using such kinds of more expressive link or mapping languages has been shown in many application scenarios. [12], for instance, analyse the problem of relating an ontology encoding the linguistic spatial semantics of natural language utterances as represented in GUM [1] with spatial calculi, using the example of the double-cross calculus DCC [8] for projective relations (orientations).

Clearly, the problem of theory interpretation search takes a different turn in such a situation. Given ontologies $\mathcal{S}_1$ and $\mathcal{S}_2$, find an appropriate bridge theory $\mathcal{B}$ (for instance a set of bridge rules in the sense of [3]) and a signature morphism σ such that for all formulae ϕ (in an appropriate signature)

$$\mathcal{S}_1 \models \phi \text{ implies } \langle \mathcal{S}_1, \mathcal{S}_2, \mathcal{B} \rangle \models \sigma(\phi)$$

Whilst the bridge theory typically interacts with the semantics of $\mathcal{S}_1$ and $\mathcal{S}_2$, it is often natural to assume that $\mathcal{B}$ is conservative in at least one direction (see e.g. [4]). Different variants of this definition need to be analysed. Moreover, the algebraic equational theory that is used to identify equivalent formulae needs to be adapted in order to allow the identification of axioms loosely associated through a bridge theory.

Concerning automated reasoning support, a tool for the automated discovery of theory interpretations in first-order logic has already been implemented [20], and is currently being integrated into the Hets system [14, 19] with the aim of adding specialised routines for decidable ontology languages and corresponding integration problems. At the moment we perform experimental tests on a set of ontologies to evaluate the potential of first-order based theory interpretation search.

Acknowledgements

For the work reported in this article we gratefully acknowledge the financial support of the European Commission through the OASIS project (Open Architecture for Accessible Services Integration and Standardisation) and the Deutsche Forschungsgemeinschaft through the Collaborative Research Center on Spatial Cognition (SFB/TR 8). The authors would like to thank Joana Hois for fruitful discussions.

REFERENCES

[1] J. Bateman, T. Tenbrink, and S. Farrar, 'The Role of Conceptual and Linguistic Ontologies in Discourse', *Discourse Processes*, **44**(3), 175–213, (2007).

[2] A. Borgida, 'On Importing Knowledge from DL Ontologies: some Intuitions and Problems', in *Proc. of DL*, (2007).

[3] A. Borgida and L. Serafini, 'Distributed Description Logics: Assimilating Information from Peer Sources', *Journal of Data Semantics*, **1**, 153–184, (2003).

[4] B. Cuenca-Grau and O. Kutz, 'Modular Ontology Languages Revisited', in *Proc. of the IJCAI'07 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa), Hyderabad, India, January 2007*, (2007).

[5] B. Cuenca Grau, B. Parsia, and E. Sirin, 'Ontology Integration Using $\mathcal{E}$-Connections', in *Ontology Modularization*, eds., H. Stuckenschmidt and S. Spaccapietra, Springer, (2009). To Appear.

[6] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, Heidelberg, 2007.

[7] W. M. Farmer, 'Theory Interpretation in Simple Type Theory', in *Higher-Order Algebra, Logic, and Term Rewriting*, volume 816 of *LNCS*, pp. 96–123. Springer, (1994).

[8] C. Freksa, 'Using orientation information for qualitative spatial reasoning', in *Theories and methods of spatio-temporal reasoning in geographic space*, volume 639 of *LNCS*, 162–178, Springer, (1992).

[9] F. Giunchiglia, F. Mcneill, M. Yatskevich, J. Pane, P. Besana, and P. Shvaiko, 'Approximate structure-preserving semantic matching', in *Proceedings of ODBASE*, 1217–1234, (2008).

[10] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, 'Semantic Matching: Algorithms and implementation', *Journal on Data Semantics*, **IX**, 1–38, (2007).

[11] P. Graf, *Term Indexing*, volume 1053 of *Lecture Notes in Computer Science*, Springer, 1996.

[12] J. Hois and O. Kutz, 'Natural Language meets Spatial Calculi', in *Spatial Cognition VI. Learning, Reasoning, and Talking about Space. 6th International Conference on Spatial Cognition*, eds., C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, LNCS, pp. 266–282. Springer, (2008).

[13] O. Kutz, D. Lücke, and T. Mossakowski, 'Heterogeneously Structured Ontologies—Integration, Connection, and Refinement', in *Advances in Ontologies. Knowledge Representation Ontology Workshop (KROW 2008)*, eds., T. Meyer and M. A. Orgun, volume 90 of *CRPIT*, pp. 41–50, Sydney, Australia, (2008). ACS.

[14] O. Kutz, D. Lücke, T. Mossakowski, and I. Normann, 'The $\mathcal{OWL}$ in the CASL—Designing Ontologies Across Logics', in *OWL: Experiences and Directions, 5th International Workshop (OWLED-08)*, October 26–27, ISWC, Karlsruhe, Germany, (2008).

[15] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev, '$\mathcal{E}$-Connections of Abstract Description Systems', *Artificial Intelligence*, **156**(1), 1–73, (2004).

[16] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, 'Reasoning Support for Mapping Revision', *Journal of Logic and Computation*, (2008).

[17] J. Meseguer, 'General logics', in *Logic Colloquium 87*, pp. 275—329. North Holland, (1989).

[18] Mizar mathematical library. Web Page at `http://www.mizar.org/library`.

[19] T. Mossakowski, C. Maeder, and K. Lüttich, 'The Heterogeneous Tool Set', in *TACAS 2007*, eds., Orna Grumberg and Michael Huth, volume 4424 of *LNCS*, pp. 519–522. Springer, (2007).

[20] I. Normann, *Automated Theory Interpretation*, Ph.D. dissertation, Department of Computer Science, Jacobs University, Bremen, 2009.