DBpedia Mapping Language

Authors: Anja Jentzsch, Christopher Sahnwaldt, Robert Isele, Christian Bizer

Last revision: 3/8/10

Abstract

This document specifies the DBpedia Ontology Schema Definition Language, the DBpedia Templates to Ontology Mapping Language and the Tables to Ontology Mapping Language. Using these languages, the ontology and the mappings are defined and maintained in a MediaWiki instance.

Table of Contents

1. Introduction

The type of Wikipedia content that is most valuable for the DBpedia extraction are infoboxes and tables. Infoboxes display an article's most relevant facts as a table of attribute-value pairs on the top right-hand side of the Wikipedia page. By mapping Wikipedia templates and tables to the DBpedia ontology, a basis is established to improve the quality of the infobox extraction and to permit table extraction. In the following, we specify a suitable mapping language based on requirements deducted from existing Wikipedia infobox and table usage and outline the related mapping and extraction process.

This document is structured as follows: In section 2, we give an overview of the components of the mapping and extraction framework, whereas the extraction process itself is described in section 3. Section 4 collects requirements for the mapping framework. Section 5 specifies the DBpedia Ontology Schema Definition Language, the DBpedia Templates to Ontology Mapping Language and the Tables to Ontology Mapping Language. Section 6 demonstrates how the requirements are fulfilled by the designed languages. Section 7 contains an example of a complete template and a table to Ontology mapping.

2. Component Overview

The components of the DBpedia mapping and extraction framework are:

- **Ontology Schema Definition.** Covers the definition of classes in the DBpedia namespace as well as their properties. The definition can be edited within a MediaWiki instance.
- **Wikipedia Templates to DBpedia Ontology Mappings.** Allows users to map Wikipedia templates to DBpedia ontology classes, and Wikipedia template properties to DBpedia ontology properties. The mappings can be edited within a MediaWiki instance.
- **Wikipedia Tables to DBpedia Ontology Mappings.** Allows users to map Wikipedia tables to the DBpedia ontology. The mappings can be edited within a MediaWiki instance.
- **Schema Parser.** Parses and validates the Ontology Schema Definition and converts it into OWL.
- **Mapping Definition Parser.** Parses and validates the Template/Table To Ontology mappings.
- **Extraction Framework and Parsers.** Extracts Wikipedia templates and tables from Wikipedia page sources, parses them and translates their content into DBpedia instance data.

3. Mapping and Extraction Process

The definition of the ontology schema and the template/table to ontology mappings take place on pages of a MediaWiki installation. This allows the ontology and the mappings to be maintained in Wikipedia itself at a later stage.

We will therefore create MediaWiki template pages that define DBpedia ontology classes and properties as well as template/table to ontology mappings. The DBpedia extraction framework will parse and validate these templates.

The mapping and extraction process takes place as follows:

1. Definition of the DBpedia Ontology Schema
2. Definition of Wikipedia Template to DBpedia Ontology Mappings
3. Definition of Wikipedia Table to DBpedia Ontology Mappings
4. Extraction of the ontology and the mappings from the MediaWiki instance.
5. Extraction of structured data from the Wikipedia pages according to the ontology using the template and table mappings.

4. Requirements

This section deducts requirements from observed Wikipedia template and table usage and provides examples in wiki page source.

4.1. Wikipedia Templates to DBpedia Ontology Mapping

R1. In most cases, one Wikipedia template can be mapped to one ontology class.

Example: The template `Infobox Automobile` shall be mapped to the ontology class `Automobile`.

```
{{ Infobox Automobile
| name = Volkswagen Golf
...
}}
```

R2. If a Wikipedia page contains multiple infobox templates, all relevant templates should be processed.

Example: The templates `Infobox Automobile` and `Infobox Automobile engine` occur on one page.

```
{{ Infobox Automobile
| name = Volkswagen Golf
...
}}

{{ Infobox Automobile engine
| name = Volkswagen air cooled engine
...
}}
```

4.2. Wikipedia Template Properties to DBpedia Ontology Properties Mapping

R3. In most cases, each template property of a template can be mapped to a corresponding property of the mapped DBpedia ontology class.

Example: All template properties of the template `Infobox Automobile` are mapped to corresponding properties of the ontology class `Automobile`.

R4. Template property values have certain types and units.

R1.1. For template properties with values pointing to other resources, it must be possible to specify the ontology class range.

Example: The template property `occupation` on the template `Infobox Person` contains `Organisations`.

```
{{ Infobox Person
| name = Dieter Zetsche
| occupation = Chairman of [[Daimler AG]]
...
}}
```

R1.2. For template properties with datatype values, it must be possible to specify datatypes such as integer, date and duration.

Example: The template property `foundation` on the template `Infobox company` should be mapped to the ontology property `foundationDate` on `Organisation`.

```
{{ Infobox company
| company_name     = Volkswagen
| foundation       = May 28, 1937
...
}}
```

R1.3. For template properties that contain numeric values, it must be possible to specify a unit, e.g. meter, square kilometer, Euro.

