

# **SUI**

## **Eine Service-orientierte Schnittstelle zur Einbindung von Fachsystemen in die semantische Suche nach Umweltinformationen**

*C. Düpmeier; C. Greceanu; T. Schlachter; R. Weidemann  
Karlsruher Institut für Technologie  
Institut für Angewandte Informatik  
Hermann-von-Helmholtz-Platz 1  
76344 Eggenstein-Leopoldshafen*

*U. Bügel; F. Chaves; M. Schmieder; B. Schnebel  
Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung  
Fraunhoferstr. 1  
76131 Karlsruhe*

*R. Ebel; M. Tauber  
LUBW Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg  
Griesbachstr. 1  
76185 Karlsruhe*

*K. Zetzmann  
Ministerium für Umwelt, Klima und Energiewirtschaft Baden-Württemberg  
Kernerplatz 9  
70182 Stuttgart*

<b>1. EINLEITUNG.....</b>	<b>39</b>
<b>2. RESTFUL SERVICES.....</b>	<b>40</b>
<b>3. GENERISCHE SUCHSCHNITTSTELLEN AUF BASIS VON FEED-FORMATEN.....</b>	<b>41</b>
<b>4. EINE GENERISCHE SERVICE-API FÜR CMS-SYSTEME IM UIS .....</b>	<b>42</b>
4.1 IMPLEMENTIERUNG DER SERVICE-API .....	44
<b>8. FAZIT UND AUSBLICK .....</b>	<b>45</b>
<b>9. LITERATUR.....</b>	<b>46</b>

# 1. Einleitung

Im Projekt Semantische Suche nach Umweltinformationen (SUI) vom Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) und dem Institut für Angewandte Informatik (IAI) des Karlsruher Instituts für Technologie (KIT) wurden in Zusammenarbeit mit der LUBW Landesanstalt für Umwelt, Messungen und Naturschutz und dem Ministerium für Umwelt, Klima und Energiewirtschaft Baden-Württemberg (UM BW) Informationstechnologien entwickelt, mit denen die Volltextsuche in Umweltportalen durch den Einsatz von semantischen Technologien und Serviceorientierung verbessert werden kann /1/.

Das Konzept von SUI basiert u.a. darauf, dass eine semantische Beschreibung der Informationen von ausgewählten Fachinformationssystemen (sogenannter Zielsysteme) in Zielsystembeschreibungen (Definition von Informationsklassen, Orts- und/oder Zeitbezug, zugehöriger URL-Aufruf) im Portal gespeichert wird. Im Rahmen einer intelligenten Vorverarbeitung von Suchbegriffen im Portal /2/, welche zur inhaltlichen Klassifikation der Suchbegriffe die im Rahmen des SUI-Projektes entwickelte SUI-Ontologie verwendet, werden Suchbegriffe dann auf die jeweiligen inhaltlichen Kategorien, Orts- und Zeitangaben abgebildet. Anschließend werden auf dieser Basis passende Suchergebnisse über die Serviceschnittstellen der ausgewählten Zielsysteme extrahiert und schließlich im Portal in einer Suchergebnisseite übersichtlich angezeigt /1/.

In einem ersten Schritt zur Spezifikation geeigneter Serviceschnittstellen wurden im Rahmen des SUI-Projektes Inhaltsangebote und Zugriffsschnittstellen ausgewählter Zielsysteme analysiert und darauf aufbauend eine Spezifikation zur formalen Beschreibung von Zielsystemen auf Basis des OpenSearch-Description-Standards /3/ entwickelt. Im Rahmen der aktuellen MAF-UIS-Kooperation Phase I sollten dann vor allem CMS-basierte Zielsysteme, wie der Themenpark Umwelt und das Fachdokumentenmanagementsystem FADO als Zielsysteme in die SUI-Suche eingebunden werden. Hierzu wurde für den Themenpark Umwelt /4/ zunächst eine SUI-kompatible Serviceschnittstelle entworfen und implementiert, die derzeit durch flexible Konfigurationsmechanismen so generalisiert wird, dass sie sich auf beliebige Web-Genesis basierende Systeme, wie z.B. FADO, übertragen lässt.

Der Themenpark Umwelt /5/ eignet sich deshalb gut als erstes Entwicklungssystem für die CMS Service API, da die verschiedenen Arten und Typen von Informationen im Themenpark in Bezug auf verschiedenste Inhalts- und Medienklassen typisiert und stark strukturiert sind. Weiter gibt es im Themenpark Inhaltsobjekte mit Raumbezug. Daher lassen sich alle Facetten der spezifizierten Service-API auch durch Themenpark-Inhalte abbilden und testen.

