A Framework for Distributed Reasoning on the Semantic Web Based on Open Answer Set Programming

Cristina Feier

Digital Enterprise Research Institute, Universität Innsbruck,AT
{firstname.lastname}@deri.org

Abstract. The Semantic Web envisions a Web where information is represented by means of formal vocabularies called ontologies for enabling automatic processing and retrieval of information. It is expected that ontologies will be interconnected by mappings forming a network topology. Reasoning with such a network of ontologies is a big challenge due to scalability issues. The local model semantics seems to be a promising approach in this direction that has served as the basis of several frameworks/distributed languages. The intent of the work described in this paper is to define a new framework for representing and reasoning with interconnected ontologies that will be based on a new language for representing ontologies and mappings called Distributed Open Answer Set Programming (DOASP). DOASP is a syntactical extension of OASP in the direction of distributedness and its semantics is a combination between the local model semantics and the OASP semantics. The reason for choosing OASP is the emerging interest in hybrid formalisms for the Semantic Web.

1 Research Context and Problem Statement

While the current Web is only usable by humans, the Semantic Web [1] envisions to make information processable and services on the Web usable by machines as well. The main technology for establishing the Semantic Web is the creation of so-called *ontologies*, which can be used to represent information and services on the Web. Ontologies are commonly defined as *formal specifications of shared conceptualizations* [2, 3] where the sharing aspect is important: ontologies have to be *reusable*.

This reusability does not mean that on the Web there will be a single ontology for capturing all the knowledge (not even all the knowledge corresponding to a given domain); there will be different ontologies that describe the same domain at different levels of granularity or from different perspectives, or that describe partially overlapping domains. In order to effectively use such overlapping ontologies one needs a means to describe the way they are interconnected. In particular, one uses logical axioms called *mappings* to relate elements of one ontology to elements of others.

In order to represent such ontologies and mappings, logical languages such as Description Logics (DLs) [4] or Logic Programming(LP) [5] can be used. The usage of formal languages allows in general for well-defined formal reasoning and thus the necessary automation of tasks such as checking consistency of ontologies. Some popular languages for representing ontologies anchored in one or both of the mentioned formalisms are WSML [6], OWL [7], SWRL [8].

Description Logics are attractive as the basis for ontology representation formalisms due to their many different decidable expressive fragments and their suitability for conceptual modeling as they adopt the Open World Assumption. However standard DLs lack support for non-monotonicity and have rather rigid constructs for expressing knowledge. Logic Programming (LP) on the other hand is a knowledge representation formalism with support for non-monotonic reasoning through the *negation as failure* operator. Moreover, the rule-based presentation of LP makes representing knowledge rather intuitive. A disadvantage of most LP approaches is their Closed World Assumption which is not realistic in an open like the Web, where knowledge is notoriously incomplete. One approach which tries to reconcile the use of negation of failure with the Open World Assumption which is characteristic to the Semantic Web is to restrict the use of negation in the form of so-called negation-as-failure [?]. Other approaches which try to combine advantages of the DL and LP paradigms are the so-called hybrid representation formalisms. Among them is also the language *Open Answer Set Programming* (OASP) [9–11] which keeps the rule-based presentation and the nonmonotonic capability from LP and drops the closed-domain assumption to allow for open domains as is the case in Description Logics (and first-order logic).

In the context of interconnected ontologies with mappings, an important factor, besides the syntax and semantics of the specific language(s) used for representing ontologies, is the choice for a semantics and algorithms for reasoning with the whole network of ontologies. The simplest approach, also called *global semantics* is to assume that all information relevant to a query (i.e., all the relevant ontologies and mappings) is put together and reasoning is performed as with a local ontology. Such an approach is considered for example in [12]. While this is a simple approach to deal with interconnected ontologies, it has some inconveniences: language specificity is constrained (ontologies and mappings have all to be expressed into formalism(s) which the particular reasoning engine supports), and local inconsistency propagates to the whole reasoning space. Also, computing a global model for the whole reasoning space imposes a heavy computational burden for the inferencing task.

Some existing work from the AI area of contextual reasoning seems to come handy for reasoning with networks of ontologies. In [13] a semantics called *local model semantics*, that captures the main intuitions behind context modeling, which are *locality* and *compatibility*, has been introduced. The advantages of this semantics over the simple approach described above are the possibility of using different local reasoners, thus also different formalisms for representing ontologies, the possibility to isolate local inconsistencies, and better scalability. It is common for the systems that implement this semantics to be based on distributed reasoning procedures.