Example: The template `Infobox company` contains the property `net_income` which contains currency values.

```
{{ Infobox company
| name = AUDI AG
| net_income = {{profit}} [[euro|€]] 2.915 [[1000000000
(number)|billion]] (2007)
...
}}
```

R1.4. Template property dimension (e.g. distance, area, point in time) must equal the dimension of the mapped ontology property. If a template property contains a unit it has to be valid / defined for the given dimension. If the template property value contains a unit which is not in the target dimension, the value has to be ignored. If there is no unit defined in a template property value, the unit of the Wikipedia template definition should be taken as the default unit.

R1.5. If the template property value contains a unit which is in the target dimension, but the target unit differs, the value has to be converted.

Example: A Wikipedia page using the template `Infobox Mountain` defines the `elevation` property as `'8.4km'`. The target unit of the mapped ontology property is meter.

R5. A template property value must be split and mapped to several ontology properties. The target ontology properties can have different domains. The value can contain a n-tuple (several values of different types, e.g. birth date und birth place) or a list of values or n-tuples (e.g. list of cities, list of value pairs).

Example: The template property `born` on the template `Infobox Person` contains the birth place and birth date of a person.

```
{{ Infobox Person
| name = Dieter Zetsche
| born = {{birth date and age|1953|5|5}}, [[Istanbul]], [[Turkey]]
...
}}
```

R1.6. Template properties can contain (e.g. time) intervals.

Example: The template property `production` on the template `Infobox Automobile` contains two date values, which shall be mapped to the ontology properties `productionStartDate` and `productionEndDate`.

```
{{ Infobox Automobile
| name       = Volkswagen Beetle
| production = 1938-2003
...
}}
```

R1.7. The original correspondence of template property values should be preserved.

Example: The template property `engine` on the template `Infobox Automobile` can contain a list of engine numbers and the corresponding power and torque.

```
{{ Infobox Automobile
| name     = Volkswagen Passat
| engine   = 1.6L 35 kW 135 Nm <br>
             1.8L 75 PS 140 Nm
}}
```

Example: The template property `class` on the template `Infobox Automobile` can contain a list of classes and the corresponding years the automobile was of this class.

```
{{ Infobox Automobile
| name     = Volkswagen Passat
| engine   = 1.6L 35 kW 135 Nm <br>
             1.8L 75 PS 140 Nm
}}
```

R6. Several template properties must be mapped to one or more ontology properties, preserving their relation.

Example: Geo coordinates are often distributed to four or six properties: for latitude and longitude and the corresponding degree, minute, second, south/north, east/west. See template `Infobox Settlement`.

```
{{ Infobox settlement
|latd = 39  |latm = 52  |latNS = N
|longd = 32 |longm = 52 |longEW = E
...
}}
```

R7. The template property value contains a list of values of the same type which have to be extracted as multiple triples for the same ontology property. It should be possible to easily define that a template property contains list values and the range of these values. The conventions to split a list are diverse, e.g. commas, <br/> or wiki tables. Ideally one can configure how a list is split.

Example: The template property `engine` on the template `Infobox Automobile` can contain a list of engine numbers and the corresponding power and torque.

```
{{ Infobox Automobile
| name     = Volkswagen Passat
| engine   = 1.6L 35 kW 135 Nm <br>
            1.8L 75 PS 140 Nm
}}
```

R8. Sometimes there is a great amount of equivalent template properties with similar names. To cover these cases, properties should be addressable using name patterns.

Example: Template properties defined on the template `Infobox Settlement`: `p1`, `p2` to `p50`.

4.3. Wikipedia Tables to DBpedia Ontology Mapping

R9. All rows of a table shall be mapped to instances of an ontology class, all of its columns shall be mapped to ontology properties.

Example: The rows of tables with the columns `Engine Name`, `Type`, `Engine code`, and `Top speed` are mapped to instances of the ontology class `Automobile Engine`, while the columns are mapped to the corresponding ontology properties. If an infobox that is mapped to the ontology class `Automobile` is in the same section as the table, the instances generated for the table rows are connected to the instance generated from the infobox by the ontology property `MeanOfTransportation/engine`.

R10. Table column headers can slightly differ in their naming but still have the same meaning and mapping.

Example: The rows of tables with the columns `Engine Name`, `Type`, `Engine code`, and `Top speed` are mapped to instances of the ontology class `Automobile Engine`, but the column `Top Speed kmh` would have the same meaning as the column `Top Speed`.

5. Language Specification

In the following, we specify the Ontology Schema Definition Language (5.1), the Template to DBpedia Ontology Mapping Language (5.2) and Tables to Ontology Mapping Language (5.3).

5.1. Ontology Schema Definition Language

The language consists of the following Wikipedia templates to cover the ontology schema definition:

- `DBpediaClass`
- `DBpediaDatatypeProperty`
- `DBpediaObjectProperty`

The `DBpediaClass` template offers the following template parameters:

- `rdfs:label`
- `rdfs:subClassOf`
- `owl:equivalentClass`

The `DBpediaDatatypeProperty` template offers the following template parameters:

- `rdfs:label`
- `rdfs:domain`
- `rdfs:range`
- `owl:equivalentProperty`

The `DBpediaObjectProperty` template offers the following template parameters:

- `rdfs:label`
- `rdfs:domain`
- `rdfs:range`
- `owl:equivalentProperty`

Examples

The definition of the DBpedia ontology class `Person`:

```
{{ DBpediaClass
 | rdfs:label    = Person
}}
```

The definition of the DBpedia ontology property `height`:

```
{{ DBpediaDatatypeProperty
 | rdfs:label  = height
 | rdfs:domain = Person
 | rdfs:range  = centimeter
}}
```

5.2. Template to Ontology Mapping Language

When mapping a Wikipedia template to an ontology class and mapping template properties to ontology properties for this template, users will have to edit the corresponding template documentation page in MediaWiki.