Die für CMS-Systeme spezifizierte Serviceschnittstelle folgt dem Prinzip von RESTful-Services (siehe Kapitel 2), wobei als Rückgabeformat für Suchabfragen ein AtomFeed-Format als Containerformat vorgesehen ist. Das AtomFeed-Format erlaubt es dabei, Feed-Einträge durch beliebige Einschübe aus anderen XML-Namespaces zu ergänzen. Auf diese Weise lassen sich Einträge z.B. um GeoRSS zur Angabe des Raumbezugs, MediaRSS zur Einbindung von Medien oder OpenSearch-Format zur Ergänzung von Suchinformationen erweitern.

Weitere Details der CMS Service-API und deren Implementierung am Beispiel des Themenparks werden in den folgenden Kapiteln näher erläutert.

## 2. RESTful Services

REST (Representational State Transfer) ist ein Architekturprinzip für verteilte, Internet-basierte Service-orientierte Anwendungen, das erstmalig in der Dissertation „Architectural Styles and the Design of Network-based Software Architectures“ von Roy Fielding beschrieben wurde [6]. Roy Fielding, der bereits als Student an der Standardisierung des HTTP-Protokolls mitgearbeitet hatte, ging dabei von der Überlegung aus, wie eine solche Service-orientierte Architektur beschaffen sein müsste, damit sie den grundlegenden Prinzipien des Web folgt und möglichst kompatibel zur bereits existierenden Struktur von Internetdiensten basierend auf dem HTTP-Protokoll und anderen Webstandards ist.

Gemäß den REST-Prinzipien werden alle über das Internet ansprechbaren Dienste und Informationen als Ressourcen aufgefasst, denen unterschiedliche Repräsentationen zugeordnet sein können. Jede Ressource ist durch eine spezifische URL eindeutig identifizierbar. Wenn wir die Basis-URL `http://themenpark.iai.fzk.de/servlet/is` durch „<baseUrl>“ abkürzen, dann repräsentiert z.B. die URL „<baseUrl>/resource/schutzgebiete“ eindeutig die Ressource „Liste von Schutzgebieten“ im Themenpark, während die URL „<baseUrl>/resource/schutzgebiete/10027“ das einzelne Schutzgebiet mit der ID „10027“ im Themenpark identifiziert.

Ein REST-Service gibt als Ergebnis einer Anfrage typischerweise stark strukturierte Informationen zu der jeweiligen Ressource in einem dafür geeigneten technischen Format zurück. Das technische Format, in dem diese Informationen codiert sind, wird in der Regel durch „Content Negotiation“ bestimmt. Mit dieser Funktionalität des HTTP-Protokolls kann ein Client eines Services in den Metainformationen eines HTTP-Requests festlegen, dass er z.B. gerne „application/json“ oder „application/xml“ als technisches Rückgabeformat hätte.

Die Aktionsverben „GET“, „POST“, „PUT“ und „DELETE“ des HTTP-Protokolls implementieren bei RESTful-Services schließlich grundlegende „CRUD“-Operationen (Create, Read, Update und Delete) auf den Ressourcen, die durch die URLs definiert sind. GET entspricht dabei dem Lesen in Form der Anforderung einer Repräsentation der Ressource in einem geeigneten Format. DELETE löscht die angegebene Ressource. Eine PUT-Operation entspricht Create, d.h. dem Erzeugen einer Ressource. Die hierfür nötige Datenrepräsentation des neu zu erzeugenden Objektes wird dabei als Nutzlast des PUT-URL-Aufrufes vom Client zum Server übertragen. Ein POST-Aufruf entspricht schließlich einer Änderungsoperation. Hierbei werden entweder Informationsattribute einer bestehenden Ressource geändert oder z.B. ein neues Objekt zu einer „Liste von Objekten“ als Ressource hinzugefügt.

### 3. Generische Suchschnittstellen auf Basis von Feed-Formaten

Als ein erweiterbares Containerformat eignet sich das AtomFeed-Format sehr gut als Rückgabeformat für RESTful-Service basierte Suchschnittstellen. In einem Containerelement (in XML `<feed>`) gibt ein AtomFeed eine Liste von Resultatobjekten (in XML repräsentiert durch das `<entry>`-Tag) zurück, die durch einige Metadaten zum Feed selbst, wie z.B. Autor `<author>`, Titel `<title>`, Kategorie `<category>`, ID, URL und Datum der letzten Änderungen ergänzt werden (siehe Abbildung 1).

