

OpenSeaMap mit Semantic MediaWiki

Seminararbeit von

Christian Lütje

Am Institut für Angewandte Informatik und Formale
Beschreibungsverfahren (AIFB)

31. März 2014

Betreuer: Basil Ell

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufgabenstellung	3
1.2	Semantic MediaWiki	3
1.3	OpenSeaMap	4
2	Konzept	5
2.1	Integration der OpenSeaMap	5
2.2	Strukturierung der Daten	6
3	Realisierung	7
3.1	Erweiterung der Maps-Extension	7
3.2	Implementierung des Subobjects und der Formulare	10
4	Fazit und Ausblick	13
4.1	Bestehende Probleme	13
4.2	Verbesserungsmöglichkeiten	15

1 Einleitung

Die vorliegende Seminararbeit beschäftigt sich mit der Konzeption eines auf Semantic MediaWiki basierenden, interaktiven Online-Portals für Wassersportler und ist im Rahmen des Seminars “Anwendungen von Semantic MediaWiki“ am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) des Karlsruher Instituts für Technologie (KIT) entstanden. Ziel des Seminars war es, Konzepte für innovative Anwendungen auf Basis von Semantic MediaWiki zu konzipieren und zu präsentieren. Die Seminararbeit gibt im Folgenden einen Überblick über die konkrete Aufgabenstellung, das Konzept zur Lösung und dessen Realisierung, zieht dann ein Fazit und gibt einen Ausblick auf Verbesserungs- und Erweiterungsmöglichkeiten.

1.1 Aufgabenstellung

Dienste wie Google Maps erfreuen sich in den letzten Jahren wachsender Popularität und bieten inzwischen eine Vielzahl an Informationen über Orte, wie z.B. die Lage von Geschäften und Ärzten inklusive weiterer zugehöriger Daten wie Öffnungszeiten und Kontaktmöglichkeiten. Diese Dienste beschränken sich jedoch auf die Landperspektive, was sie für die Bedürfnisse von Wassersportlern weniger attraktiv macht.

Ziel der Aufgabe ist also die Konzeption und Realisierung eines Online-Portals, auf dem Wassersportler selbsttätig neue nautisch und touristisch relevante Informationen über Orte auf dem Wasser und an Land eintragen können sollen. Diese Orte können beispielsweise Fischgründe, Tauchspots, Häfen oder Reparaturwerkstätten sein, aber auch nautische Gefahrenstellen wie Untiefen, die nicht in den offiziellen Seekarten kartographiert sind. Des Weiteren sollen diese Orte auf einer Karte inklusive der besonderen nautischen Zeichen wie z.B. Tonnen, Brücken und Leuchttürmen dargestellt werden können. Das Online-Portal soll über ein Semantic MediaWiki realisiert werden und die Kartendarstellung mit Hilfe der OpenSeaMap erfolgen.

1.2 Semantic MediaWiki

Semantic MediaWiki (SMW)¹ ist eine Erweiterung der MediaWiki-Software², einer Implementation eines Wikis. Ein Wiki ist eine Webseite, die die Inhalte in Artikeln und Kategorien organisiert und die grundsätzlich von jedem online im Browser bearbeitet und um neue Inhalte erweitert werden kann. Ein bekanntes Beispiel für ein Wiki auf Basis von MediaWiki ist die Online-Enzyklopädie *Wikipedia*³ [1, S. 3-4].

