Concept Modeling on Semantic Wiki

Presented by Jie Bao, RPI
baojie@cs.rpi.edu
http://baojie.org

Joint work with Li Ding and Zhenning Shangguan

SemanticWiki mini-series session-3
Dec 11, 2008

Summary

- SMW supports a subset of OWL modeling
- Using meta-modeling, we can extend SMW to support
  - (more complete) OWL Modeling
  - Simple rule modeling
  - Integrity constraints.

Useful Extensions

- Semantic Template (part of SMW)
- Parser Function
  http://meta.wikimedia.org/wiki/ParserFunctions
- Regular Expression
  http://www.mediawiki.org/wiki/Extension:RegexFunctions
- String Function
  http://www.mediawiki.org/wiki/Extension:StringFunctions
- Variables
  http://www.mediawiki.org/wiki/Extension:VariablesExtension
- Set operation:
  diff, union and intersection (will release soon)

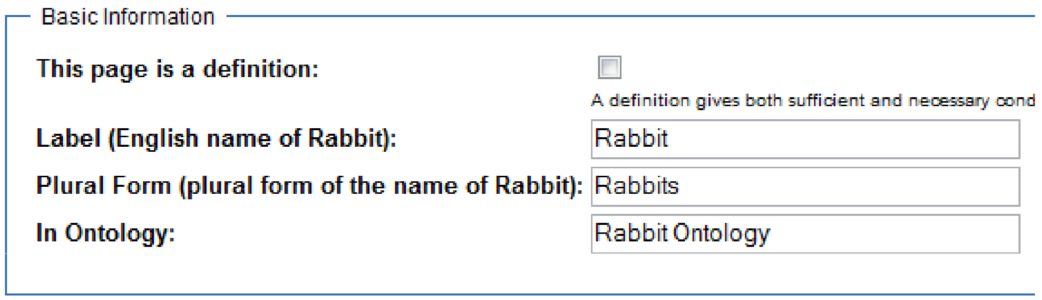
1. OWL Modeling

- Goal: Encoding OWL ontology constructs (e.g., axioms) on SMW pages
  - Class
  - Property
  - Individual
- Solution: Using templates and semantic forms to help users maintain ontological description

1. OWL Modeling (cont.)

OWL Abstract Syntax:

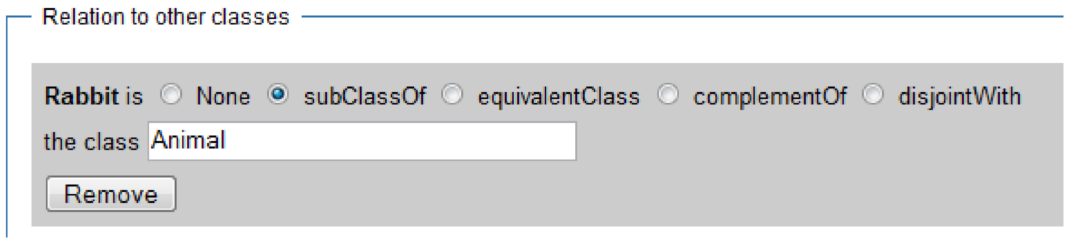
Class(Rabbit partial intersectionOf (Animal
restriction(eat someValuesFrom(FreshVegatable))))



Basic Information

This page is a definition:	☐ A definition gives both sufficient and necessary cond
Label (English name of Rabbit):	Rabbit
Plural Form (plural form of the name of Rabbit):	Rabbits
In Ontology:	Rabbit Ontology

```
{{Class
| is definition=No
| label=Rabbit
| plural=Rabbits
}}
```