Da das AtomFeed-Format es erlaubt, beliebige weitere Informationselemente aus anderen Namensräumen an beliebiger Stelle den Feedgrundelementen hinzuzufügen, kann man die Basismetadaten des Feeds unter Nutzung des OpenSearch-Namensraums um OpenSearch-Informationselemente, wie die Angabe der Suchworte (`searchTerms` `<os:Query role="request" searchTerms="..." ...>`) oder die Anzahl der Suchtreffer (`os:totalResults`) erweitern. Die Beschreibungen der einzelnen Suchresultate in den `<entry>`-Elementen, die gemäß Atomformat zunächst nur generische Informationen wie Titel des Eintrags, Link auf den Eintrag, ID und zusammenfassende Beschreibung des Resultats enthalten, können ebenfalls mittels Fremdnamensräumen wie GeoRSS für Angaben zur Geolokation `<georss:point>` sowie MediaRSS für Einbettung eines Vorschaubilds `<media:thumbnail ...>` angereichert werden. Zur Ergänzung durch eigene, spezifische Informationsattribute kann man neue Namensräume definieren und den Feed dann durch die eigenen spezifischen Konstrukte ergänzen.

Folgender Quelltext-Schnipsel einer AtomFeed-Rückgabe der Themenpark Serviceschnittstelle zeigt noch einmal die wesentlichen AtomFeed-Elemente und Elemente von Fremdnamensräumen, wie OpenSearch (Präfix `os`), GeoRSS (Präfix `georss`) und MediaRSS (Präfix `media`) in der Übersicht:

```
1 <feed xmlns="http://www.w3.org/2005/Atom" xmlns:os="http://a9.com/-/spec/opensearch/1.1/">
2   <title type="text">Feed von themenpark.iai.fzk.de</title>
3   <updated>2012-05-24T12:30:15.936Z</updated>
4   <author>
5     <name>WebGenesis-Autor</name>
6   </author>
7   <id>http://themenpark.iai.fzk.de/servlet/is/resource/erlebnisorte?count=5&page=1&q=natur&require=geo&require=media</id>
8   <link href="http://themenpark.iai.fzk.de/servlet/is/resource/erlebnisorte?count=5&page=1&q=natur&require=geo&require=media"
9     rel="self" />
10  <os:Query role="request" searchTerms="natur" totalResult="29" count="5" startIndex="1" startPage="1" />
11  <os:totalResults>29</os:totalResults>
12  <os:itemsPerPage>5</os:itemsPerPage>
13  <os:startIndex>1</os:startIndex>
14  <subtitle type="text">5 Feed-Einträge der Kategorie Erlebnisort; 29 gesamt</subtitle>
15  <category term="Erlebnisort" />
16  <entry xmlns:georss="http://www.georss.org/georss" xmlns:media="http://search.yahoo.com/mrss/">
17    <title type="html">Lotharpfad – Der Lotharpfad (Ldkr. Freudenstadt)</title>
18    <link href="http://themenpark.iai.fzk.de/servlet/is/24210/?path=7160;24157;" />
19    <id>http://themenpark.iai.fzk.de/servlet/is/24210/?path=7160;24157;</id>
20    <summary type="html">&lt;table cellpadding="5"&gt;&lt;tr valign="top"&gt;&lt;td&gt;&lt;img
21      src="/servlet/is/24211/_THUMB_lotharpfad-1.jpg?command=downloadContent&filename=__THUMB_lotharpfad-1.jpg" width="130"
22      height="173" alt="Lotharpfad" /&gt;&lt;td&gt;Der Lotharpfad, ein Walderlebnispfad auf einer naturbelassenen
23      Sturmwurffläche am Schliffkopf im Nordschwarzwald. Machen Sie sich selbst ein Bild von den Kräften der Natur. &lt;/td&gt;&lt;tr&gt;&lt;/table&gt;</summary>
24    <category term="Erlebnisort" />
25    <updated>2010-01-18T12:25:32.000Z</updated>
26    <georss:point>48.5064 8.2231</georss:point>
27    <media:content>
28      <media:title type="plain">Lotharpfad</media:title>
29      <media:thumbnail url="http://themenpark.iai.fzk.de/servlet/is/24211/_THUMB_lotharpfad-1.jpg?command=downloadContent&filename=__THUMB_lotharpfad-1.jpg" />
30    </media:content>
31  </entry>
32 </feed>
```

Abbildung 1: Beispielcode - Rückgabe der Suchergebnisse eines Service-API Aufrufes in AtomFeed-Format



{category} SUI-Klasse (Objektart) der gesuchten Ressourcen (Objekte), also z.B.:

- Umweltobjekte (Moor, Geotop, Schutzgebiet)
- Erlebnisorte

Bsp.: <baseURL>/resource/geotope // Liste aller Geotope

{id} WebGenesis-ID eines konkreten Themenpark-Eintrags der jeweiligen Informationskategorie

Bsp.: <baseURL>/resource/erlebnisorte/7141 // Erlebnisort mit ID 7141.