¹<http://semantic-mediawiki.org>

²<http://www.mediawiki.org/wiki/MediaWiki/de>

³<http://de.wikipedia.org>

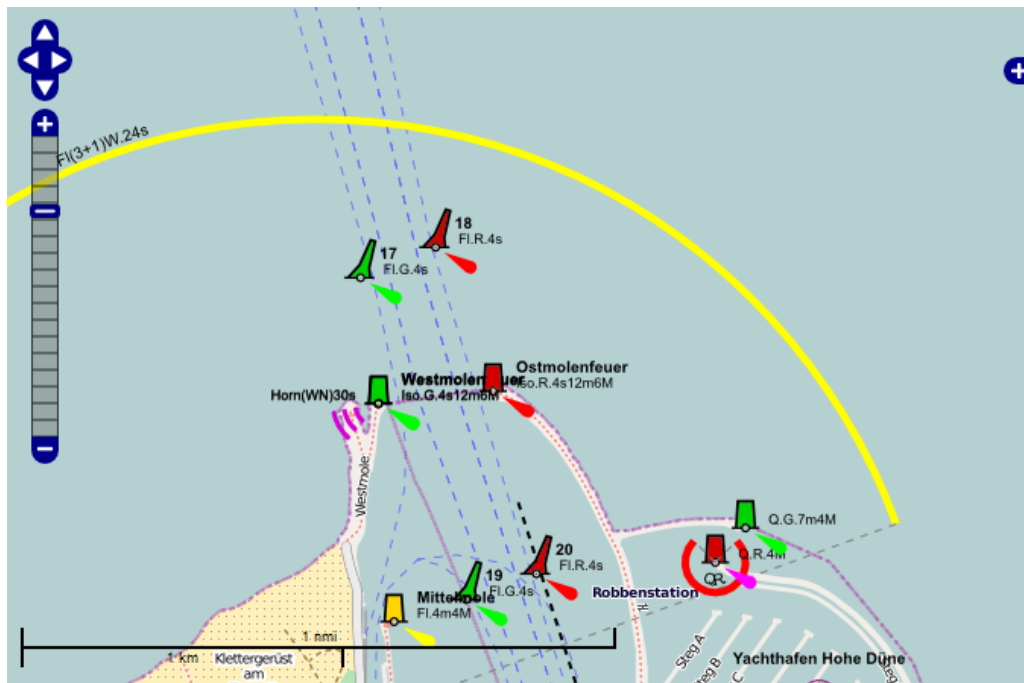


Abbildung 1: Seekartendarstellung mit OpenSeaMap

SMW erweitert MediaWiki um semantische Funktionen, mit denen sich Informationen im Wiki strukturieren sowie maschinell lesen und verarbeiten lassen. Mit entsprechenden Erweiterungen bietet SMW auch Formularfunktionen, so dass Daten bequem neu eingeben bzw. editiert werden können ohne die Syntax des Wiki-Codes beherrschen zu müssen. Im Kontext der Aufgabenstellung ermöglicht SMW also, dass wassersportlich-relevante Plätze mit weiteren Zusatzinformationen komfortabel beschrieben und über komplexe Suchanfragen auch wieder gefunden werden können.

Sowohl MediaWiki als auch SMW sind Open Source-Anwendungen und stehen kostenlos zur Verfügung.

1.3 OpenSeaMap

Wie eingangs erwähnt, soll die Darstellung der Orte mittels der OpenSeaMap erfolgen. Die OpenSeaMap⁴ ist ein freies, community-gestütztes Projekt, das es sich zum Ziel gesetzt hat, eine elektronische Seekarte aufzubauen, die von jedermann frei nutz- und erweiterbar ist. OpenSeaMap nutzt hierfür die Kartendaten des OpenStreetMap-Projektes und ergänzt sie um nautische Informationen wie Seezeichen, Leuchtfeuer und Hafenanlagen, die als eigene Ebenen (Layer) über die Kartendaten gelegt werden [2]. Abbildung 1 zeigt eine Seekartendarstellung mit OpenSeaMap.

Die OpenSeaMap unterliegt allerdings einigen Beschränkungen:

⁴<http://www.openseamap.org>

- Das Einfügen neuer Objekte in die Seekarte ist relativ komplex und bedarf der Installation und Einrichtung eines externen Editors [3]
- Der Typ der Objekte in der Seekarte ist im Wesentlichen beschränkt auf Seezeichen und Häfen
- Zusatzinformationen zu den Objekten lassen sich nur in geringem Maße eingeben
- Komplexe Suchen nach Objekten in der Seekarte sind nicht möglich

Die Nutzung von SMW bietet sich hier also an, um einerseits mit semantischen Attributen die zusätzlichen Informationen über die Orte zu strukturieren und andererseits dem Benutzer eine komfortable Eingabe- und Editiermöglichkeit zur Verfügung zu stellen, die ohne weitere externe Software auskommt. Ein weiterer Vorteil ist die leichte Exportierbarkeit der Daten aus dem SMW in verschiedene Datenformate wie RDF oder XML und die daraus resultierende Weiterverwendbarkeit der Daten.

