

# Rule Modeling Using Semantic MediaWiki

Jie Bao, Li Ding

Department of Computer Science,  
Rensselaer Polytechnic Institute,  
Troy, NY 12180, USA.  
baojie@cs.rpi.edu  
dingl@cs.rpi.edu

Paul R. Smart

School of Electronics and  
Computer Science,  
University of Southampton,  
Southampton, SO17 1BJ, United Kingdom.  
ps02v@ecs.soton.ac.uk

Dave Braines, Gareth Jones

Emerging Technology Services,  
IBM United Kingdom Ltd, Hursley Park,  
Winchester, Hampshire,  
SO21 2JN, United Kingdom.  
dave\_braines@uk.ibm.com  
garethj@uk.ibm.com

**Abstract**—Semantic MediaWiki (SMW) is a popular semantic wiki engine that enables collective knowledge modeling within the expressivity of a small subset of OWL. However, SMW is limited in offering native support for modeling rules. In this paper, we show that by using templates and semantic queries provided by SMW, it is possible to model several types of rules within SMW, e.g., OWL entailment rules and logic programs. Such modeling practice enables us to perform some broadly useful inference tasks on SMW, e.g., integrity constraint checking.

## I. INTRODUCTION

Semantic MediaWiki (SMW)<sup>1</sup> is an extension of MediaWiki, the wiki engine used by Wikipedia, that allows users to add simple semantic annotations (such as categories and typed links) to wiki contents. SMW is probably the most successful semantic wiki engine with wide adoption and an active user community. One of the limitations of SMW, however, is its lack of native support for modeling rules. Thus, it does not offer the ability to infer new facts from explicitly stated facts in the wiki. This limitation results in the absence of some features that would otherwise be highly useful. For example, SMW does not provide inferences about the domain and range of a property.

To overcome this limitation, we show that by using several extensions of SMW it is possible to model several types of rules. In the current paper, we report the results of our rule modeling efforts with respect to two rule types: OWL entailment rules and logic programs. We also demonstrate how SMW rule modeling capabilities can be used to support rule-based integrity constraint checking.

SMW has been used in ITA to demonstrate its ability in intelligence collection and plan modeling [1]. It may serve as a tool for military personnel (e.g., soldiers) to codify their first-line observations and facilitate automated debriefing. However, due to the lack of rule inference support in SMW, the current efforts in ITA are limited in addressing some critical needs, e.g., modeling complex operational plans. This paper provides some foundations that may help addressing these issues.

## II. RULE MODELING IN SMW

We now describe a general strategy in modeling rules within SMW. The modeling of rules in SMW is enabled by several

components or extensions of MediaWiki and SMW, including the following:

- “Ask” Query (part of SMW): SMW provides a simple query language using the “ask” function<sup>2</sup>. An ask query retrieves facts from the wiki based on its query conditions, and presents the query result in a user specified way.
- Semantic Template (part of SMW): a semantic template is a “template” wiki page that can take parameters (in the form of {{{para}}}), and specifies the layout, content and semantic annotations of its instances (which are also wiki pages).
- Parser Function<sup>3</sup>: this is an extension to MediaWiki that includes the support for conditional content generation, using the syntax: `{{#if: condition | A | B }}`. If “condition” is true, “A” will be generated, otherwise “B” will be generated.

- Arraymap (part of SMW): this is a loop function.

The general strategy we adopted is to model a rule as a template. The antecedents of a rule are specified by a user or by a semantic query; the satisfaction of the antecedents is tested using a “if” or an “arraymap” (when a loop is necessary) parser function, and the consequent of the rule corresponds to some new semantic annotations. For example, suppose we have the following wiki script on a page “Template:Uncle”, which specifies “if X is a *brother of* Y and Y is the *father of* Z, then X is an *uncle of* Z”.

```
{{#arraymap:
  {{getValue|[[{{{PAGENAME}}}]]|brother of}}|,|Y|
  {{#arraymap:{{getValue|[[Y]]|father of}}|,|Z|
    [[uncle of::Z]]  }} }}
```

Where “getValue” is a template such that `{{getValue|A|P}}` will return list of property P values of the wiki page A. It is essentially an “ask” query:

```
{{#ask: [[{{{1}}}] ]?{{{2}}}=
  |mainlabel=- |format=list |link=none}}.
```

To invoke this rule at a page X, we can simply add the script `{{Uncle}}` to X. Thus if X is related to any wiki page Z via the “brother of” and “father of” property chain, then X also has the annotation “`[[uncle of::Z]]`” (the SMW script that states that X has an “uncle of” property value Z).

## III. CONCRETE RULE MODELING SCENARIOS

In this section we illustrate the modeling of several broadly useful types of rules.

<sup>2</sup>A Mediawiki function is an extension to the basic Mediawiki script syntax, in the form of `{{#f: ...}}` where f is the function name.

<sup>3</sup><http://meta.wikimedia.org/wiki/ParserFunctions>.

<sup>1</sup><http://semantic-mediawiki.org>. In this paper we study SMW 1.4.