We will define the following Wikipedia templates to cover the template to ontology schema mapping:

- `DBpediaTemplateMapping`

  Mapping from Wikipedia templates to ontology classes.

- `DBpediaPropertyMapping`

  Mapping from Wikipedia template properties to ontology properties.

- `DBpediaIntermediateNodeMapping`

  For extracting multiple values from a single property it is necessary to introduce an intermediate node. The `DBpediaIntermediateNodeMapping` allows to express mappings from Wikipedia template properties to ontology properties on an additional node and to connect the additional node to the mapped instance.

- `DBpediaCustomMapping`

  To cover specific, more complex mapping cases, the DBpedia extraction framework can be extended with custom parsers which have to implement a specific PHP interface. These parsers invoked using custom mappings.

DBpedia Template Mapping

The `DBpediaTemplateMapping` template offers the following template parameters:

- `mapToClass`

  Templates are mapped to ontology classes. The template parameter `mapToClass` allows one DBpedia ontology class as a value.

- `correspondingClass, correspondingProperty`

  In the case that different templates are used on the same page (for instance Automobile and Automobile Generation), the instance resulting from the second grade template (Automobile Generation) can be connected to the instance of the first grade template (Automobile) using a corresponding property. Thus, if an instance of type `correspondingClass` is found on the same page, it will be connected to the instances of the mapped template by `correspondingProperty`.

- `mappings`

  Mappings map template properties to ontology properties, they have to be defined by using `DBpediaPropertyMapping` or `DBpediaIntermediateNodeMapping`. Custom, user-defined, mappings like the `DBpediaGeocoordinatesMapping` can also be defined.

DBpedia Property Mapping

The `DBpediaPropertyMapping` template offers the following template parameters:

- `ontologyProperty`

  A template property to ontology property mapping should list one ontology property.

- `templateProperty`

  A template property to ontology property mapping should list one template property which is to be mapped.

- `unit`

  If a template property containing a numerical value and a unit is mapped, the unit has to be defined.

  If a template property has no default unit defined, e.g. its values can contain different units of the same dimension, the dimension has to be defined for usability as well as validation reasons. Possible dimensions are `Length` or `Mass`.

DBpedia Intermediate Node Mapping

The `DBpediaIntermediateNodeMapping` template offers the following template parameters:

- `nodeClass, ontologyProperty`

  Creates an additional node of the type `nodeClass`, which will be connected to the instance extracted from template by the property provided by `ontologyProperty`.

- `mappings`

  Mappings map template properties to ontology properties, they have to be defined by using `DBpediaPropertyMapping`, `DBpediaIntermediateNodeMapping`, or a `DBpediaCustomMapping`.

Custom Mappings
For specific tasks, such as extracting durations or calculating a geo-location-ID based on multiple properties, we allow the DBpedia extraction framework to be extended with custom value parsers and allow the definition of DBpedia custom mapping templates. The name of a custom mapping template has to be equal to the name of the corresponding DBpedia parser class.

As examples of custom mapping, we define the `DBpediaDateIntervalMapping` and the `DBpediaGeocoordinatesMapping`.

The `DBpediaDateIntervalMapping` template provides an exact mapping from start and end dates of a template property value to ontology properties. It offers the following template parameters:

- `templateProperty`
- `startDateOntologyProperty`
- `endDateOntologyProperty`

The `DBpediaGeocoordinatesMapping` template offers the following template parameters:

- `latitudeDirection`
- `latitudeDegrees`
- `latitudeMinutes`
- `latitudeSeconds`
- `longitudeDirection`
- `longitudeDegrees`
- `longitudeMinutes`
- `longitudeSeconds`

5.3. Table to Ontology Mapping Language

Tables in Wikipedia are identified by their table signature. The table signature consists of all column headers of a table. Table mappings are only applied to tables that contain all of the mapped columns. Each table row will be mapped to one instance of the defined type (ontology class).

Tables can occur on pages without any infobox templates as well as on pages with one or more template infoboxes.

If a table is located on a page without a infobox template, the type of this page might be derived from this table.

If a table is located on a page containing one or more infobox templates, one can derive its correspondence to the infobox template from the section the table is located in.

We will define the following Wikipedia template to cover the table to ontology schema mapping:

- `DBpediaTableMapping`

Table mappings can be defined on ontology class definition pages or on separate pages.

DBpedia Table Mapping

The `DBpediaTableMapping` template offers the following template parameters:

- `mapToClass`

  Table rows are mapped to ontology classes. The template parameter `mapToClass` allows one DBpedia ontology class as a value.

- `correspondingClass, correspondingProperty`

  If an instance of type `correspondingClass` is found in the same section as the table, it will be connected to the instances extracted from the table rows by `correspondingProperty`.

- `keywords`

  Keywords define mandatory table column headers or parts of them. Keywords can be separated by semicolons meaning they have to occur in different column headers. Alternatives of keywords can be separated by colons.