2 Konzept

Für die Umsetzung des Wassersportportals auf SMW-Basis gemäß der Aufgabenstellung lassen sich zwei zentrale Problemstellungen identifizieren:

- Die Integration der OpenSeaMap in das Wiki
- Eine sinnvolle Strukturierung der Daten mit semantischen Attributen

2.1 Integration der OpenSeaMap

Es existieren bereits einige Erweiterungen (Extensions) für MediaWiki, mit denen Kartenmaterial in das Wiki integriert werden kann. Eine der populärsten Karten-Extensions ist *Maps*⁵, die hauptsächlich auf die Integration von Google Maps ausgerichtet ist, aber auch andere Kartendienste wie Microsofts Bing Maps, Yahoo Maps oder die OpenStreetMap einbinden kann. Die Extension *Semantic Maps*⁶ ist ein Aufsatz für die *Maps*-Extension und fügt ihr semantische Funktionen hinzu, u.a. einen neuen Datentyp für geographische Koordinaten. *Semantic Maps* ermöglicht auch die Suche nach geographischen Daten mit semantischen Suchabfragen.

Nach einer Analyse der Funktionsweise und des Quellcodes der *Maps* und *Semantic Maps*-Extension sowie der Hinweise zum Einbinden von OpenSeaMap auf Webseiten [4] zeigte sich, dass die OpenSeaMap als weitere Karte in die beiden angesprochenen Extensions eingebunden werden kann und somit die Entwicklung einer neuen eigenständigen Extension für die OpenSeaMap nicht sinnvoll erschien.

⁵<http://www.mediawiki.org/wiki/Extension:Maps>

⁶http://semantic-mediawiki.org/wiki/Semantic_Maps

Die Einbindung der OpenSeaMap ist im Unterabschnitt 3.1 genauer beschrieben.

Darüber hinaus wurde eine weitere Karte eingebunden, die detaillierte Informationen über Brücken sowie die Wasserwege nach CEMT-Klassifikation zeigt. Die CEMT-Klassifikation teilt die Binnenwasserstrassen in verschiedene Klassen anhand der größtmöglichen Abmessung der Wasserfahrzeuge ein, von denen sie gerade noch befahren werden können [5]. Diese Karte wird nicht von OpenSeaMap betrieben, sondern von einem OpenStreetMap-User gepflegt⁷, nutzt aber die selben Mechanismen wie OpenSeaMap und wurden daher ebenfalls in das Wiki-Portal eingebunden.

2.2 Strukturierung der Daten

Um die zu den Orten eingegebenen Zusatzinformationen maschinell verarbeitbar zu machen, ist eine sinnvolle Beschreibung der Daten mittels geeigneter semantischer Attribute nötig. Eine einheitliche Klassifikation der Orte ist jedoch schwierig, da naturgemäß viele unterschiedliche Typen von Orten in das Wassersportportal eingegeben werden können, die wiederum jeweils unterschiedlich viele sinnvolle semantische Attribute haben. So sind beispielsweise für ein Freiluftschwimmbad *Straße* und *Öffnungszeiten* sinnvolle Attribute, wohingegen die selben Attribute bei der Beschreibung eines Schiffswracks nicht sinnvoll sind. In Tabelle 1 wird diese Problematik noch einmal anhand eines Beispiels verdeutlicht, die fraglichen Attributwerte sind hier mit einem ? markiert.

Attribute	Rheinstrandbad	Wrack "RMS Titanic"
Position	48° 59' 56"N, 8° 17' 32E	41° 43' 57"N, 49° 56' 49"W
Typ	Freiluftbad	Wrack
Seegebiet	Rhein	Atlantik
Land	Deutschland	?
Öffnungszeiten	8.00-20.00 Uhr	?
Straße	Hermann-Schneider-Allee 54	?
Liegt_in_Tiefe	?	3821m

Tabelle 1: Strukturierung der Daten mit semantischen Attributen

Als gemeinsame Attribute für alle Typen von Orten sind somit nur wenige sinnvoll, die in Tabelle 2 exemplarisch angegeben sind.