Current approaches for representing and reasoning with interconnected ontologies using the local model semantics are based on Propositional Logic [14–17], Default Logic [18], First Order Logic [19, 20] and Description Logics [21, 22]. As concerns the reasoning support, propositional multi-context systems were put in correspondence with bounded modal logic K_n and shown that any contextual satisfiability problem can be reduced to that of satisfying some formula in K_n whose modal depth is at most equal to one [15]. Also contextual satisfiability problems were tractably encoded into purely propositional ones, which enables SAT-based implementations of such systems [14, 16].

A special case is that of Propositional Logic theories connected by bridge rules in which negation as failure can appear, the so-called information chain theory [17]. In this case a solution based on a fix-point operator that computes at each step a chain/anti-chain pair was devised. An algorithm based on a variation of the Well-Founded Semantics for Default Logic, WFS2 [23], was devised for reasoning with contextual default theories [18]. A similar algorithm (in the sense that it isolates local inconsistencies) based on the stable model semantics was devised for reasoning with ground ASP. A sound and complete calculus for DFOL based on ML [24] is presented in [19]. A distributed tableaux reasoning algorithm for DDL that works only for acyclic TBoxes is described in [25] and [21].

The main goal of the project is, given the advantages of OASP compared to DLs, and the benefits of the local model semantics, similarly to the extension of the DL semantics to the interconnected case in [21], to extend OASP to its distributed variant *Distributed Open Answer Set Programming* (DOASP). Moreover, I will investigate decidability of this extended framework (fragments) together with associated reasoning procedures.

2 Objectives and Approach

The main objectives of this work are:

- to define a language based both on LP and DL for representing ontology spaces, more specifically a distributed version of Open Answer Set Programming, called Distributed Open Answer Set Programming.
- to provide a declarative semantics for this distributed language in the style of Local Model Semantics described in [20].
- to design algorithms for reasoning with this language according to the proposed semantics.

In the following I give a short overview of how each of the precedent issues is/will be addressed together with corresponding evaluation criteria.

DOASP. Distributed Open Answer Set Programming is a language that extends Open Answer Set Programming in order to accommodate the representation of context spaces in accordance with the two principles underlying the notion of context: locality and compatibility. A context is a set of OASP rules. Every literal has attached the identifier of the context it is part of. The context of the head of the rule is always the context the rule makes part from. Rules that contain only literals belonging to the current context are local rules, while those who contain foreign literals in the body are bridge rules. Thus, it is possible for a bridge rule to connect more than two ontologies, depending on the number of foreign literals in the body.

Semantics. The semantics for DOASP is a blend between the local model semantics [13] and the OASP semantics. The most straightforward way to define it would be to simply extend the stable model based semantics of OASP to the distributed case similarly to the extension of the stable model semantics in the information chain approach. However, defining the semantics in such a way leads to the propagation of local inconsistencies, and does not lend itself to distributed reasoning. The possibility to define the semantics of DOASP similar with the well-founded based semantics defined for the information chain approach (the one based on a fix point which iterates over chain, anti-chain pairs) will be investigated, or furthermore, a para-consistent semantics like in the case of Con-DL might be considered.

The evaluation will consist in considering several concrete scenarios (DOASP encodings of real world situations) in order to justify the intuition behind the new semantics and the utility of the approach.

Algorithms. Most of the DL-based approaches which adopt the local model semantics are able to deal only with very restricted ontology spaces, while the non-monotonic approaches based on this kind of semantics rely on propositionalization. This, together with the fact that so far only decidable fragments of OASP were identified [11] without associated effective reasoning algorithms, points to the fact that the the identification of such algorithms is a non-trivial task. Note that due to the presence of open domains, propositionalization cannot be applied (at least, not straightforwardly) for reasoning with OASP as is commonly done with ASP.

In order to identify practical reasoning algorithms for reasoning with integrating languages, it is promising to investigate the relation with pure database languages. For example, in [26] the relation of Open Answer Set Programming with the database language Datalog LITE was established. Given the close relation of Datalog LITE with the relational algebra, one could build upon existing database techniques for reasoning with DOASP. Another direction of investigation is about the possibility of employing modularity results from the Stable Model Semantics community like splitting sets [27], Rosati's weak safeness condition [28], in order to identify what has to be evaluated together and what can be evaluated separately.

As full-fledged OASP is undecidable, and thus also DOASP, I will start with considering subsets of the language with reduced expressivity and then incrementally I will consider more expressive subsets for devising algorithms. As an evaluation criteria, the algorithms has to be proven as sound and complete.

