Combining SWRL rules and OWL ontologies with Protégé OWL Plugin, Jess, and Racer

Christine Golbreich & Atsutoshi Imai

Laboratoire d'Informatique Médicale, Faculté de Médecine, Université Rennes 1

Av du Pr. Léon Bernard, 35043 Rennes, France

Christine.Golbreich@univ-rennes1.fr, atsutoshi.imai@etudiant.univ-rennes1.fr

Abstract. The presentation concerns a draft implementation with Protégé OWL Plugin for SWRL, based on the RDF concrete syntax of the SWRL proposal. A first prototype of a SWRL Tab Widget has been achieved. It is a bridge between Protégé OWL, Racer, and Jess, intended to help reasoning with SWRL rules combined with OWL ontologies. A small example is given including an OWL ontology representing the family usual relationships, and a SWRL rule base representing the dependencies between those relationships. The goal is to illustrate that a key requirement is to interoperate between SWRL rules and OWL, not only syntactically and semantically, but also *inferentially* to get all the inferences.

1. Introduction

Most people now rather agree that a Web rule language is needed. According to the Semantic Web stack, rules are on the top of ontology. A Web rule language should be useful to express different kinds of rules: "standard-rules", for chaining ontologies properties, such as the transfer of properties from parts to wholes, "bridging-rules" for reasoning across domain, "mapping rules" between Web ontologies for data integration, "querying-rules" for expressing complex queries upon the Web, "meta-rules" for facilitating ontology engineering (acquisition, validation, maintenance) [6] etc. The recent proposal for a Semantic Web Rule Language SWRL[1], based on a combination of OWL DL and OWL Lite [1] with the Unary/Binary Datalog RuleML [2] sublanguages is a first important step. But a step further is needed for interoperating between SWRL and OWL, not only syntactically and semantically, but also *inferentially*. In other words, it is not enough to be able to create SWRL rules in OWL that can use the vocabulary of an OWL ontology, but a key requirement is *to reason* in a semantically consistent way, that is to exploit both the ontology and the rule base knowledge to draw inferences. Different solutions are being investigated like DLP fusion [7], but such theoretical issues are out of the scope of this abstract. Motivated by medical ontologies, where rules are needed to express the propagation of properties, e.g. the transfer of properties from parts to wholes, or the brain-cortex anatomy dependencies [3], we investigate how to combine OWL ontologies and SWRL rules for reasoning. We present a pragmatic approach using the available Protégé tools and widgets. A draft implementation for SWRL has been achieved in Protégé OWL, as close as possible to the RDF concrete syntax and swrl.owl OWL ontology of the SWRL proposal. A first prototype of a SWRL Tab Widget has been implemented to bridge between Protégé OWL[2] Racer [8], and Jess [5] for reasoning with SWRL rules combined with OWL ontologies. A small example including an OWL ontology representing the family usual relationships, and a SWRL rule base representing the dependencies between those relationships, illustrates the needs to interoperate between SWRL rules and OWL *inferentially*, in order to get all the inferences.

---

[1] http://www.daml.org/rules/proposal/

[2] http://protege.stanford.edu/plugins/owl/

2. The OWL ontology for SWRL

The implemented swrl.owl ontology (Fig. 1) is close to the proposal's one, although it exhibits some differences for instance, 'head' is voluntary limited to at most one Atom, 'body' is defined as a collection of multiple Atom, instead of using List, etc. Rules are represented as instances of Imp. Like in the proposal, this ontology is OWL full. For instance the range of the classPredicate property of Atoms is Class. But since here SWRL only serves to support the edition of rules, which are next mapped to Jess rules for reasoning, it does not matter.