Neben diesen fest vorgegeben Attributen soll der Benutzer aber auch selbst weitere Attribute hinzufügen können, mit denen der Ort näher beschrieben werden kann. Diese Funktionalität wird über das Objekt *Subobject* realisiert, welches aus den zwei Attributen *Property name* und *Property value* besteht, die ein Paar aus Attributname und Attributwert repräsentieren. Jeder Ort kann beliebig viele Instanzen von *Subobject* haben und somit stehen dem Benutzer in der Theorie

⁷<http://maps.grade.de/legende.htm>

Attribut	Beschreibung
Has position	Die geographischen Koordinaten des Ortes
Is type of	Typ des Ortes (Hafen, Tauchspot...)
Has sea area	Das Seegebiet, an oder in dem der Ort liegt
Is interesting for	Die Zielgruppe(n), für die dieser Ort interessant sein könnte

Tabelle 2: gemeinsame Attribute aller Orte

unendlich viele selbstdefinierbare Felder zur Verfügung, mit denen er den Ort genauer beschreiben kann. Die Erstellung der *Subobjects* erfolgt dabei komfortabel über ein Formular bei der Anlage eines neuen Ortes, so dass keinerlei Kenntnisse über die Syntax des zugrundeliegenden Wikicodes nötig sind. Der Benutzer hat hier auch die Möglichkeit, aus einer Dropdown-Liste bereits im Wiki verwendete Attribute auszuwählen und bekommt über eine Autocomplete-Funktion bei der Eingabe Vorschläge für Attributnamen und Werte. Auf diese Weise wird eine große Freiheit erreicht, ob und mit welchen Attributen der Benutzer einen Ort beschreibt, gleichzeitig bleibt aber eine gute Konsistenz der Datenstruktur gewährleistet. Abbildung 2 zeigt das Eingabeformular mit der Dropdown-Funktionalität. Die Details der Implementierung werden in Unterabschnitt 3.2 beschrieben.

Abbildung 2: Dynamisches Eingabeformular

3 Realisierung

Der folgende Abschnitt beschreibt die Erweiterung der *Maps*-Extension sowie die Implementierung der verwendeten Datenstruktur genauer. Tabelle 3 liefert eine Übersicht über die verwendeten Anwendungen bzw. Extensions. Eine lauffähige Demoversion des Wassersportportals ist unter <http://www.student.kit.edu/~uvau/mediawiki> verfügbar.

3.1 Erweiterung der Maps-Extension

Die *Maps*-Extension bedient sich verschiedener Schnittstellen zum Einbinden von Kartenmaterial. Für Google Maps wird die google-eigene API verwendet,

Komponente	Version	Beschreibung
MediaWiki	1.21.3	Die grundlegende Wiki-Software
Semantic MediaWiki	1.8.0.5	Erweitert MediaWiki um semantische Funktionen
Validator	0.5.1	Stellt generische Parameterbehandlung für andere Extensions zur Verfügung
ParserFunctions	1.5.1	Erweitert den Parser um logische Funktionen
Maps	2.0.1	Bindet Kartenmaterial in das Wiki ein
Semantic Maps	2.0.1.2	Erweitert <i>Maps</i> um semantische Funktionen
Semantic Forms	2.5.3	Stellt Formulare zur Eingabe und zum Bearbeiten von semantischen Daten zur Verfügung

Tabelle 3: Verwendete Anwendungen und Extensions

wohingegen für OpenStreetMap, auf dessen Karten die OpenSeaMap aufbaut, die Bibliothek *OpenLayers*⁸ zum Einsatz kommt. *OpenLayers* ist eine freie JavaScript-Bibliothek zur Entwicklung von Web-Mapping-Anwendungen und bietet eine einheitliche API, um verschiedenste Kartendienste einzubinden [6, S. 57]. Darüber hinaus stellt *OpenLayers* verschiedene grundlegende Methoden und Steuerelemente zur Verfügung, mit denen die Karte beispielsweise verschoben, vergrößert/verkleinert oder Mausclicks innerhalb der Karte verarbeitet werden können.

OpenLayers unterscheidet zwischen sogenannten Baselayern und einfachen Layern [6, S. 75]. Baselayer sind Basiskarten, von denen sinnvollerweise immer nur eine zur Zeit angezeigt werden kann. Beispiele für Basiskarten sind topographische Landkarten, Luftkarten oder Satellitenkarten. Einfache Layer sind transparente Ebenen, die über den Baselayer gelegt werden können und die Basiskarte um weitere Informationen ergänzen. Typischerweise besteht eine Web-Mapping-Karte somit aus einem Baselayer mit einem oder mehreren darüber liegenden einfachen Layern mit weiteren Informationen.