Über vorgegebene URL-Parameter, wie sie in Tabelle 1 zusammengestellt sind, lässt sich die Suche weiter einschränken und das genaue Rückgabeformat beeinflussen.

URL-Parameter	entspricht OpenSearch-Parameter <sup>1</sup>	Wert	Bemerkung	Beispiel/e
q	searchTerms	Text	URL-kodierter Suchbegriff	?q=naturschutzzentrum ?q=fl%E4che (fläche)
bbox	-	Zahl, Zahl	Bounding Box (linke untere und rechte obere Ecke)	?bbox=VAL1,VAL2
count	itemsPerPage	Zahl		?count=20
page	startPage	Zahl	nur in Kombination mit Parameter count sinnvoll	?count=5&page=3
require	-	Liste der Schlüsselworte „media“ oder „geo“		?require=geo ?require=media ?require=media&require=geo

**Tabelle 1: Parameter für die Suchanfragen**

In der ersten Version der API ist dabei zunächst spezifiziert, dass der thematische Bezug von Objekten neben der „Inhaltsklassenangabe“ durch weitere Freitextsuchbegriffe über den Parameter q (bzw. searchTerms) weiter eingeschränkt werden kann. Für Systeme, die eine Suche über Raumbezug erlauben, ist es weiter vorgesehen, dass man über Parameter den Ortsbezug genauer spezifizieren kann. Da das SUI-Portal bereits verschiedene Ortsbezugsbegriffe auflösen kann, ist hier zur Zeit nur der einfachste Fall der Ortseinschränkung über Angabe einer Bounding Box mit dem Parameter „bbox“ für die Implementierung vorgesehen. Weitere Möglichkeiten der Ortsangabe im Suchaufruf können bei Bedarf implementiert werden.

Mit den Parametern count und page kann man weiter ein Blättern durch die Ergebnisliste einer Suche implementieren. Schließlich kann man über die Angabe eines require-Parameters festlegen, dass man nur an Suchergebnissen interessiert ist, die einen Medienbezug oder Raumbezug (oder beides gleichzeitig) besitzen.

<sup>1</sup> [http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch\\_1.1\\_parameters](http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_1.1_parameters)

Auch Parameterkombinationen sind möglich, wie folgender Aufruf, der als Ergebnis den Beispielcode in Abbildung 1 liefert, verdeutlicht:

```
http://themenpark.iai.fzk.de/servlet/is/resource/erlebnisorte?count=5&page=1&q=natur&require=geo&require=media
```

Fehlerhafte Anfragen (durch nicht existierende oder ungültige IDs, unbekannte bzw. ungemappte Kategorien, keine Suchergebnisse für Suchbegriff, ...) bzw. ungültige Parameterkombinationen liefern einen HTTP-Fehlercode (z.B. 404 für „Not Found“) zurück.

## 4.1 Implementierung der Service-API

Die technische Umsetzung der RESTful-Schnittstelle für WebGenesis-basierte Systeme nutzt die Open-Source-Java-Frameworks Jersey<sup>2</sup> sowie Apache Abdera<sup>3</sup>. Jersey stellt die Referenzimplementierung der Java-API für RESTful Web Services (JAX-RS) dar. Apache Abdera implementiert das Atom-Format, Erweiterungen wie GeoRSS, MediaRSS und OpenSearch sind ebenfalls enthalten. Beide Frameworks lassen sich mit geringem Aufwand in ein WebGenesis-Basissystem integrieren, ohne dass hierbei die Kompatibilität bereits vorhandener Funktionalitäten verloren geht.