- `header`

  Lists all mapped table column headers, separated by semicolons. Each column definition can contain alternative definitions, separated by colons. If the column is defined by several keywords, the keywords have to be separated by `&` in the order of their appearance.

- `mappings`

  Mappings map table columns to ontology properties, they have to be defined by using the same templates as in `mappings` on `DBpediaTemplateMapping`. A separate mapping for each alternative column header definition has to be specified.

6. Examples by Requirement

This section shows how the mapping language is applied to fulfill the requirements defined in section 4.

6.1. Wikipedia Templates to DBpedia Ontology Mapping

R1. In most cases, one Wikipedia template can be mapped to one ontology class.

Example: The template `Infobox Automobile` shall be mapped to the ontology class `Automobile`.

```
{{ Infobox Automobile
| name = Volkswagen Golf
...
}}
```

Implementation: The template `Infobox Automobile` is mapped to the ontology class `Automobile` by using the `DBpediaTemplateMapping` template and its template parameter `mapToClass`.

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
}}
```

R2. If a Wikipedia page contains multiple infobox templates, all relevant templates should be processed.

Example: The templates `Infobox Automobile` and `Infobox Automobile engine` occur on one page.

```
{{ Infobox Automobile
| name = Volkswagen Golf
...
}}

{{ Infobox Automobile engine
| name = Volkswagen air cooled engine
...
}}
```

Implementation: The first template infobox on a page defines the type of this page, while further infobox templates will be extracted as instances of the corresponding types and own URIs.

The template `Infobox Automobile` shall be mapped to the ontology class `Automobile`, while the template `Infobox Automobile engine` shall be mapped to the ontology class `AutomobileEngine`. The correspondence of the infobox template instances can be preserved by defining a `correspondingClass` and `correspondingProperty` on the `Automobile` instance pointing to the `AutomobileEngine` instance.

The mapping for the template `Infobox Automobile` to the ontology class `Automobile` can be defined as follows:

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
}}
```

The mapping for the template `Infobox Automobile engine` to the ontology class `AutomobileEngine` can be defined as follows:

```
{{ DBpediaTemplateMapping
| mapToClass            = AutomobileEngine
| correspondingClass    = Automobile
| correspondingProperty = Automobile/engine
}}
```

6.2. Wikipedia Template Properties to DBpedia Ontology Properties Mapping

R3. In most cases, each template property of a template can be mapped to a corresponding property of the mapped DBpedia ontology class.

Example: All template properties of the template `Infobox Automobile` are mapped to corresponding properties of the ontology class `Automobile`.

Implementation: The template property `platform` of the template `Infobox Automobile` is mapped to the ontology property `platform` on the ontology class `Automobile`.

The ontology property `platform` on the ontology class `Automobile` is defined as follows:

```
{{ DBpediaObjectProperty
 | rdfs:label  = platform
 | rdfs:domain = Automobile
 | rdfs:range  = AutomobilePlatform
}}
```

The template property `platform` is mapped to the defined ontology property `platform` as follows:

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
| mappings =

  {{ DBpediaPropertyMapping | ontologyProperty = Automobile/platform |
  templateProperty = platform }}

}}
```

R4. Template property values have certain types and units.

R4.1. For template properties with values pointing to other resources, it must be possible to specify the ontology class range.

Example: The template property `occupation` on the template `Infobox Person` contains `Organisations`.

```
{{ Infobox Person
| name = Dieter Zetsche
| occupation = Chairman of [[Daimler AG]]
...
}}
```

Implementation: The template property `occupation` on the template `Infobox Person` should be mapped to the ontology property `occupation` on `Person` which has the range `Organisation`.

The ontology property `occupation` on the ontology class `Person` is defined as follows:

```
{{ DBpediaObjectProperty
 | rdfs:label  = occupation
 | rdfs:domain = Person
 | rdfs:range  = Organisation
}}
```

The template property `occupation` is mapped to the ontology property `occupation` on the ontology class `Person` as follows:

```
{{ DBpediaTemplateMapping
| mapToClass = Person
| mappings =

  {{ DBpediaPropertyMapping | ontologyProperty = Person/occupation |
  templateProperty = occupation }}

}}
```

R4.2. For template properties with datatype values, it must be possible to specify datatypes such as integer, date and duration.

Example: The template property `foundation` on the template `Infobox company` should be mapped to the ontology property `foundationDate` on `Organisation`.

```
{{ Infobox company
|company_name     = Volkswagen
|foundation       = May 28, 1937
...
}}
```

Implementation: The template property `foundation` on the template `Infobox company` should be mapped to the ontology property `foundationDate` on `Organisation` which has the range `xsd:date`. The template itself should be mapped to the ontology class `Company` which is a subclass of `Organisation`.

The ontology class `Company` is defined as follows:

```
{{ DBpediaClass
 | rdfs:label      = Company
 | rdfs:subClassOf = Organisation
}}
```

The ontology property `foundationDate` on the ontology class `Organisation` is defined as follows:

```
{{ DBpediaDatatypeProperty
 | rdfs:label  = foundationDate
 | rdfs:domain = Organisation
 | rdfs:range  = xsd:date
 | rds:label   = foundation date
}}
```

The template property `foundation` is mapped to the ontology property `foundationDate` on the ontology class `Organisation` as follows:

```
{{ DBpediaTemplateMapping
| mapToClass = Company
| mappings =
  {{ DBpediaPropertyMapping | ontologyProperty =
  Organisation/foundationDate | templateProperty = foundation }}

}}
```

R4.3. For template properties that contain numeric values, it must be possible to specify a dimension, e.g. length, area, money. It must be possible to specify an unit, e.g. meter, square kilometer, Euro.

Example: The template `Infobox company` contains the property `net_income` which contains currency values.