OpenStreetMap bietet für die Nutzung über *OpenLayers* eine spezielle JavaScript-Datei⁹ an, in dem die Layer definiert sind, die der Kartendienst anbietet. Diese JavaScript-Datei muss zunächst eingebunden werden, um den Kartendienst mit *OpenLayers* nutzen zu können. Bei der *Maps*-Extension ist diese JavaScript-Datei bereits standardmäßig eingebunden. OpenStreetMap bietet verschiedene Baselayer an, für das Wassersportportal wird der mit *Mapnik* gerenderte Baselayer verwendet. *Mapnik*¹⁰ ist ein OpenSource-Renderer, der Grafiken mit Kartenkacheln aus Geodaten erzeugt. Der Vorteil ist hier, dass die

⁸<http://www.openlayers.org>

⁹für OpenStreetMap: <http://www.openstreetmap.org/openlayers/OpenStreetMap.js>

¹⁰<http://www.mapnik.org>

Geodaten nicht auf dem Client gerendert werden müssen, sondern bereits fertig auf dem Server vorhanden sind und direkt ausgeliefert werden können.

Um den von OpenSeaMap angebotenen Layer einzubinden, muss dieser in die angesprochene JavaScript-Datei von OpenStreetMap eingetragen werden. Diese Datei namens *OpenStreetMap.js* befindet sich ausgehend vom Hauptverzeichnis der *Maps*-Extension unter *includes/services/OpenLayers/OSM*. Das Listing 1 zeigt die nötigen Ergänzungen der *OpenStreetMap.js*. Hier wird ein neuer Layer unter dem Namen OSeaM definiert und in Zeile 11 die Adresse des OpenSeaMap-Servers angegeben, auf dem sich die Kartenkacheln des Layers befinden. In Zeile 13-19 werden weitere Optionen für den Layer festgelegt, u.a. die Anzahl der verfügbaren Zoomstufen, die Transparenz sowie die Nutzung als Nicht-Baselayer.

```
1 OpenLayers.Layer.OSM.OSeaM = OpenLayers.Class(OpenLayers.Layer.OSM,
2   {
3     /**
4     * Constructor: OpenLayers.Layer.OSM.OSeaM
5     *
6     * Parameters:
7     * name - {String}
8     * options - {Object} Hashtable of extra options to tag onto
9     *           the layer
10    */
11   initialize: function(name, options) {
12     var url = [
13       "http://t1.openseamap.org/seamark/{z}/{x}/{y}.png"
14     ];
15     options = OpenLayers.Util.extend({
16       numZoomLevels: 19,
17       buffer: 0,
18       transitionEffect: "resize",
19       alpha           : true,
20       isBaseLayer    : false
21     }, options);
22     var newArguments = [name, url, options];
23     OpenLayers.Layer.OSM.prototype.initialize.apply(this,
24       newArguments);
25   },
26   CLASS_NAME: "OpenLayers.Layer.OSM.OSeaM"
27 });
```

Listing 1: Erweiterung der *OpenStreetMap.js*

Der in Unterabschnitt 2.1 angesprochene zusätzliche Layer mit den Informationen über Brücke und Wasserstraßen sowie der von OpenSeaMap bereitgestellte Layer “Sport“ mit Informatonen über Kajakstrecken und Schwimmmöglichkeiten wurden in gleicher Weise eingebunden.