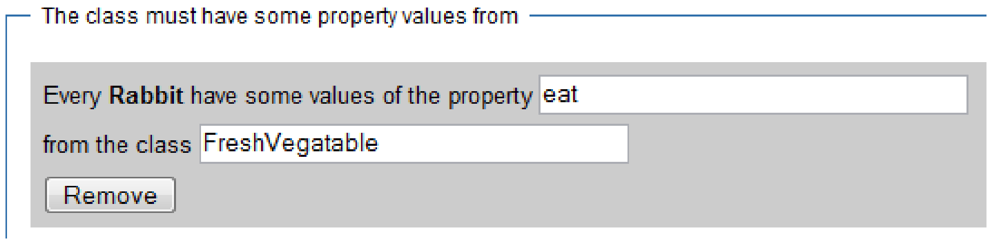


Relation to other classes

Rabbit is ○ None [✓ subClassOf] ○ equivalentClass ○ complementOf ○ disjointWith

the class Animal

```
{{ClassRelation
| type=subClassOf
| class=Animal
}}
```



The class must have some property values from

Every Rabbit have some values of the property eat

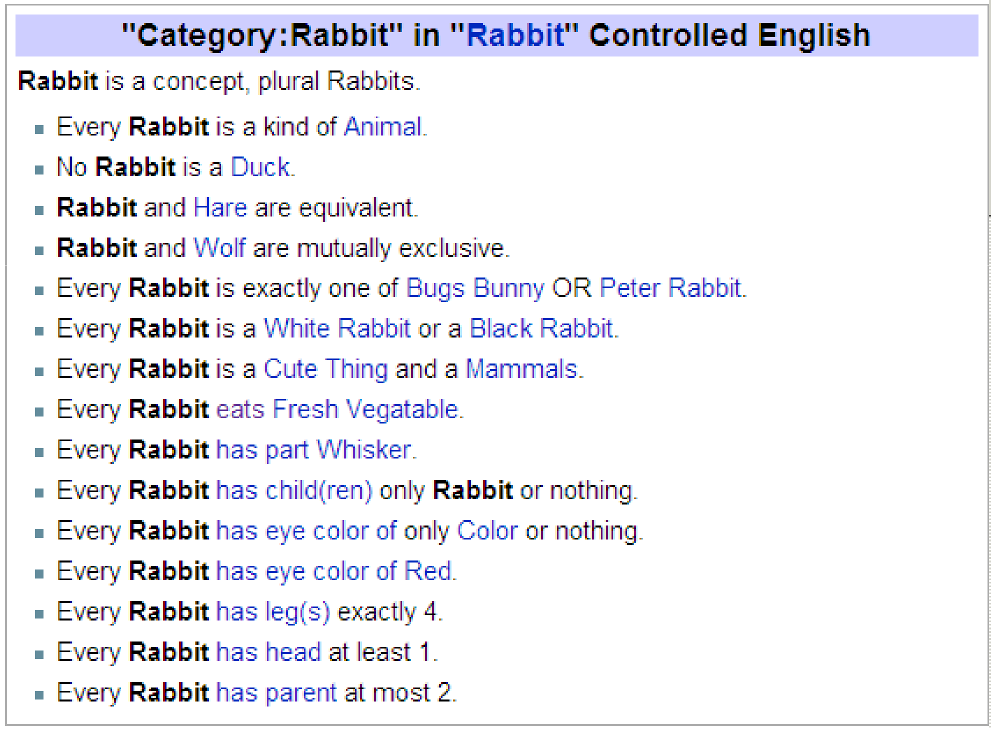
from the class FreshVegatable

```
{{Some
| on property=eat
| on class=FreshVegatable
}}
```

http://tw.rpi.edu/dev/cnl/Category:Rabbit

1. OWL Modeling (cont.)

- Controlled Natural Language Interface



"Category:Rabbit" in "Rabbit" Controlled English

Rabbit is a concept, plural Rabbits.

- Every **Rabbit** is a kind of Animal.
- No **Rabbit** is a Duck.
- **Rabbit** and Hare are equivalent.
- **Rabbit** and Wolf are mutually exclusive.
- Every **Rabbit** is exactly one of Bugs Bunny OR Peter Rabbit.
- Every **Rabbit** is a White Rabbit or a Black Rabbit.
- Every **Rabbit** is a Cute Thing and a Mammals.
- Every **Rabbit** eats Fresh Vegatable.
- Every **Rabbit** has part Whisker.
- Every **Rabbit** has child(ren) only **Rabbit** or nothing.
- Every **Rabbit** has eye color of only Color or nothing.
- Every **Rabbit** has eye color of Red.
- Every **Rabbit** has leg(s) exactly 4.
- Every **Rabbit** has head at least 1.
- Every **Rabbit** has parent at most 2.

- Supported languages: Rabbit, Rabbit Chinese (Yayan) and Ace

2. Rule Modeling

- Modeling rules as templates.
- Example: Template-based “Domain” inference

```
{{#vardefine:value|{{#ask:
[[:{{FULLPAGENAME}}]]
|?{{{1}}}=
|mainlabel=-
|format=list
|link=none
}} }}

{{#if:{{#var:value}}|[[Category:{{{2}}}]]}}
```

“Rule:Entailment”

Usage: {{Rule:Domain|hasAuthor|Document}}

2. Rule Modeling
Logic Program -> SMW

URL http://tw.rpi.edu/dev/cnl/LP_Test

Examples:

