Visualizing SWRL Rules: From Unary/Binary Datalog and PSOA RuleML to Graphviz and Grailog

Ismail Akbari, Bo Yan, Junyan Zhang, Harold Boley
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
Email: {iakbari, b.yan, boliuy12, harold.boley} AT unb.ca

***Abstract*- SWRL rules are transformed in two steps for visualization in a subset of Grailog. A Unary/Binary Datalog rule in SWRL presentation syntax is translated to a corresponding PSOA RuleML rule in a RIF-like presentation syntax employing frame formulas. This is then translated to the Graphviz DOT language so that the Graphviz tool can render it visually as a Grailog graph with an object identifier and slots. Supported by the obtained visual graphs, users can more easily analyze the original symbolic logic rules.**

Keywords*- *Semantic Web; SWRL rules; Unary/Binary Datalog; F-logic; PSOA RuleML; Grailog; Visualization; Graphviz; Transformation

I. INTRODUCTION

The Semantic Web Rule Language (SWRL) [1] combines the sublanguages Web Ontology Language Description Logic (OWL DL) with the Unary/Binary Datalog RuleML sublanguage of the Rule Markup Language. The Graph inscribed logic (Grailog) has been introduced as a systematic graph standard for visual-logic knowledge [2]. This work uses transformations targeting the Graphviz tool [8] to visualize SWRL rules as Grailog 1.0 graphs. SWRL rules are translated to corresponding PSOA (Positional-Slotted, Object-Applicative) RuleML [6] frame rules, which are then translated to the Graphviz DOT language for rendering as Grailog graphs.

II. LANGUAGES AND TOOLS

There exist many methods and tools to visualize data and knowledge [3] in diverse areas. One of these areas is the Semantic Web, whose knowledge can be visualized via Directed Labeled Graphs (DLGs) and DLG-extending Grailog graphs.

A. OWL DL and OWL Lite

OWL achieves machine interpretability of Web ontologies by providing an XML syntax and a formal semantics [4]. SWRL's sublanguage OWL DL supports users who want high expressiveness while retaining computational completeness and decidability. OWL DL's sublanguage OWL Lite supports those users primarily needing a classification hierarchy and simple constraints.

B. Frame Logic

Frame logic (F-logic) is a frame-based language using slot-described objects typed by classes that are organized as a light-weight ontology (taxonomy) [5]. The semantics of F-logic makes the closed world assumption as opposed to the open world assumption of description logics. Also, F-logic is generally undecidable whereas OWL DL is decidable.

C. PSOA RuleML

PSOA RuleML is a rule language that deeply integrates relational (predicate-based) and object-centered (frame-based) modeling. In PSOA RuleML, the notion of a PSOA term is introduced as a generalization of: (1) the positional-slotted term in POSL [10] and (2) the frame term and the class membership term in F-logic and RIF-BLD [6].

D. Graphviz

Graph Visualization Software (Graphviz) is a package of open source tools that was introduced by AT&T Labs Research for graph drawing, e.g. via DOT language scripts [7]. Graphviz layout programs take the description of graphs in a simple text file based on the DOT language script format and generate diagrams (graphs) in the desired output format [8].

III. UNARY/BINARY FRAME DATALOG

In Grailog, we extend Unary/Binary Datalog with frames. A unary relation is a class pointing to the relation's single argument as the node it types. A binary relation describes a relationship between two nodes.

A. Frame Formulas: Associating Slots with an Object Identifier

Slots in Grailog are drawn as special, bullet-attached arrows distinguishing a start node as playing the role of the Object IDentifier (OID). In Unary/Binary Frame Datalog, a node (an instance or a variable), acting as the OID of a frame, can be pointed to by a class-originating arrow for ('unary') typing and can have outgoing slot arrows. The same node can also act as the first or second argument of a binary relation, drawn as a regular (bullet-free) arrow. See figure 1 for an example.

IV. Structure of the Implementation

The main steps of our prototype implementation are as follows. First, the tool receives SWRL's (Unary/Binary) Datalog rules from the input and translates them into Frame Datalog. Next, it splits each rule into its components, including instances, classes and slots, written to a text file. From these components, it then generates the Graphviz DOT file. Finally, it calls Graphviz for the visual rendering of the graph output.