Zusätzlich ist die Anpassung der Konfigurationsdatei *Maps_Setting.php* im Hauptverzeichnis der *Maps*-Extension nötig. Hier müssen die neuen Layer eingetragen werden, damit sie dem Benutzer im Wiki zur Verfügung stehen. In

Listing 2 sind die zu ergänzenden Codezeilen angegeben. Den Layern werden hier in Zeile 9, 10 und 11 Bezeichner zugewiesen, unter denen sie dann später im Wiki angesprochen werden können. Außerdem kann hier der Text festgelegt werden, der in der Legende der Karte erscheinen soll.

```

1 $egMapsOLLayers = array(
2   ...
3   'osm-oseam',
4   'osem-oseam-sport',
5   'osm-oseam-cemt'
6 );
7
8 $egMapsOLAvailableLayers = array(
9   ...
10  'osm-oseam' => array('OpenLayers.Layer.OSM.OSeaM("Seamarks)", '
    osm'),
11  'osm-oseam-sport' => array('OpenLayers.Layer.OSM.OSeaMSport("
    Sports)", 'osm'),
12  'osm-oseam-cemt' => array('OpenLayers.Layer.OSM.OSeaMCEM("
    Waterways/Bridges)", 'osm'),
13 );
14
15 $egMapsOLLayerGroups = array(
16   ...
17   'osm' => array('osmarender', 'osm-mapnik', 'osm-cyclemap', 'osm-
    oseam', 'osm-oseam-sport', 'osm-oseam-cemt'),
18 );

```

Listing 2: Erweiterung der *Maps_Settings.php*

Die Karte kann dann im Wiki über die *#display_map*-Parserfunktion aufgerufen werden, wobei die neuen Layer *osm-oseam*, *osm-oseam-sport* und *osm-oseam-cemt* in *layers*-Paramater angegeben werden müssen. Listing 3 zeigt einen solchen Aufruf und Abbildung 3 die daraus resultierende Darstellung im Wiki.

```

1 {{#display_map: {{{Position}}}}
2 | height=800px
3 | width=500px
4 | zoom=14
5 | service=openlayers
6 | layers=osm-mapnik, osm-oseam, osm-oseam-sport, osm-oseam-cemt }}

```

Listing 3: Wiki-Code zur Anzeige der Karte

3.2 Implementierung des Subobjects und der Formulare

Wie schon in Unterabschnitt 2.2 beschrieben, soll der Nutzer die Möglichkeit haben, zu jedem Ort beliebig viele freie Felder zur Beschreibung definieren zu können. Diese Funktionalität wird durch die von *Semantic MediaWiki* bereitgestellte Parserfunktion *#subobject* realisiert, die die Speicherung von



Abbildung 3: Seekartendarstellung mit OpenSeaMap im Wiki

Attribut-Attributwertpaaren ermöglicht, ohne gesonderte Seiten im Wiki für diese Paare erstellen zu müssen [7]. Zunächst wurden zwei neue semantische Attribute *Property name* und *Property value* angelegt, die den Namen und den Wert des Attributes aufnehmen. Das Template *Dynamic Property* verbindet diese beiden Attribute dann über die *#subobject*-Parserfunktion zu einem Attribut-Attributwertpaar:

```
{{ #subobject: |Property name={{Property name|}}|Property value={{Property value|}} }}
```

In einem Eintrag für einen Ort kann dieses Template nun genutzt werden (auch mehrfach), um Paare aus Attributname und Attributwert abzubilden. Listing 4 zeigt einen solchen Eintrag, in dem als zusätzliche Felder die Durchfahrtshöhe sowie das Baumaterial einer Brücke eingetragen wurden.

```

1  {{Place
2  |Position=49.036729459596 , 8.3034324645996
3  |Is type of=Bridge
4  |Located on=Sea
5  |Country=Germany
6  |Sea area=Rhein
7  |Nearest city=Karlsruhe
8  |Interesting for=Sailing yachts , Motor yachts , Surfers , Rowers /
   Kayaks
9  }}
10 {{Dynamic Property
11 |Property name=Clearance height
12 |Property value=9,25m
13 }}
14 {{Dynamic Property
15 |Property name=Building materials
16 |Property value=Concrete , Steel
17 }}
18 Bridge over the Rhein.
```

Listing 4: Wiki-Code eines Ortseintrages

Des weiteren wird das Template für das Formular zur Eingabe eines neuen bzw. zum Bearbeiten eines Ortes verwendet. Dort wird die von *Semantic Forms* bereitgestellte Möglichkeit für multiple Formulare, also die mehrfache Nutzung eines eigenständigen Formulars innerhalb eines Formulars, genutzt [8]. Dieses Formular ist mit dem *Dynamic Property*-Template verknüpft und der Benutzer kann so beliebig viele Instanzen des Dynamic Property Objektes anlegen wie in Abbildung 2 gezeigt. Der für das multiple Formular notwendige Wiki-Code ist in Listing 5 ersichtlich.

