SMWCon Fall 2017
October 4 to 6, Rotterdam

Introduction and tutorial to Semantic MediaWiki

Karsten Hoffmeyer, WikiHoster.net
October 4, 2017
SMWCon Fall 2017 – Introduction and tutorial

Overview

• What does semantic mean?

• What is Semantic MediaWiki?

• How does it work?

• How-tos and examples galore!
  – Setting values
  – Querying pages
What does semantic mean?
What does semantic mean?

sem•an•tic (sĭ-măn'tĭk)
adj.: of or relating to meaning,
especially meaning in language.

→ Basically information is given an explicit meaning.
What is Semantic MediaWiki?
First proposed in August 2005 at Wikimania in Frankfurt am Main, Germany
- Paper by Markus Krötzsch, Denny Vrandečić and Max Völkel
What is Semantic MediaWiki?

- First released in September 2005 (v0.1)
  - Extension to the MediaWiki software
- First stable release in December 2007 (v1.0)
- Upcoming stable release in Q4.2017 or Q1. 2018 (v3.0)
  - Now a full-fledged framework extending MediaWiki with many spin-off extensions
What is Semantic MediaWiki?

- Aims at making data more readable for humans and machines
  - Lets you store and query data within the wiki's pages
  - Lets you publish data via the Semantic Web
What is Semantic MediaWiki?

- Can be used to transform a wiki into a powerful information and/or data management system
  - Best used if out of the box software does not provide the required functionality
  - Best used if creating a software with the required functionality is too expensive
How does it work?
Fact:

RouterModule has a population of 3,520,031 people.

A fact consists of three parts:

- Wikipage (subject), e.g. “RouterModule”
- Property (predicate), e.g. “Has population”
- Value (object), e.g. “3,520,031”
SMWCon Fall 2017 - Introduction and tutorial

How does it work?

- These three parts can be split into a relation and a statement:
  - Relation: Berlin has a population.
  - Statement: The population is “3,520,031”.
How does it work?

- A statement consists of a property and its value:
  - Property (predicate), e.g. “Has population”
  - Value (object), e.g. “3,520,031”

→ Adding a statement to a page (subject) is called annotation.
A property must have a datatype:

- Needed to give the property a certain meaning
  e.g. a property of datatype “Telephone number” can only validly hold telephone numbers

  - Page namespace: “Property”
  - Page name: “Has telephone number“
  - Page content: [[Has type::Telephone number]]
    ➔ This is the datatype assignment.

Default is datatype “Page” if nothing was explicitly added.
SMWCon Fall 2017 - Introduction and tutorial

How does it work?

- A property must have a datatype:
  - Needed to be able to also do a formal check of the assigned value
    - e.g. a property value of “abc” annotated to a property of datatype “Telephone number”
      - An error message will be issued.
A property must have a datatype:

- Needed to allow a certain behaviour, i.e. add links, add formatting, etc.

  e.g. a property value of “+49-30-45-08-49-40” added to a property of datatype “Telephone number” will automatically be linked and additionally preceeded with URI prefix “tel:”
• A property must have a datatype:
  
  – There are a lot of datatypes  
    e.g. “Page”, “Text”, “Number”, “Email”, “URL”, “Telephone number”, “Temperature”, “Boolean”, etc.  
    ➔ See “Special:Types” on your wiki  

  – A property can only use one datatype.
SMWCon Fall 2017 - Introduction and tutorial

How does it work?

- A property can be a “special property”:
  - It has a predefined meaning and purpose
    e.g. “Has type” to hold the datatype information
  - It cannot be created manually since it is already provided by Semantic MediaWiki
How does it work?

• A property can be a “special property”:
  
  – It cannot be changed and used otherwise, i.e. must be unique in naming and usage.

  – There are a lot of special properties, e.g. “Has type”, “Allows value”, “Modification date”, “Has improper value for”, “Has subobject”, “Has query”, etc.

  ➔ See “Special:Properties” on your wiki (marked in italics)
A property can be a “special property”

- Some special properties are assigned manually
  e.g. special property “Has type” which assigns the datatype to a user defined property

- Some special properties are assigned automatically
  e.g. “Has improper value for” stores incorrect values assigned to a property
How-tos and examples galore!
Annotate information/data
Setting values (syntax):

- In-text annotation (single value for one property)

[[Property name::Property value]]

The property value is directly shown on the page. Note that this is not working with restricted characters as part of the data value, e.g. "::" as in IPv6 addresses: annotation must use the #set parser function.
How-tos and examples galore! Annotate information/data.

• Setting values (syntax):
  
  – In-text annotation (multiple values for one property)

    ```
    [[Property name::Property value 1]]
    [[Property name::Property value 2]]
    
    The property values are directly shown on the page.
    ```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Annotate information/data.

