Some thoughts on next steps for Semantic Forms

+ Bug the developer

Yaron Koren
SMWCon Fall 2010
September 19, 2010
Yaron Koren

New York-based MediaWiki developer and consultant

Company:

Wiki Works
MediaWiki consulting

Wiki farm:

Referata
Semantic Forms

My most important extension
im in ur wiki

structuring ur data
Semantic Forms can undoubtedly be improved in many ways.
Possible new extension:

“Semantic Classes”
We can think of a certain data type as a “class”, and the set of wiki pages for that type – category, form, template(s), properties, filters – as the “class pages”.
Semantic Classes extension

Would allow for defining all the information about a "class" - for display, form, and drilldown - within XML contained within a category page. This XML could then be used to generate template, property, form and filter pages. The XML could then also be edited via some kind of helper tool.
Special pages

Probably two:

- Special:EditClassXML
- Special:GenerateClassPages
Sample XML

In a page called “Category:Cities”:

```xml
<SemanticClass>
  <Name>City</Name>
  <Form>City</Form>
  <Field>Population</Field>
  <SemanticProperty>
    <Name>Has population</Name>
    <Type>Number</Type>
  </SemanticProperty>
  <FormInput>
    <InputType>text</InputType>
    <Size>20</Size>
  </FormInput>
  <Filter>
    <Label>Population</Label>
  </Filter>
</SemanticClass>
```
Advantages of Semantic Classes approach

- Single point where all data for a "class" is stored
- Separation of data structure from display (mostly - the one exception is the ordering of fields)
- A class can be edited, not just created, via a helper form - for basic data structures, admins may never have to edit wiki text
SMW queries have a format-driven structure – the format determines the rest of the allowed parameters.

(Visible in Special:Ask)
Perhaps Semantic Forms could similarly have an input-type driven approach?

Currently, display and allowed parameters are driven by a combination of input type, “autocomplete on…”, “values from…”, “list”, etc.

Maybe it should be just input type?
Current approach:

```{{field|author|autocomplete on category=Authors|list}}
```

Possible new approach:

```{{field|author|input type=text with autocomplete|values from category=Authors|list}}
```
Current approach:

```text
{{field|author|values from namespace=Author}}
```

Possible new approach:

```text
{{field|author|input type=dropdown|values from namespace=Author}}
```
Current approach:

```{{field|author|no autocomplete}}
```

Possible new approach:

```{{field|author|input type=text}}
```
(Fields would still have a default input type, based on the type of the property they correspond to)
Advantages of input-type-based approach

• Easier to understand (?)

• Allows for more complete set of options in Special:CreateForm (and maybe Special:EditClassXML as well)
Bug the developer

Feature requests, bug reports, other questions/comments.