## A. Modeling OWL Entailment Rules

Inference with OWL (Web Ontology Language) may be realized using a set of entailment rules<sup>4</sup>. In general, OWL entailment rules may generate sound but not necessarily complete inference results. While SMW has only limited native support for inference, we show that some of those rules can be implemented using the aforementioned template-based approach. For example, the following wiki script on the page “Template:Domain” defines the “rdfs:domain” inference rule:

```
{{#if:{{getValue|[[[:{{PAGENAME}}]]|{{1}}]}|
|[[Category:{{2}}]]}}
```

Thus, if “{{Domain|has author|Document}}” is added to a wiki page, then whenever the page has a “has author” property value, the page must be an instance of Category:Document. In practice, it is often useful to use another template page as a container of all the rules a user wants to use, and have the container template directly or indirectly (via some template inclusion chain) included on relevant pages. In general, this approach will render sound, but not necessarily complete, inference results. Many other OWL entailment rules can be defined in a similar manner<sup>5</sup>.

## B. Modeling Logic Programs

Logic Programs (LP) are a widely used family of rules. A logic program is a set of rules of the form “ $H :- B_1, \dots, B_n$ ”, where  $H$  is the *head* of the rule, and  $B_i$  is a *body* literal (a body literal may be a positive or a negative *atom* predicate). In LP, “not” indicates negation as failure. For example, a statement that “every person is by default right-handed, unless it is known that the person is left-handed” (this is known as a *default* rule) can be represented as a LP rule:

```
RightHanded(x) :- Person(x), not LeftHanded(x).
```

To model this rule in SMW, we use the following wiki script:

```
{{LP Rule
|body= 1::Person; 1:not:LeftHanded
|head= RightHanded }}
```

Where “LP Rule” is a template name, “1::Person” indicates a unary positive body literal and “1:not:LeftHanded” indicates a unary negative body literal. “Template:LP Rule” is designed in such a way that whenever the body literals are all true, the head will be added as an inferred fact (e.g., as [[Category:RightHanded]]). Due to space limitations, the full definition of the LP Rule template is given online<sup>6</sup>.

## C. Application: Integrity Constraint Checking

We now show a specific application of rule modeling, namely integrity constraint (IC) checking. An IC rule describes when and how a dataset is not complete. From the modeling point of view, an IC rule can be represented as a special type of LP rule with an empty head. For example, if we want to

specify that “Every person who is not a child should have a SSN number”, we can represent this constraint as a LP rule:

```
:- Person(x), not Child(x), not Has_SSN(x,y)
```

We also implement IC rules in SMW in a way that is similar to that seen for LP rules. This approach supports a number of IC

The screenshot shows a section titled "Integrity Constraints" with a sub-section "Violation:" containing a bullet point: "Rule.SSN (Every person who is not a child should have a SSN number)". Below this is a section titled "Facts about Person 3" with a sub-section "Apply IC" showing the rule being applied: "Rule.SSN + [?] and Rule.Name + [?]". Other facts listed include "CnlLabelChinese: Person 3 + [?]", "Has name: John + [?]", "Owl:hasValue: Property:Has name:John + [?]", "Rdf:type: Category:Person + [?]", and "Violate IC: Rule.SSN + [?]".

Fig. 1. IC Checking Example

diagnostic capabilities: it automatically evaluates instances that violate an IC rule, and it generates a warning on wiki pages containing instances that violate an IC rule (e.g., see Fig. 1). A more complete example is available online<sup>7</sup>.

## IV. DISCUSSION AND CONCLUSION

In this paper, we have shown that by combining SMW with several other MediaWiki extensions, it is possible to support the modeling of several types of rules, e.g. OWL entailment rules and logic programs. We have also shown an application of rule modeling to the problem of integrity constraint checking in SMW.

The most closely related work is the rule extension prototype of SMW from the Vulcan project halAR<sup>8</sup>. This prototype allows users to encode F-Logic rules in order to derive inferred knowledge in the wiki, relying on an external rule engine. In contrast, our approach does not require any external rule engine. Our approach also allows rules to be locally invoked by controlling template inclusion. This may help to deal with issues of scalability in rule execution.

Our future work will focus on the encoding of OWL 2 RL entailment rules in SMW. We will also explore a number of ways to support rule authoring using supportive user interfaces.

## ACKNOWLEDGEMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

## REFERENCES

- [1] Jie Bao, Paul R. Smart, Nigel R. Shadbolt, and Dave Braines. A Controlled Natural Language Interface for Semantic Media Wiki Using the Rabbit Language. In *Workshop on Controlled Natural Language (CNL'09) (In Press)*, 2009.

<sup>4</sup><http://www.w3.org/TR/owl2-profiles>

<sup>5</sup>For OWL entailment rules that rely on constructs not natively supported by SMW, they can be designed with respect to an OWL meta-model extension [1].

<sup>6</sup>[http://tw.rpi.edu/dev/cnl/LP\\_Test](http://tw.rpi.edu/dev/cnl/LP_Test)

<sup>7</sup>[http://tw.rpi.edu/dev/cnl/Integrity\\_Constraint](http://tw.rpi.edu/dev/cnl/Integrity_Constraint)

<sup>8</sup><http://ruledemo.ontoprise.com>