V. SWRL-to-PSOA Transformation

This section describes how to transform Datalog SWRL rules to Frame Datalog PSOA RuleML rules, used by our Grailog visualization and reusable generally. SWRL rules use a conjunctive formula as premise and as conclusion. After receiving a SWRL rule, it will be translated to a Frame Datalog rule in PSOA RuleML. As an example, consider the following SWRL rule. The "?" symbol indicates variables and "^" denotes conjunction:

```
Person(?x) ^                                    (1)
Man(?y) ^
hasAge(?x,?age1) ^
hasAge(?y,?age2) ^
hasSibling(?x,?y) ^
swrlb:greaterThan(?age2,?age1)
->
hasOlderBrother(?x,?y)
```

This is translated to the following PSOA RuleML rule, whose first two premises represent single-slot frames, where the term f(t) encodes the slot f->t:

```
hasOlderBrother(?x ?y) :-                        (2)
        And(?x#Person(hasAge(?age1))
            ?y#Man(hasAge(?age2))
            hasSibling(?x ?y)
            swrlb:greaterThan(?age2 ?age1))
```

The frame premises check that object "?x" of class "Person" has property "hasAge" with value "?age1" and object "?y" of class "Man" has property "hasAge" with value "?age2".

VI. Illustrative Rule Rendering

An example is used to show the tool's operation. Consider formula (1) as the SWRL rule input. Its transformation to formula (2) and further processing described in [9] lead to the output (the graph) shown in figure 1. The red arrows show the premises of the rule. The green arrow shows its conclusion. Recall that a bullet distinguishes the OID of a slot arrow. An oval shows a class and an octagon shows a variable.

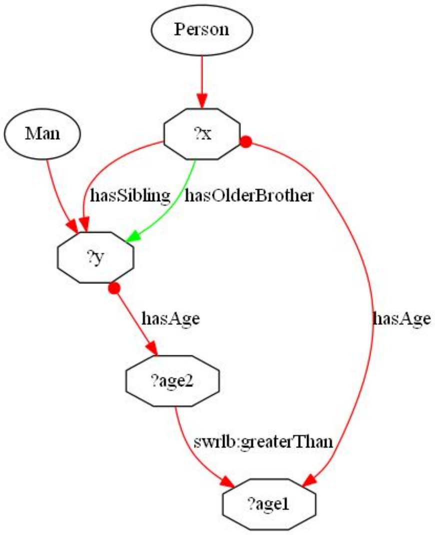

Figure 1. Graph rendered from SWRL rule in formula (1)

VII. Conclusion

Our tool transforms SWRL from Unary/Binary Datalog rules to Frame Datalog PSOA RuleML. The Graphviz-rendered visualization of frame rules as Grailog graphs lets people more easily analyze the logic of SWRL rules. By visualizing SWRL rules, this work is an implementation of a Grailog 1.0 subset. A demo and more details about the implemented system are online [9].

References

[1] Horrocks, Ian, et al. SWRL: A Semantic Web rule language combining OWL and RuleML.W3C subm. 21 (2004): 79.

[2] Harold Boley. Grailog 1.0: Graph-Logic Visualization of Ontologies and Rules. Proc. RuleML 2013, Seattle, Washington, USA, July 2013, Springer LNCS 8035. Preprint: http://cs.unb.ca/~boley/papers/GrailogVisOntoRules.pdf

[3] Goldstein, Ira P. The genetic graph: a representation for the evolution of procedural knowledge. International Journal of Man-Machine Studies 11.1 (1979): 51-77.

[4] Pascal Hitzler, et al., OWL 2 Web Ontology Language Primer (Second Edition), W3C Consortium, Recommendation REC-owl2-primer-20121211, Dec 2012.

[5] Michael Kifer, et al.,. Logical foundations of object-oriented and frame-based languages. Journal of the ACM (JACM) 42.4 (1995): 741-843.

[6] Harold Boley. A RIF-Style Semantics for RuleML-Integrated Positional-Slotted,Object-Applicative Rules, Proc. 5th Int'l Symposium on Rules: Research Based and Industry Focused (RuleML-2011 Europe), Barcelona, Springer, July 2011.

[7] Koutsofios, Eleftherios, and Stephen North. Drawing graphs with dot. Technical Report 910904-59113-08TM, AT&T Bell Laboratories, Murray Hill, NJ, 1991.

[8] http://graphviz.org/

[9] http://2012team8project.weebly.com/index.html

[10] Harold Boley. Integrating Positional and Slotted Knowledge on the Semantic Web, *JETWI* 2(4):2010, 343-353.