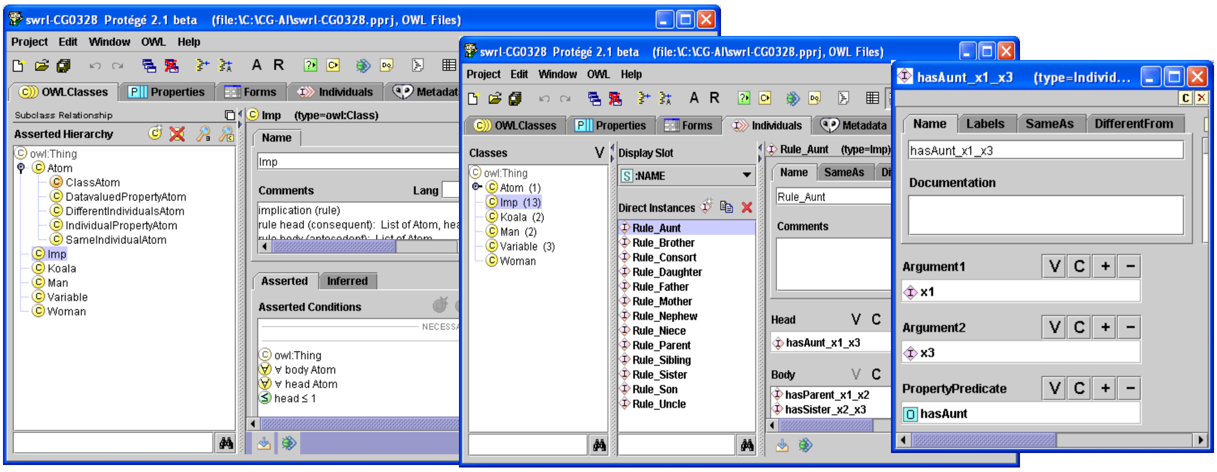

Fig. 1:The SWRL.owl ontology and a SWRL rule base in Protégé OWL

3. SWRL Tab Widget

A SWRL Tab Widget has been achieved for reasoning with OWL and SWRL. It allows mapping SWRL rules instances to Jess rules, OWL individuals to Jess facts. Using it, it is possible to infer knowledge from an OWL ontology knowledge base combined with a SWRL rules base, both described using the Protégé OWL editor. The OWL plugin is used for editing the ontologies, Racer for classifying them, JessTab [3] for integrating Jess and Protégé, Jess is used as a rule inference engine, Rice and Racer for visualizing the inferred results.

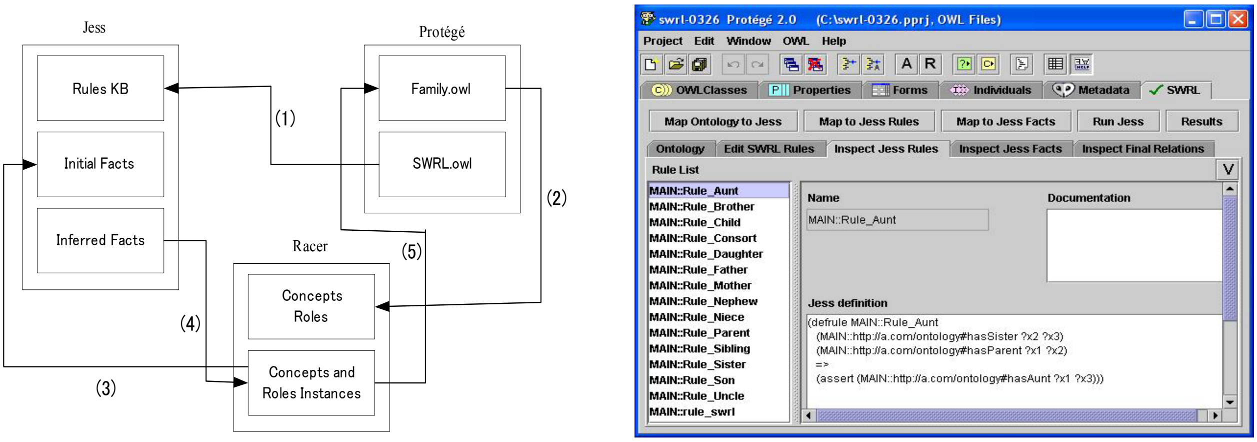

Fig. 2: Using the SWRL Tab Widget

The overall process (Fig. 2 left) consists of the following steps:

1. SWRL rules instances are mapped to Jess rules with the "Map to Jess Rules" button of the SWRL Tab Widget (Fig. 2) and can be inspected with the "Inspect Jess Rules" Tab.
2. The Racer Abox roles instances, corresponding to Protégé properties from the OWL ontology, are mapped to Jess facts with the "Map to Jess Facts" button and can be inspected with the "Map to Jess Facts" Tab.
3. Jess rule engine is executed thanks to the "Run Jess" button. The new Jess facts inferred from the Jess rules and initial facts, can be inspected with the "Inspect Jess Facts" Tab.