An orthogonal direction for reasoning is the use of heuristics in order to maintain soundness but not necessarily preserving completeness. Performing an exhaustive search on the Web seems not desirable in many concrete scenarios, so pruning the context space or simplifying it might be an option.

3 Status, Future work

I consider myself to be at the end of the phase of defining the research problem. So far, the syntax and a declarative semantics for DOASP has been defined. I have been interested in this topic for around one and a half years. As future work, I plan to follow the steps mentioned in the previous section.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284** (2001) 34–43
2. Gruber, T.: An ontology for engineering mathematics. Technical report, KSL-94-18, Stanford, Ca (1994)

3. Borst, W.: Construction of engineering ontologies. Technical report, University of Tweenty. Enschede, NL-Centre for Telematica and Information Technology (1997)
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook. Cambridge University Press (2003)
5. Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming. In: Proc. of ICLP'88, Cambridge, Massachusetts, MIT Press (1988) 1070–1080
6. de Bruijn, J., eds.: The WSML Family of Representation Languages. Deliverable D16.1v0.2, WSML (2005) Available from http://www.wsmo.org/TR/d16/d16.1/v0.2/.
7. Dean, M., Schreiber, G., eds.: OWL Web Ontology Language Reference. (2004) W3C Recommendation 10 February 2004.
8. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. 21 may 2004, W3C (2004)
9. Heymans, S., Nieuwenborgh, D.V., Vermeir, D.: Open answer set programming with guarded programs. ACM Transactions on Computational Logic (TOCL) (2006)
10. Heymans, S., Nieuwenborgh, D.V., Vermeir, D.: Conceptual logic programs. Annals of Mathematics and Artificial Intelligence (Special Issue on ASP) (2006)
11. Heymans, S., Van Nieuwenborgh, D., Vermeir, D.: Guarded Open Answer Set Programming. In: LPNMR 2005. Number 3662 in LNAI, Springer (2005) 92–104
12. Haase, P., Wang, Y.: A decentralized infrastructure for query answering over distributed ontologies. In: Proc. of the 22nd Annual ACM Symposium on Applied Computing. (2007)
13. Ghidini, C., Giunchiglia, F.: Local model semantics, or contextual reasoning = locality+compatibility. Artificial Intelligence **127** (2001) 221–259
14. F. Roelofsen, L.S., Cimatti, A.: Many hands make light work: Localized satisfiability for multi-context systems. In: ECAI-04, Valencia, Spain (2004)
15. Serafini, L., Roelofsen, F.: Complexity of contextual reasoning. In: Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, California, USA (2004)
16. Serafini, L., Roelofsen, F.: Satisfiability for propositional contexts. In: KR' 04). (2004) 369
17. Roelofsen, F., Serafini, L.: Minimal and absent information in contexts. In: (IJCAI-05), Edinburgh, Scotland (2005)
18. Brewka, G., Roelofsen, F., Serafini, L.: Contextual default reasoning. In: Proc. IJCAI-07, Hyderabad, India (2007)
19. Ghidini, C., Giunchiglia, F.: Distributed first order logic. Frontiers of Combining Systems 2, Research Studies Press Ltd. (2000) 121–139
20. Ghidini, C., Serafini, L.: Distributed first order logic. Technical report, ITC-irst (1998)
21. Serafini, L., Tamilin, A.: Drago: Distributed reasoning architecture for the semantic web. Technical report, ITC-irst (2004)
22. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing ontologies. Journal of Web Semantics **1** (2004)
23. Brewka, G., Gottlob, G.: Well-founded semantics for default logic. Fundamenta Informaticae **31** (1997) 221–236
24. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics. AI (1994) 29–70
25. Serafini, L., Tamilin, A.: Local tableaux for reasoning in distributed description logics. (In: Proc. of the 2004 Intl. Workshop on Description Logics (DL2004))
26. Heymans, S., Van Nieuwenborgh, D., Vermeir, D.: Guarded Open Answer Set Programming with Generalized Literals. In Dix, J., Hegner, S., eds.: FoIKS 2006. (2006) 179–200
27. Lifschitz, V., Turner, H.: Splitting a logic program. In Hentenryck, P.V., ed.: Proc. of Eleventh Intl Conf. on Logic Programming, San Jose, Calif. (1994) 23–37
28. Rosati, R.: On the decidability and complexity of integrating ontologies and rules. **3** (2005) 41–60