• Setting values (syntax):

  – In-text annotation (single value for one property)

    [[Property name::Property value| ]]

    The property value is not directly shown on the page. Probably better here: annotation using the \#set parser function.
• Setting values (syntax):
  - \texttt{\#set} annotation (single value for one property)

\texttt{{\#set:Property name=Property value}}

The property value is not directly shown on the page. Note that this must be used if there are restricted characters as part of the data value, e.g. “::” as in IPv6 addresses:

\texttt{{\#set:Has IPv6 address=2001:db8::1428:57ab}}

Note that there is a way to directly show values added via \texttt{\#set}. 
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Annotate information/data.

- Setting values (syntax):

  - \#set annotation (multiple values for one property)

  ```
  {{#set:
  Property name=Property value 1; Property value 2
  |+sep=;
  }}
  ```

  The property values are not directly shown on the page. Uses the \+sep\ parameter.
Setting values (syntax):

- \texttt{#set} annotation (single value for one property)

\begin{verbatim}
{{#set:
  Property name=Property value
  |template=Show property value
}}
\end{verbatim}

The property value is directly shown on the page. Uses the template parameter.

Template “Show property value”:

\begin{verbatim}
{{{value}}}
\end{verbatim}

The annotated data value is directly shown with {{value}}.
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Annotate information/data.

- Setting values (syntax):
  - `#set` annotation (multiple values for one property)
    ```
    {{#set:
       Property name=Property value 1; Property value 2
       |+sep=;
       |template=Show property values
    }}
    ```

    The property value is directly shown on the page. Uses the `template` parameter. Requires ParserFunctions extension (`#if`).

    Template “Show property values”:
    ```
    {{{value}}}{{#if: {{{last-element}}} | |, }}
    ```

    The trailing separator is being avoided with `{{{last-element}}}`.  

Let's work with a real life use case:
Organizing information about books.

- Information to be stored:
  - Title of the book
  - Author of the book
  - Publisher of the book
  - Publishing year of the book
  - Publishing country of the book
  - ISBN of the book
  - Rating for the book
Let's work with a real life use case: Organizing information about books.

- Needed properties and their datatypes:
  - “Has book title” → datatype “Text”
  - “Has book author” → datatype “Page”
  - “Has book publisher” → datatype “Page”
  - “Has book publishing year” → datatype “Date”
  - “Has book publishing country” → datatype “Page”
  - “Has book rating” → datatype “Number”
Let's work with a real life use case:

Organizing information about books.

- Syntax to store and annotate the information:

This page is about the book called [[Has book title::The Picture of Dorian Gray]] written by [[Has book author::Oscar Wilde]]. It was last published by [[Has book publisher::Titan Books]] in the [[Has book publishing country::UK]], [[Has book publishing year::2015]]. The ISBN is [[Has book ISBN::978-1783293650]]. My personal rating for this book is [[Has book rating::1]].

[[Category:Book]]

Most probably one would prefer a tabular instead of a text presentation for this kind of subject but that's entirely up to you. Also you will most probably rather use a form (requires the Page Forms extension) and a template to edit and display the data.
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Annotate information/data.

• Let's work with a real life use case:
  Organizing information about books.

Here you cannot see if the information is stored semantically. Therefore use special page “Browse” via the sidebar to have a look and see what annotations are stored. ➔
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Annotate information/data.

- Let's work with a real life use case:
  Organizing information about books.
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Relevant documentation:
  - Property naming
  - Datatypes
  - In-text annotation
  - #set-annotation
  - Special properties
  - Special page "Browse"
How-tos and examples galore!
Querying pages
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

- Querying pages (syntax):
  - `#show query`
    
    `{#{#show: Page name | ?Property name }}`

* Display only a single property value for a single page.

* This is not about selecting pages which is what Semantic MediaWiki is actually there for.
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  – #ask or inline or embedded or semantic query

• Consists of two parts

  – Query condition

    ➔ Which pages containing information should be selected.

  – Printout statement with display format

    ➔ Which information found on the selected pages should be displayed and how should it be displayed.
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages (syntax):
  
  - \#ask query

• Query condition (selecting pages)

```{#ask:
  [[Category:Book]]
  [[Has book title::+]]
  printout statement
}
```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):
  - #ask query

• Printout statement (displaying information) with default thus basic display formatting

```{#ask:
  query condition
  | ?Has book title
  | ?Has book author
  | ?Has book publishing year
  | ?Has book ISBN
  | ?Has book rating
}
```
How-tos and examples galore! Querying pages.