4. The new facts are converted into Racer role assertions with the “Results” button and can be inspected with “Inspect Final Relations”, clicking the “Showgraph” button in RICE.

4. The family example

Protégé OWL plugin has been used to define an OWL ontology describing the usual classes, `Person`, `Man`, `Woman`, `Child`, `Daughter`, `Son`, `Parent`, `Father` etc., and relationships, `hasConsort`, `hasSex`, `hasChild` etc within a family, (Fig. 3 left). A Person is characterized by 12 properties. 10 Man and 10 Woman individuals have been defined, having the relations depicted (Fig. 3 right). Only the properties `hasConsort`, `hasChild`, `hasSex` were initially filled for men and hasConsort, hasSex for women. A SWRL rule base representing some dependencies between the properties has been edited with the Protégé OWL Plugin, including rules such as the Rule_Aunt rule: `hasParent(?x1,?x2) Λ hasSister(?x2,?x3) ⇒ hasAunt(?x1,?x3)`

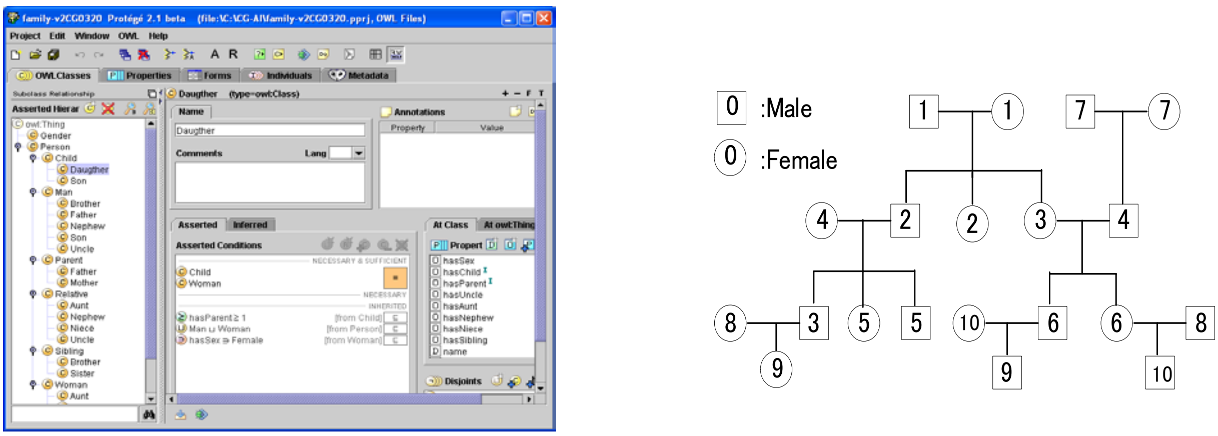

Fig. 3: The family ontology and the initial relations between its individuals

-1- Person := Man ∪ Woman

-2- Parent := Person ∩ hasChild ≥ 1 (a) Child := Person ∩ hasParent ≥ 1 (b) (hasChild) $^{-1}$ = hasParent (c)

-3- Father := Parent ∩ Man Mother := Parent ∩ Woman

-4- Son := Child ∩ Man Daughter := Child ∩ Woman

-5- Brother := Sibling ∩ Man Sister := Sibling ∩ Woman

-6- Nephew := Man ∩ (hasUncle ≥ 1 ∪ hasAunt ≥ 1)

-7- Relative := Child ∪ Parent ∪Aunt ∪ Nephew ∪ Niece ∪ Uncle ∪ Sibling

-8- hasParent(?x1,?x2) Λ hasConsort (?x2,?x3) ⇒ hasParent(?x1,?x3)

-9- hasParent(?x1,?x2) Λ hasSister(?x2,?x3) ⇒ hasAunt(?x1,?x3)

-10- hasSibling (?x1,?x2) Λ hasDaughter(?x2,?x3) ⇒ hasNiece(?x1,?x3)

MR1- hasSibling(?x1,?x2) Λ Man(?x2) ⇒ hasBrother(?x1,?x2)

MR2- hasSibling(?x1,?x2) Λ Woman(?x2) ⇒ hasSister(?x1,?x2)

MR3- hasParent(?x1,?x2) Λ Man(?x2) ⇒ hasFather(?x1,?x2)

Fig. 4: Extracts from the family.owl ontology, SWRL rule base (8-9-10), rules mirroring the ontology