Bei der Anzeige des Wiki-Artikels zu einem Ort fragt eine eingebettete semantische Abfrage alle zu diesem Ort gehörenden Attribut-Attributwertpaare ab und formatiert die Ausgabe ansprechend über das eigene Template *Dynamic Property Output* [9].

```

1 {{{for template|Dynamic Property|label=Add more information|
   multiple}}}
2
3 {| class="formtable"
4 | colspan="4"| Select a property or create a new property and add
   some information in the value field:
5 |-
6 ! Property:
7 | {{{field|Property name|input type=combobox}}}
8 ! Value:
9 | {{{field|Property value|input type=text}}}
10 |}
11
12 {{{end template}}}

```

Listing 5: Wiki-Code zur Erzeugung eines multiplen Formulars

Die eingebettete semantische Abfrage

```

{{{#ask: [| -Has subobject::{{{FULLPAGENAME}} }|]
|format=template
|template=Dynamic Property Output
|? Property name
|? Property value
}}}

```

ist im Template *Place* notiert und gibt die Attribute *Property name* und *Property value* zurück, in denen die Informationen über den Ort gespeichert sind. Die Ausgabe erfolgt dann in einer “Factbox“ wie in Abbildung 4 gezeigt.

4 Fazit und Ausblick

Als Fazit lässt sich ziehen, dass die Integration von OpenSeaMap in ein SMW dank der offenen Architektur der *Maps*-Extension sowie der Nutzung von *OpenLayers* mit relativ geringem Aufwand möglich ist. Die leistungsstarken Formularfunktionen von *Semantic Forms* ermöglichen es mit der WikiCode-Syntax nicht vertrauten Benutzern, Orte bzw. Daten in das Wiki einzutragen und zu beschreiben. An die Autoren der *Maps*-Extension um Jeroen De Dauw¹¹ wurde der Vorschlag herangetragen, die Änderungen für OpenSeaMap dauerhaft in die *Maps*-Extension aufzunehmen, so dass OpenSeaMap als Karte standardmäßig in einem SMW zur Verfügung steht.

4.1 Bestehende Probleme

Beim Anlegen bzw. Editieren eines Ortes soll der Benutzer dessen Position mit einem Klick in eine Karte festlegen können. Hierfür sollte natürlich auch die

¹¹http://www.mediawiki.org/wiki/User:Jeroen_De_Dauw

Rheinstrandbad Rappenwört

Position: 48° 59' 56" N, 8° 17' 32" E

Type: Beach

Located on: Land

Country: Germany

Sea Area: Rhein

Nearest city: Karlsruhe

Interesting for: Swimmers

Street: Hermann-Schneider-Allee 54

Postal code: 76189

City: Karlsruhe

Website: <http://www.ka-baeder.de/rappenwoertbad/>

Telephone: +497211335229

Email:

Opening hours: 8.00h - 20.00h

Abbildung 4: Ausgabe der Daten im Wiki

OpenSeaMap-Karte genutzt werden. Es zeigte sich jedoch, dass das von *Semantic Maps* bereitgestellte Kartenformularfeld in Verbindung mit über *OpenLayers* eingebundenen Karten nicht funktioniert. Auch mit Hilfe der SMW-Community ließ sich für dieses Problem keine Lösung finden, so dass aktuell als Workaround die Google Maps-Karte zur Eingabe von Koordinaten dient.

Generell lässt sich feststellen, dass die Karten-Extensions sich eher auf die Integration von Google Maps fokussieren, die robuster und besser dokumentiert ist als für über *OpenLayers* eingebundene Karten.

4.2 Verbesserungsmöglichkeiten

Für die Auswahl der Koordinaten eines Ortes über eine Karte kann es hilfreich sein, zwischen verschiedenen unterschiedlichen Karten umschalten zu können. Beispielsweise enthält Google Maps in der Regel detailliertere Informationen über Orte an Land als die OpenSeaMap, so dass die Auswahl eines landgebundenen Ortes über Google Maps einfacher sein kann als über OpenSeaMap.