• Querying pages (syntax):
  - \#ask query

• Query condition + printout statement

```{{#ask:
[[Category:Book]]
[[Has book title::+]]
| ?Has book title
| ?Has book author
| ?Has book publishing year
| ?Has book ISBN
| ?Has book rating
}}
```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

- Querying pages:
  - \#ask query (result)

... but the first column is missing a label!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask query`

```{#ask:
   [[Category:Book]]
   [[Has book title:::+]]
   | ?Has book title
   | ?Has book author
   | ?Has book publishing year
   | ?Has book ISBN
   | ?Has book rating
   | mainlabel=title
}
```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages:
  – #ask query (result)

... but the book title is duplicated!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

- Querying pages (syntax):

```wkt
{{#ask:
[[Category:Book]]
[[Has book title::+]]
|?Has book author
|?Has book publishing year
|?Has book ISBN
|?Has book rating
|mainlabel=title
}}
```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

- Querying pages:
  - `#ask query (result)`

... but the column labels should be changed!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask` query

  ```
  {{#ask:
    [[Category:Book]]
    [[Has book title::+]]
    | ?Has book author=author
    | ?Has book publishing year=year
    | ?Has book rating=rating
    | mainlabel=title
  }}
  ```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages:
  – `#ask query (result)`

… but the column labels should not be linking to the property page!
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask query`
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages:
  – `#ask query (result)`

... but the table should be sorted by year!
• Querying pages (syntax):

  - `#ask query`

```{#ask:
[[Category:Book]]
[[Has book title::+]]
| ?Has book author=author
| ?Has book publishing year=year
| ?Has book rating=rating
| mainlabel=title
| headers=plain
| sort=Has book publishing year
}
```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages:
  - `#ask query (result)`

... but the year should be sorted in descending order!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask query`

```{#ask:
  [[Category:Book]]
  [[Has book title::+]]
  | ?Has book author=author
  | ?Has book publishing year=year
  | ?Has book rating=rating
  | mainlabel=title
  | headers=plain
  | sort=Has book publishing year
  | order=descending
}
```
• Querying pages:
  - `#ask query (result)`

... but the author should also be sorted but in ascending order!
Querying pages (syntax):

- **#ask** query

```{{#ask:
[[Category:Book]]
[[Has book title::+]]
|?Has book author=author
|?Has book publishing year=year
|?Has book rating=rating
|mainlabel=title
|headers=plain
|sort=Has book publishing year,Has book author
|order=descending,ascending
}}```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages:
  – `#ask query (result)`

```
<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>year</th>
<th>ISBN</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Das Glasperlenspiel</td>
<td>Hermann Hesse</td>
<td>2015</td>
<td>978-3518463574</td>
<td>1</td>
</tr>
<tr>
<td>The Picture of Dorian Gray</td>
<td>Oscar Wilde</td>
<td>2015</td>
<td>978-1783293650</td>
<td>1</td>
</tr>
<tr>
<td>Venus in Furs</td>
<td>Leopold von Sacher-Masoch</td>
<td>2008</td>
<td>978-1604501292</td>
<td>3</td>
</tr>
<tr>
<td>The Turn of the Screw</td>
<td>Henry James</td>
<td>1991</td>
<td>978-0486266848</td>
<td>2</td>
</tr>
<tr>
<td>Il Gattopardo</td>
<td>Giuseppe Tomasi di Lampedusa</td>
<td>1990</td>
<td>978-8807810282</td>
<td>2</td>
</tr>
<tr>
<td>Johann Christof</td>
<td>Romain Rolland</td>
<td>1977</td>
<td>978-3423020329</td>
<td>1</td>
</tr>
<tr>
<td>Siddhartha</td>
<td>Hermann Hesse</td>
<td>1974</td>
<td>978-3518366820</td>
<td>1</td>
</tr>
</tbody>
</table>
```

... but the values for year and rating should be centered!
• Querying pages (syntax):

- `##ask query`

```{}
{{#ask:
[[Category:Book]]
[[Has book title::+]]
|?Has book author=author
|?Has book publishing year=year|+align=center
|?Has book rating=rating|+align=center
|mainlabel=title
|headers=plain
|sort=Has book publishing year, Has book author
|order=descending,ascending
}}
```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages:
  - `#ask query (result)`

... but I would like to have a short introductory sentence!
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages (syntax):

- #ask query

{{#ask:
[[Category:Book]]
[[Has book title::+]]
| ?Has book author=author
| ?Has book publishing year=year|+align=center
| ?Has book rating=rating|+align=center
| intro=<b>This table shows the books I have stored here:</b>
| mainlabel=title
| headers=plain
| sort=Has book publishing year,Has book author
| order=descending,ascending
}}
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

● Querying pages:
  - `#ask query (result)`

... but I would like show only the first three results!
Querying pages (syntax):

- `#ask query`