```
{{ Infobox company
|name = AUDI AG
|net_income = {{profit}} [[euro|€]] 2.915 [[1000000000
(number)|billion]] (2007)
...
}}
```

Implementation: The template property `net_income` on the template `Infobox company` is mapped to the ontology property `profit` on `Organisation` which has the range `euro`, which is the dimension `Currency` and the default unit Euro. The template itself is mapped to the ontology class `Company` which is a subclass of `Organisation`.

```
{{ DBpediaTemplateMapping
| mapToClass = Company
| mappings =

  {{ DBpediaPropertyMapping | ontologyProperty = Organisation/profit |
  templateProperty = net_income | unit = euro }}

}}
```

The ontology property `profit` on the ontology class `Organisation` is defined as follows:

```
{{ DBpediaDatatypeProperty
 | rdfs:label  = profit
 | rdfs:domain = Organisation
 | rdfs:range  = euro
}}
```

R4.4. Template property dimension must equal the dimension of the mapped ontology property. If a template property contains a unit it has to be valid / defined for the given dimension. If the template property value contains a unit which is not in the target dimension, the value has to be ignored. If there is no unit defined in a template property value, the unit of the Wikipedia template definition should be taken as the default unit.

Implementation: The DBpedia parser for validating the mapping to the ontology schema has to check whether the template mapping defines the same dimension as the ontology schema definition does.

If the template property mapping defines a unit, the DBpedia parser for validating the mapping to the ontology schema has to check whether the dimension of this unit equals the dimension of the ontology schema definition.

If the template property value contains a unit, the DBpedia parser for parsing unit values in templates has to check whether the dimension of this unit equals the dimension of the ontology schema definition. If this is not the case, the value has to be ignored. If the template property value contains no unit, the DBpedia parser for parsing unit values in templates should use the unit of the template property mapping.

R4.5. If the template property value contains a unit which is in the target dimension, but the target unit differs, the value has to be converted.

Example: A Wikipedia page using the template `Infobox Mountain` defines the `elevation` property as `'8.4km'`. The target unit of the mapped ontology property is meter.

Implementation: The DBpedia parser for parsing unit values in templates has to convert unit values of the same dimension as defined in the ontology schema definition to the target unit of the mapped ontology property.

R5. A template property value is split and mapped to several ontology properties. The target ontology properties can have different domains. The value can contain a n-tuple (several values of different types, e.g. birth date und birth place) or a list of values or n-tuples (e.g. list of cities, list of value pairs)

Example: The template property `born` on the template `Infobox Person` contains the birth place and birth date of a person.

```
{{ Infobox Person
| name = Dieter Zetsche
| born = {{birth date and age|1953|5|5}}, [[Istanbul]], [[Turkey]]
...
}}
```

Implementation: The template property `born` on the template `Infobox Person` is mapped to the ontology properties `birthDate` and `birthPlace`. The DBpedia parsers will decide which part of the template property value will be extracted for which ontology property depending on the ontology property definition, in this case the defined ranges.

```
{{ DBpediaTemplateMapping
| mapToClass = Person
| mappings =
  {{ DBpediaPropertyMapping | ontologyProperty = Person/birthDate |
  templateProperty = born }}
  {{ DBpediaPropertyMapping | ontologyProperty = Person/birthPlace |
  templateProperty = born }}
}}
```

R5.1. Template properties can contain (e.g. time) intervals.

Example: The template property `production` on the template `Infobox Automobile` contains two date values, which shall be mapped to the ontology properties `productionStartDate` and `productionEndDate`.

```
{{ Infobox Automobile
| name       = Volkswagen Beetle
| production = 1938-2003
...
}}
```

Implementation: The template property `production` on the template `Infobox Automobile` is mapped to the ontology properties `productionStartDate` and `productionEndDate` by using the `DBpediaDateIntervalMapping`. The template is mapped to the ontology class `Automobile`.

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
| mappings =

  {{ DBpediaDateIntervalMapping | templateProperty = production |
  startDateOntologyProperty = Automobile/productionStartDate |
  endDateOntologyProperty = Automobile/productionEndDate }}

}}
```

The ontology properties `productionStartDate` and `productionEndDate` are defined as follows:

```
{{ DBpediaDatatypeProperty
 | rdfs:label  = production start date
 | rdfs:domain = Automobile
 | rdfs:range  = xsd:date
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = production end date
 | rdfs:domain = Automobile
 | rdfs:range  = xsd:date
}}
```

R5.2. The original correspondence of template property values should be preserved.