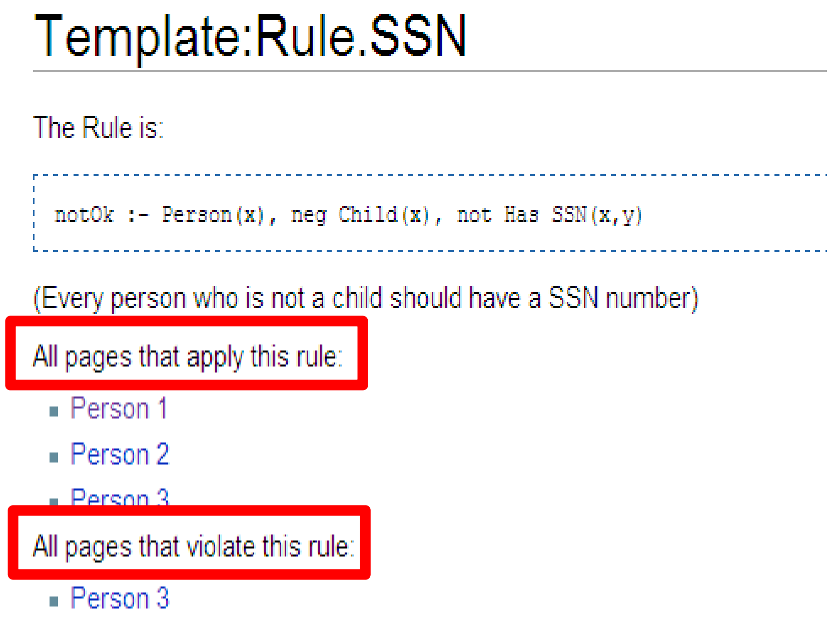
- Person (x) :- Student(x)
  ```
  {{LP Rule|body=1::Student|head=Person}}
  ```
- RightHanded (x) :- Person(x), not LeftHanded(x)
  ```
  {{LP Rule|body=1::Person; 1:not:LeftHanded
  |head=RightHanded}}
  ```
- NotHirable(x) :- Person(x), not hasSSN(x,y)
  ```
  {{LP Rule|body= 1::Person ; 2:not:hasSSN
  |head= NotHirable}}
  ```

3. Integrity Constraint (cont.)

URL: http://tw.rpi.edu/dev/cnl/Integrity_Constraint

An integrity constraint is a special rule with empty head:

- Every person should have a name
  :- Person(x), not Has name(x,y)
- Every person who is not a child should have a SSN number
  :- Person(x), neg Child(x), not Has SSN(x,y)



Template:Rule.SSN

The Rule is:

```
notOk :- Person(x), neg Child(x), not Has SSN(x,y)
```

(Every person who is not a child should have a SSN number)

All pages that apply this rule:

- Person 1
- Person 2
- Person 3

All pages that violate this rule:

- Person 3

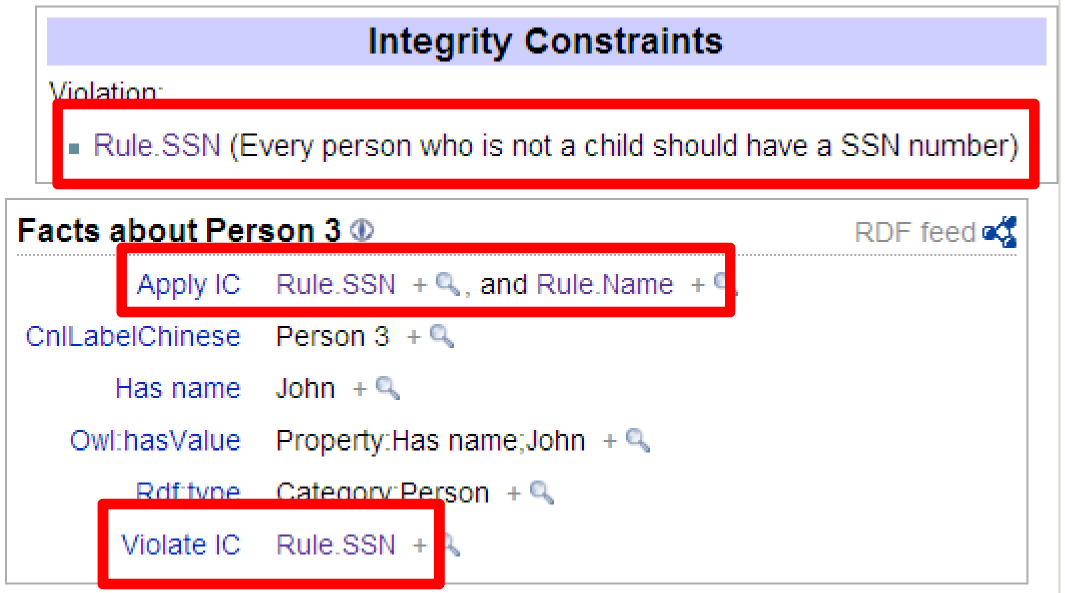
Page “Template:Rule.SSN” does

• (on an instance page) integrity **checking**
• (on the rule) **Retrieval** of all pages that apply/violate the rule

Person 3 contains:

```
{{RuleSet.Person}}
```

Template: RuleSet.Person contains
• Template:Rule.SSN
• Template:Rule.Name



Integrity Constraints

Violation:

- Rule.SSN (Every person who is not a child should have a SSN number)

Facts about Person 3 RDF feed

Apply IC	Rule.SSN + , and Rule.Name +
CnlLabelChinese	Person 3 +
Has name	John +
Owl:hasValue	Property:Has name;John +
Rdf:type	Category:Person +
Violate IC	Rule.SSN +

Conclusions

- By combining SMW with several other extensions, we can do meta-modeling beyond RDF
  - Meta-data can be queried by “ask” query.
  - Inference can be enabled by “if-then” parser functions
  - Simple, Open, and Pay-as-you-go
- Show several useful design patterns
  - OWL
  - Rules: RDF-like and LP-like
  - Integrity Checking