```
{{#ask:
  [[Category:Book]]
  [[Has book title::+]]
  | ?Has book author=author
  | ?Has book publishing year=year | +align=center
  | ?Has book rating=rating | +align=center
  intro=<b>This table shows the books I have stored here:</b>
  mainlabel=title
  headers=plain
  sort=Has book publishing year,Has book author
  order=descending,ascending
  limit=3
}}
```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages:
  - `#ask query (result)`

<table>
<thead>
<tr>
<th>title</th>
<th>author</th>
<th>year</th>
<th>ISBN</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Das Glasperlenspiel</td>
<td>Hermann Hesse</td>
<td>2015</td>
<td>978-3518463574</td>
<td>1</td>
</tr>
<tr>
<td>The Picture of Dorian Gray</td>
<td>Oscar Wilde</td>
<td>2015</td>
<td>978-1783293650</td>
<td>1</td>
</tr>
<tr>
<td>Venus in Furs</td>
<td>Leopold von Sacher-Masoch</td>
<td>2008</td>
<td>978-1604501292</td>
<td>3</td>
</tr>
</tbody>
</table>

... in English and a bit more specific please!
Querying pages (syntax):

- \texttt{#ask query}

\begin{verbatim}
{{#ask:
[[Category:Book]]
[[Has book title:::+]]
| ?Has book author=author
| ?Has book publishing year=year|+align=center
| ?Has book rating=rating|+align=center
| intro=<b>This table shows the books I have stored here:</b>
| mainlabel=title
| searchlabel=... more books
| headers=plain
| sort=Has book publishing year,Has book author
| order=descending,ascending
| limit=3
}}
\end{verbatim}
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

- Querying pages:
  - #ask query (result)

Shouldn’t the columns “title” and “author” be wider and of the same width?
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):
  - `#ask query`

```wiktart
{{#ask:
  [[Category:Book]]
  | ?=title|+width=20em
  | ?Has book author=author|+width=20em
  | ?Has book publishing year=year|+align=center
  | ?Has book rating=rating|+align=center
  | intro=This table shows the books I have stored here:
  | mainlabel=--
  | searchlabel=... more books
  | headers=plain
  | sort=Has book publishing year,Has book author
  | order=descending,ascending
  | limit=3
}}
```

Note: SMW 3.0.0 or later only!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

- Querying pages:
  - #ask query (result)

Isn’t there a cool new predefined class (view) around?
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask` query

```{#ask:
[[Category:Book]]
[[Has book title:::+]]
|?=title|+width=40%
|?Has book author=author|+width=40%
|?Has book publishing year=year|+align=center|+width=5%
|?Has book rating=rating|+align=center|+width=5%
intro=<b>This table shows the books I have stored here:</b>
|mainlabel=-
|searchlabel=... more books
|headers=plain
|sort=Has book publishing year,Has book author
|order=descending,ascending
|limit=3
|class=datatable <!-- comes with width=100% -->}
```

Note: SMW 3.0.0 or later only!
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages:
  - `#ask` query (result)

Can I also provide a link to download this information?
How-tos and examples galore! Querying pages.

• Querying pages (syntax):

  - `#ask` query

  ```
  {{#ask:
    [[Category:Book]]
    [[Has book title:::+]]
    |?=title|+width=20em
...
  |intro=<b>This table shows the books I have stored here:</b>
  |mainlabel=--
  |searchlabel=... more books
  |headers=plain
  |sort=Has book publishing year, Has book author
  |order=descending, ascending
  |limit=3
  |class=datatable
  |outro=→ {{#ask: ... }} <!-- Add #ask query no. 2 here -->
  }}
  ```
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

• Querying pages (syntax):
  – `#ask query no. 2`

```{#ask:
  [[Category:Book]]
  [[Has book title::+]]
  |?Has book author=author
  |?Has book publishing year=year
  |?Has book rating=rating
  |mainlabel=title
  |searchlabel=CSV-download of stored books
  |sort=Has book publishing year,Has book author
  |order=descending,ascending
  |format=csv
}}```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

• Querying pages:
  – \#ask query (result)

Theoretically: What happens if there are no results?
SMWCon Fall 2017 - Introduction and tutorial

How-tos and examples galore! Querying pages.

- Querying pages (syntax):
  - `#ask query`
    ```
    {{#ask:
      [[Category:Book]]
      [[Has book title::+]]
      |?=title|+width=20em
      ...
      |intro=<b>This table shows the books I have stored here:</b>
      |mainlabel=-
      |searchlabel=... more books
      |default=<b>No books have been stored yet.</b>
      |headers=plain
      |sort=Has book publishing year,Has book author
      |order=descending,ascending
      |limit=3
      |class=datatable
      |outro=  {{#ask: ... }}
    }}
    ```
SMWCon Fall 2017 - Introduction and tutorial
How-tos and examples galore! Querying pages.

- Relevant documentation:
  - Selecting pages
  - Displaying information
  - Inline queries
Thank you!