Example: The template property `engine` on the template `Infobox Automobile` can contain a list of engine numbers and the corresponding power and torque.

```
{{ Infobox Automobile
| name     = Volkswagen Passat
| engine   = 1.6L 35 kW 135 Nm <br>
             1.8L 75 PS 140 Nm
}}
```

Implementation: The template property `engine` on the template `Infobox Automobile` is mapped to an intermediate node of the type `AutomobileEngine` via the ontology property `engine`. For this purpose we use a `DBpediaIntermediateNodeMapping` which defines this connection property (`engine`) and the mappings from values occuring in the template values of the template property `engine` to ontology properties on `AutomobileEngine`.

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
| mappings =

  {{ DBpediaIntermediateNodeMapping | nodeClass = AutomobileEngine |
  correspondingProperty = Automobile/engine | mappings =

    {{ DBpediaPropertyMapping | ontologyProperty =
    AutomobileEngine/number | templateProperty = engine }}

    {{ DBpediaPropertyMapping | ontologyProperty =
    AutomobileEngine/power | templateProperty = engine | unit = Power }}

    {{ DBpediaPropertyMapping | ontologyProperty =
    AutomobileEngine/torque | templateProperty = engine | unit =
    newtonMeter }}

  }}

}}
```

R6. Several template properties are mapped to one or more ontology properties, preserving their relation.

Example: Geo coordinates are often distributed to four or six properties: for latitude and longitude and the corresponding degree, minute, second, south/north, east/west. See template `Infobox Settlement`.

```
{{ Infobox settlement
|latd = 39  |latm = 52  |latNS = N
|longd = 32 |longm = 52 |longEW = E
...
}}
```

Implementation: The template `Settlement` contains the template properties `lat_deg`, `lat_min`, `lat_sec`, `lat_dir`, `long_deg`, `long_min`, `long_sec`, `long_dir` from which the DBpedia parser named `DBpediaGeocoordinatesMapping` has to generate correct latitude and longitude values and write them as GeoNames triples. Here the template `DBpediaGeocoordinatesMapping` is used.

```
{{ DBpediaTemplateMapping
| mapToClass = Settlement
| mappings =

  {{ DBpediaGeocoordinatesMapping | latitudeDirection = lat_dir |
  latitudeDegrees = lat_deg | latitudeMinutes = lat_min |
  latitudeSeconds = lat_sec | longitudeDirection = long_dir |
  longitudeDegrees = long_deg | longitudeMinutes = long_min |
  longitudeSeconds = long_sec }}

}}
```

R7. The template property value contains a list of values of the same type which have to be extracted as multiple triples for the same ontology property. It should be possible to easily define that a template property contains list values and the range of these values. The conventions to split a list are diverse, e.g. commas, <br/> or wiki tables. Ideally one can configure how a list is split.

Example: The template property `engine` on the template `Infobox Automobile` can contain a list of engine numbers and the corresponding power and torque.

```
{{ Infobox Automobile
| name     = Volkswagen Passat
| engine   = 1.6L 35 kW 135 Nm <br>
             1.8L 75 PS 140 Nm
}}
```

Implementation: The implementation for this requirement follows the implementation of R5.2. Furthermore the DBpedia parsers shall determine lists where several of the mapped n-tuples occur in template property values and there are obvious list dividers like <br> or line breaks.

R8. Sometimes there is a great amount of equivalent template properties with similar names. For these, patterns would be useful.

Example: Template properties defined on the template `Infobox Settlement`: `p1`, `p2` to `p50`. These template properties shall be mapped to the ontology property `district` on the ontology class `Settlement`.

Implementation: We will allow regular expressions for the naming of template properties.

```
{{ DBpediaTemplateMapping
| mapToClass = Settlement
| mappings =

  {{ DBpediaPropertyMapping | ontologyProperty = Settlement/district |
  templateProperty = p[0-9]* }}

}}
```

6.3. Wikipedia Tables to DBpedia Ontology Mapping

R9. All rows of a table shall be mapped to instances of an ontology class, all of its columns shall be mapped to ontology properties.

Example: The rows of tables with the columns `Engine Name`, `Type`, `Engine code`, and `Top speed` are mapped to instances of the ontology class `Automobile Engine`, while the columns are mapped to the corresponding ontology properties. If an infobox that is mapped to the ontology class `Automobile` is in the same section as the table, the instances generated for the table rows are connected to the instance generated from the infobox by the ontology property `MeanOfTransportation/engine`.

Implementation:

```
{{ DBpediaTableMapping
 | mapToClass            = AutomobileEngine
 | correspondingClass    = Automobile
 | correspondingProperty = MeanOfTransportation/engine
 | keywords              = engine; speed
 | header                = speed; code; name, engine ; type
 | mappings              =
    {{ DBpediaPropertyMapping | templateProperty = engine |
    ontologyProperty = AutomobileEngine/name }}

    {{ DBpediaPropertyMapping | templateProperty = type |
    ontologyProperty = AutomobileEngine/type }}

    {{ DBpediaPropertyMapping | templateProperty = code |
    ontologyProperty = AutomobileEngine/code }}

    {{ DBpediaPropertyMapping | templateProperty = speed |
    ontologyProperty = AutomobileEngine/speed | unit = kilometerPerHour
    }}

  }}
```

R10. Table column headers can slightly differ in their naming but still have the same meaning and mapping.

Example: The rows of tables with the columns `Engine Name`, `Type`, `Engine code`, and `Top speed` are mapped to instances of the ontology class `Automobile Engine`, but the column `Top Speed kmh` would have the same meaning as the column `Top Speed`.

Implementation: We will allow regular expressions for the naming of table column headers.

