

1. Entwurf

Im Moment bietet das Ontowiki die Möglichkeit an, sich eine Person anzuschauen, indem man auf deren Eigenschaften zugreift. Diese Sicht ist aber nicht die schönste für die normalen Ontowikinutzer, weil da ein paar unklare und auch für solche Anwender irrelevante Daten zu finden sind.

Von daher kommt auch die Idee, die Eigenschaften einer Person mit Hilfe von HTML im Personenview darzustellen, um letztendlich eine Personendarstellung ähnlich der bisherigen Variante zu erzeugen.

Um diese Sicht zu schaffen, müssen zunächst die Datenbankabfragen gestellt werden. Solange die Daten vorhanden sind, werden die den entsprechenden Buchstaben zugeordnet, die als Abkürzung der jeweiligen Kategorien dienen.

In ein paar Kategorien, die die Periods of Life betreffen, werden noch zusätzlich die Zeitspannen hinzugefügt.

Was Schwierigkeiten bereiten kann, sind die Kommentare, die manchmal auftauchen und auch nur ab und zu durchnummeriert sind. Um die mehrfache Nummerierung zu vermeiden, verzichten wir darauf überhaupt und lassen Kommentare ohne zusätzliche Ordnungszahlen stehen.

2. Dokumentation

```
$query->setProloguePart('SELECT ?label ')->setWherePart(
    'WHERE{ <'.$resource.'> <http://drw-model.saw-leipzig.de/hasFatherComment>
    ?FatherComment . ' .
    '?FatherComment <http://www.w3.org/2000/01/rdf-schema#label> ?label . ' .
    '});
$queryResult = $this->model->sparqlQuery($query);
if ((is_array($queryResult) || isset ($queryResult[0]) || is_array($queryResult[0]))){
    foreach($queryResult as $item){
        $string .= $item["label"] ;
        $string .= "<br>";
    }
}
```

Im PHP wird eine Abfrage an die Datenbank gestellt. Falls es das Label gibt, werden die Ergebnisse geliefert und im \$string gespeichert. Gäbe es mehr als ein Ergebnis zu dem Label, werden diese durch den Punktoperator zu einem String verkettet.

```
<tr>
    <td class="t100g"><h3><?php echo $this->MNamestring; ?></h3></td>
</tr>
```

Der gespeicherte String wird in HTML übergeben und an der bestimmten Stelle angezeigt.

```
$query->setProloguePart('SELECT ?image ')->setWherePart(
    'WHERE{ <'.$resource.'> <http://xmlns.com/foaf/0.1/img> ?image . ' .
    '});
$queryResult = $this->model->sparqlQuery($query);
foreach($queryResult as $item){
    $string .= $item["image"] ;
}
}
```

```
$this->view->Bildstring = $string;  
$string = "";
```

Die gleiche Vorgehensweise kann man auch bei dem Bild anwenden, indem man als Ergebnis dessen URL-Adresse bekommt und die dann in HTML zu einem Image transformiert.

3. Testfälle

- falls die Informationen unter Eigenschaften vorhanden sind, werden die auch im Personenview angezeigt – z.B. Vaterangaben
- ist das Transliterationsergebnis auch im Personenview sichtbar?
- wird das Bild richtig angezeigt?
- sind die Daten im richtigen Format, um sie formatiert auszugeben?