First, from the family ontology definitions (see above extracts), a new hierarchy is inferred in using Racer classification. From their initial relations, the 20 individuals are identified to be member of the following classes. All the individuals are identified to be Person since they are men or women (-1-), the 7 Men are identified to belong to the Parent class since they have a child (-2b-) and also to Father since they are man (-3-), 12 individuals are identified to be Child since the hasChild relation is asserted as inverse of hasParent (-2c-). Among them 7 are Son since they are Man, 5 are

Daughter since they are Woman (-4-), 15 individuals are identified to belong to Relative because they are either child, or parent etc. (-5-). Second, from these initial facts, executing the Jess Rule engine with the SWRL rule base, enables to infer 7 additional individuals that belong to Parent, 4 to Brother, 4 to Sister, 8 to Sibling, and 5 additional ones to Relative. Thus, using Racer and Jess separately, does not provide all the relations. Finally, adding Jess rules mapping the ontology knowledge to Jess, like MR1 MR2 MR3 etc. (Fig. 4) enables to infer that 3 individuals belong to Uncle, 4 to Aunt, 5 to Nephew, 3 to Niece. Indeed for instance, chaining MR2 and rule (-9-) entails new relations like hasAunt, and consequently Nephew individuals from definition (-6-). At that time, the ontology is mapped to Jess using additional SWRL rules, but we investigate the use of JessTab for automatically mapping OWL ontologies to Jess.

5. Conclusion

Although some knowledge can be represented both by DL and rules, and even if some DL extensions are possible (for example like "role inclusion axioms" to allow the transfer of properties along another property [9]), (i) their expressiveness is usually different, (ii) each paradigm better fits some particular type of knowledge and reasoning. OWL DL is well suited to represent "structured" knowledge by classes, properties and taxonomies, supporting automatic classification and class recognition of instances. SWRL is better suited to express "deductive" knowledge by rules composed of atoms, supporting answering queries and ground atoms entailment. From our experience with medical ontologies, both paradigms and services are useful. But as the 'family' ontology example clearly enlightens, to get all the inferences, the reasoning shall not be separately performed by an OWL classifier and a rule engine. The OWL DL terminology and SWRL rules components have to be closely integrated. An open issue to discuss is whether, they have to be merged within a single *Protégé OWL ontology*, or rather translated and merged within a *single logic program* exploited by a reasoning engine. It depends on whether Protégé OWL is considered as an editor used to create the two components, or as an editor of a future hybrid logical language combining parts of OWL DL and SWRL. The 'family' ontology example illustrates another point: from only 3 initial properties, the SWRL rules enabled to generate all the family relations represented with 12 properties. A motivating perspective is to propose a methodology based on such a designed rule base, serving as a "conceptual" model for building OWL ontologies and for their validation. This work is still ongoing. Since we met some practical difficulties, the different tools still being evolving or not already completely documented, we are aware that the present widget has some limitations, which might be overcome. However, we hope this work may help for future more reliable developments, and might be soon applied for building medical ontologies.

6. References

1. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I. McGuinness, L. D., Patel-Schneider, P. F., Stein, L., A.:OWL Web Ontology Language Reference. W3C Working Draft (2003)
2. Boley,H., Groso,B., Sintek,M., Tabet, S., Wagner, G. RuleML Design, Rule Markup Initiative (2002)
3. Dameron O., Gibaud B., Musen M. Using semantic dependencies for consistency management of an ontology of brain-cortex anatomy, KR-MED 2004
4. Eriksson, H.:Using JessTab to Integrate Protege and Jess. IEEE Intelligent Systems (2003), 18(2):43-50
5. Ernest J. Friedman-Hill: Jess 6.1 manual. Sandia National Laboratories (2003)
6. Golbreich, C., Dameron, O., Gibaud, B., Bur gun A. Web ontology language requirements w.r.t expressiveness of taxononomy and axioms in medecine, ISWC 2003, Springer.
7. Grosof B N.., Horrocks Ian, Volz Raphael, and Decker Stefan Description Logic Programs: Combining Logic Programs with Description Logic In: Proc. 12th Intl. Conf. on the World Wide Web (WWW-2003), Budapest, Hungary, May 20-23, 2003.
8. Haarslev V., Möller R.: Description of the RACER System and its Applications. Description Logics 2001
9. Horrocks, I., Sattler, U., Decidability of SHIQ with Complex Role Inclusion Axioms, IJCAI 03