```
  {{ DBpediaTableMapping
   | mapToClass = AutomobileEngine
   | mappings   =

    {{ DBpediaPropertyMapping | templateProperty = Engine Name |
    ontologyProperty = AutomobileEngine/name }}

    {{ DBpediaPropertyMapping | templateProperty = Type |
    ontologyProperty = AutomobileEngine/type }}

    {{ DBpediaPropertyMapping | templateProperty = Engine code |
    ontologyProperty = AutomobileEngine/code }}

    {{ DBpediaPropertyMapping | templateProperty = Top speed( kmh)? |
    ontologyProperty = AutomobileEngine/topSpeed | unit =
    kilometerPerHour }}

  }}
```

7. Complete Example Mapping

In this section we define an ontology schema for the ontology class `Automobile`, the ontology properties for the class `Automobile` and a template to ontology mapping for the Wikipedia template `http://en.wikipedia.org/wiki/Template:Infobox_automobile` to the ontology class `Automobile` along with mappings for all the template properties on this template. Furthermore we define a table to ontology mapping for a table listing engines for specific automobiles. The mapped `AutomobileEngine` instances shall be connected with the infobox template of the corresponding page or page section.

The given Wikipedia template `http://en.wikipedia.org/wiki/Template:Infobox_automobile` defines the following template properties:

- name
- manufacturer

- parent_company
- aka
- production
- production_start
- production_end
- model_years
- assembly
- predecessor
- successor
- class
- body_style
- layout
- platform
- engine
- transmission
- wheelbase
- length
- width
- height
- weight
- related
- designer

The table which rows are to be mapped to instances of `AutomobileEngine` has the following table column headers:

- Short Descr
    - alternative column name: Short Description
- Config/Cyls
- Fuel
- Displacement
- Power (Max Output)
- Torque (Max Output)
- Accel 0-100 km/h
    - alternative column name: Accel(eration) 0-100 km/h
- Top Speed

7.1. Ontology Schema Definition

The ontology class `Automobile` is defined as follows:

```
{{ DBpediaClass
| rdfs:label          = Automobile
| rdfs:subClassOf     = MeanOfTransportation
}}
```

The ontology properties for the ontology class `Automobile` and other ontology properties needed for the template mapping are defined as follows:

```
{{ DBpediaObjectProperty
 | rdfs:label  = manufacturing company
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = Organisation
}}

{{ DBpediaObjectProperty
 | rdfs:label  = parent company
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = Organisation
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = production start date
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = xsd:date
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = production end date
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = xsd:date
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = model start date
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = xsd:date
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = model end date
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = xsd:date
}}

{{ DBpediaObjectProperty
 | rdfs:label  = assembly
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = PopulatedPlace
}}

{{ DBpediaObjectProperty
 | rdfs:label  = predecessor
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = MeanOfTransportation
}}

{{ DBpediaObjectProperty
 | rdfs:label  = successor
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = MeanOfTransportation
}}

{{ DBpediaObjectProperty
 | rdfs:label  = class
 | rdfs:domain = Automobile
}}

{{ DBpediaObjectProperty
 | rdfs:label  = body style
 | rdfs:domain = Automobile
}}

{{ DBpediaObjectProperty
 | rdfs:label  = layout
 | rdfs:domain = Automobile
}}

{{ DBpediaObjectProperty
 | rdfs:label  = platform
 | rdfs:domain = Automobile
 | rdfs:range  = AutomobilePlatform
}}

{{ DBpediaObjectProperty
 | rdfs:label  = engine
 | rdfs:domain = Automobile
 | rdfs:range  = AutomobileEngine
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = transmission
 | rdfs:domain = Automobile
 | rdfs:range  = xsd:string
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = wheelbase
 | rdfs:domain = Automobile
 | rdfs:range  = millimeter
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = length
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = millimeter
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = width
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = millimeter
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = height
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = millimeter
}}

{{ DBpediaDatatypeProperty
 | rdfs:label  = weight
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = millimeter
}}

{{ DBpediaObjectProperty
 | rdfs:label  = related mean of transportation
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = MeanOfTransportation
}}

{{ DBpediaObjectProperty
 | rdfs:label  = designer
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = Person
}}

{{ DBpediaObjectProperty
 | rdfs:label  = design company
 | rdfs:domain = MeanOfTransportation
 | rdfs:range  = Organisation
}}
```

7.2. Template to Ontology Mapping

The template mapping to the ontology class `Automobile` for the template `Infobox Automobile` is defined as follows:

```
{{ DBpediaTemplateMapping
| mapToClass = Automobile
| mappings   =

  {{ DBpediaPropertyMapping | templateProperty = name | ontologyProperty =
  http://xmlns.com/foaf/0.1/name }}

  {{ DBpediaPropertyMapping | templateProperty = aka | ontologyProperty =
  http://xmlns.com/foaf/0.1/name }}

  {{ DBpediaPropertyMapping | templateProperty = manufacturer | ontologyProperty
  = MeanOfTransportation/manufacturingCompany }}