Eine weitere Idee zur Verbesserung wäre die Nutzung von externen semantischen Daten, um beispielsweise automatisiert die zu einem Ort nächst gelegene größere Stadt zu finden. Dazu könnte das Wiki zu einer gegebenen Position die Städte in einem bestimmten Umkreis ermitteln, die Stadt mit den meisten Einwohnern auswählen und als Vorschlag in das entsprechende Formularfeld eintragen.

Hilfreich wäre es auch, Flächen oder Polygone zur Darstellung von Fischgründen, Kajakstrecken u.ä. nutzen zu können. Hierfür wäre die Entwicklung eines Karten-Formularfeldes nötig, das die Eingabe von mehreren Punkten unterstützt, so dass Flächen und Polygone komfortabel erstellt werden können. Für Google Maps gibt es hier bereits einen erweiterten Map-Editor¹², der die Eingabe von mehreren Punkten unterstützt. Allerdings kann der erzeugte Wiki-Code nur unbequem mit Copy&Paste übernommen werden.

Denkbar wäre außerdem eine Plausibilitätsprüfung bei der Eingabe, die prüft, ob Orte nicht übereinander liegen oder ein Ort an Land auch wirklich an Land liegt und nicht im Wasser.

Zum Datenaustausch mit der OpenSeaMap könnte ein Programm entwickelt werden, welches die semantischen Daten des SMW in ein für OpenSeaMap verständliches Format konvertiert und die Orte im Wiki somit auch direkt in die OpenSeaMap übernommen werden können. OpenStreetMap bietet hierfür eine XML-RPC-Schnittstelle an¹³, die sich auch für OpenSeaMap nutzen lassen sollte. Auf diese Weise wäre ein Datenaustausch zwischen dem Wiki-Portal und der OpenSeaMap möglich. Alternativ könnte OpenSeaMap auch auf das Wiki zugreifen, die semantischen Daten auslesen und in die eigene Karte integrieren.

¹²<http://www.student.kit.edu/~uvau/mediawiki/index.php/Special:MapEditor>

¹³<http://josm.openstreetmap.de/xmlrpc>

Literatur

- [1] Daniel J. Barrett. *MediaWiki : [Wikipedia and beyond]*. 1. ed. Beijing: O'Reilly, 2009. ISBN: 978-0-596-51979-7.
- [2] *OpenSeaMap: Häufig gestellte Fragen*. URL: <http://www.openseamap.org/index.php?id=faq&L=0>.
- [3] *JOSM and Plugin*. URL: http://wiki.openseamap.org/index.php?title=De:JOSM_and_Plugin.
- [4] *OpenSeaMap in Website*. 2010. URL: http://sourceforge.net/apps/mediawiki/openseamap/index.php?title=De:OpenSeaMap_in_Website.
- [5] Generaldirektion Wasserstraßen und Schifffahrt -Außenstelle West-. *Wasserstrassenklasse*. 2010. URL: http://www.wsd-west.wsv.de/wasserstrassen/verkehrsweg_rhein/technische_daten/Wasserstrassenklasse.html.
- [6] Marc Jansen und Till Adams. *OpenLayers : Webentwicklung mit dynamischen Karten und Geodaten*. München: Open Source Press, 2010. ISBN: 978-3-937514-92-5.
- [7] *Parsererweiterung #subobject*. 2012. URL: http://semantic-mediawiki.org/wiki/Help:Parsererweiterung_subobject.
- [8] *Semantic Forms/Defining forms*. 2014. URL: http://www.mediawiki.org/wiki/Extension:Semantic_Forms/Defining_forms#Multiple-instance_templates.
- [9] *Eingebettete Abfrage*. 2014. URL: https://semantic-mediawiki.org/wiki/Help:Eingebettete_Abfrage.

Abbildungsverzeichnis

1	Seekartendarstellung mit OpenSeaMap	4
2	Dynamisches Eingabeformular	7
3	Seekartendarstellung mit OpenSeaMap im Wiki	11
4	Ausgabe der Daten im Wiki	14

Tabellenverzeichnis

1	Strukturierung der Daten mit semantischen Attributen	6
2	gemeinsame Attribute aller Orte	7
3	Verwendete Anwendungen und Extensions	8

Hiermit versichere ich, dass ich die vorliegende Seminararbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Karlsruhe, den 31.03.2014