  {{ DBpediaPropertyMapping | templateProperty = parent_company |
  ontologyProperty = MeanOfTransportation/parentCompany }}

  {{ DBpediaDateIntervalMapping | templateProperty = production |
  startDateOntologyProperty = MeanOfTransportation/productionStartDate |
  endDateOntologyProperty = MeanOfTransportation/productionEndDate }}

  {{ DBpediaDateIntervalMapping | templateProperty = model_years |
  startDateOntologyProperty = MeanOfTransportation/modelStartDate |
  endDateOntologyProperty = MeanOfTransportation/modelEndDate }}

  {{ DBpediaPropertyMapping | templateProperty = assembly | ontologyProperty =
  MeanOfTransportation/assembly }}

  {{ DBpediaPropertyMapping | templateProperty = predecessor | ontologyProperty =
  MeanOfTransportation/predecessor }}

  {{ DBpediaPropertyMapping | templateProperty = successor | ontologyProperty =
  MeanOfTransportation/successor }}

  {{ DBpediaPropertyMapping | templateProperty = class | ontologyProperty =
  Automobile/class }}

  {{ DBpediaPropertyMapping | templateProperty = body_style | ontologyProperty =
  Automobile/bodyStyle }}

  {{ DBpediaPropertyMapping | templateProperty = layout | ontologyProperty =
  Automobile/layout }}

  {{ DBpediaPropertyMapping | templateProperty = platform | ontologyProperty =
  Automobile/platform }}

  {{ DBpediaIntermediateNodeMapping | nodeClass = AutomobileEngine |
  correspondingProperty = Automobile/engine | mappings =

    {{ DBpediaPropertyMapping | templateProperty = engine | ontologyProperty =
    AutomobileEngine/number }}

    {{ DBpediaPropertyMapping | templateProperty = engine | ontologyProperty =
    AutomobileEngine/power | unit = Power }}

    {{ DBpediaPropertyMapping | templateProperty = engine | ontologyProperty =
    AutomobileEngine/torque | unit = newtonMeter }}
  }}
  {{ DBpediaPropertyMapping | templateProperty = transmission | ontologyProperty
  = Automobile/transmission }}
  {{ DBpediaPropertyMapping | templateProperty = wheelbase | ontologyProperty =
  Automobile/wheelbase | unit = millimeter }}
  {{ DBpediaPropertyMapping | templateProperty = length | ontologyProperty =
  MeanOfTransportation/length | unit = Length }}
  {{ DBpediaPropertyMapping | templateProperty = width | ontologyProperty =
  MeanOfTransportation/width | unit = Length }}
  {{ DBpediaPropertyMapping | templateProperty = height | ontologyProperty =
  MeanOfTransportation/height | unit = Length }}
  {{ DBpediaPropertyMapping | templateProperty = weight | ontologyProperty =
  MeanOfTransportation/weight | unit = Weight }}
  {{ DBpediaPropertyMapping | templateProperty = related | ontologyProperty =
  MeanOfTransportation/relatedMeanOfTransportation }}
  {{ DBpediaPropertyMapping | templateProperty = designer | ontologyProperty =
  MeanOfTransportation/designCompany }}
  {{ DBpediaPropertyMapping | templateProperty = designer | ontologyProperty =
  MeanOfTransportation/designer }}
}}
```

7.3. Table to Ontology Mapping

The table mapping to the ontology class `AutomobileEngine` for the given table is defined as follows:

```
{{ DBpediaTableMapping
 | mapToClass            = AutomobileEngine
 | correspondingClass    = Automobile
 | correspondingProperty = MeanOfTransportation/engine
 | header                = engine, name; power, power&kw, power&ps, output;
                           displacement, displacement&cc; 0-100, 0-60
 | keywords              = engine; power,output
 | mappings              =
  {{DBpediaPropertyMapping | templateProperty = engine | ontologyProperty =
  foaf:name}}
  {{DBpediaPropertyMapping | templateProperty = name | ontologyProperty =
  foaf:name}}
  {{DBpediaPropertyMapping | templateProperty = power | ontologyProperty =
  AutomobileEngine/powerOutput | unit = Power }}
  {{DBpediaPropertyMapping | templateProperty = power&kw | ontologyProperty =
  AutomobileEngine/powerOutput | unit = Power | unit = kilowatt }}
  {{DBpediaPropertyMapping | templateProperty = power&ps | ontologyProperty =
  AutomobileEngine/powerOutput | unit = Power | unit = pferdestaerke }}
  {{DBpediaPropertyMapping | templateProperty = output | ontologyProperty =
  AutomobileEngine/powerOutput | unit = Power }}
  {{DBpediaPropertyMapping | templateProperty = displacement | ontologyProperty =
  AutomobileEngine/displacement | unit = Volume }}
  {{DBpediaPropertyMapping | templateProperty = displacement&cc |
  ontologyProperty = AutomobileEngine/displacement | unit = cubicCentimetre }}
  {{DBpediaPropertyMapping | templateProperty = 0-100 | ontologyProperty =
  AutomobileEngine/acceleration | unit = second}}
```

```
   {{DBpediaPropertyMapping | templateProperty = 0-60 | ontologyProperty =
   AutomobileEngine/acceleration | unit = second}}

}}
```