Die Businesslogik der technischen Implementierung muss dabei die in einem URL-Aufruf enthaltenen Suchparameter auf eine interne WebGenesis-Suche abbilden, die dann zur Suchanfrage passende WebGenesis-Objekte zurückliefert. Die Liste der zurückgelieferten Objekte kann im nächsten Schritt unter Nutzung der Abdera-API in das AtomFeed-Format umgewandelt und zum Client zurückgeschickt werden. Bei diesem allgemeinen Implementierungsszenario ergeben sich eine Reihe von Problemen, die durch die Implementierung möglichst generisch gelöst werden müssen:

1. Wie kann ein Mapping zwischen „Inhaltsklassen“ (z.B. „category“-Angaben) in der Aufruf-URL auf WebGenesis-Eintragsobjekte erfolgen, die tatsächlich Inhaltsobjekte von diesem speziellen Typ darstellen?
2. Wie kann eine räumliche Suche (möglichst) generisch implementiert werden?
3. Wie bedient man Anforderungen nach spezifischen Medien, z.B. nach Bildern?

Problem 1 impliziert eine Abbildung von RESTful-URLs auf Ergebnislisten von konkreten WebGenesis-Eintragsobjekten. Eine solche wird für jedes WebGenesis-System anwendungsspezifisch per Konfigurationsdatei hinterlegt. Für diese Konfiguration wurde ein Satz von Abbildungsregeln entworfen, über die man

- einzelne bestimmte Einträge oder Listen von bestimmten Einträgen,
- alle Einträge, die zu einer bestimmten WebGenesis-Kategorie gehören, ggf. gefiltert durch einen gegebenen SQL-Ausdruck,
- alle Einträge, die einem bestimmten Ontologiekonzept entsprechen,
- alle Ober- bzw. Untereinträge eines bestimmten Eintrags (ggf. mit Begrenzung der Schachtelungstiefe),
- alle WebGenesis-Einträge, die per Relation (eines bestimmtem Relationstyps) mit einem bestimmten WebGenesis-Eintrag verbunden sind,

---

<sup>2</sup> <http://jersey.java.net/>

<sup>3</sup> <http://abdera.apache.org/>



- alle WebGenesis-Einträge, die per Relation (eines bestimmtem Relationstyps) mit einem WebGenesis-Eintrag verbunden sind, der wiederum einem bestimmten Ontologiekonzept entspricht,

einer bestimmten Inhaltsklasse zuordnen kann. Darüber hinaus sind auch Kompositionen dieser Regeln möglich:

- Konjunktion (Schnittmenge)
- Disjunktion (Vereinigungsmenge)
- Subjunktion (Differenzmenge)

Die durch diese Abbildungsregeln beschriebenen Listen von Ergebniseinträgen werden im Allgemeinen anschließend durch die Anwendung von Filter-Parametern reduziert, z.B. wenn eine Bounding Box (bbox) für Einträge mit Geobezug angegeben ist.

Beim zweiten Problem, der räumlichen Suche, muss man zwischen existierenden Implementierungen, die bereits einen Geobezug besitzen, und künftigen, generischen Implementierungen differenzieren. Da Themenpark-Objekte nicht nur stark typisiert sind, sondern die WebGenesis-Eintragsklassen für Fachobjekte wie Erlebnisorte auch Attribute zur Angabe des Raumbezugs (Felder für Longitude- und Latitude-Angaben) besitzen, ist die Einbindung von GeoRSS im Feed für solche Objekte im Themenpark möglich. Flächenhafte Themenpark-Umweltobjekte (Moore, Geotope und Schutzgebiete) haben zwar auch einen räumlichen Bezug, hier werden im Themenpark aber nur Object-ID und OAC (Objektartencode) vorgehalten und über einen RIPS<sup>4</sup>-Service der LUBW die Bounding Boxen bzw. Koordinaten dieser Objekte abgerufen. Dies ermöglicht es ebenfalls für solche Objekte einen Raumbezug herzustellen. Weiter kann hiermit auch die raumbezogene Suche nach Objekten im Themenpark unter Angabe einer Bounding Box implementiert werden. Leider ist die Themenpark-spezifische Rückgabe von GeoRSS-Informationen und die raumbezogene Suche in anderen CMS-Systemen des UIS bislang nicht möglich, da diese Systeme Autoren keine Möglichkeiten zur Angabe von Raumbezug für Objekte bereitstellen. Hier bietet sich für die Zukunft aber die Chance, solche Möglichkeiten standardisiert für alle WebGenesis-Systeme bereitzustellen. Hierfür sieht die Konfiguration der Service-API Implementierung eine Beschreibung des Zugriffs auf die notwendigen Attribute von erweiterten WebGenesis-Kategorien, ggf. auch ausgedrückt in Datatype-Properties des Ontologiesystems, vor.

Der Medienbezug (Problem 3) lässt sich in WebGenesis-Systemen vergleichsweise einfach herstellen. Hierzu muss bei den Inhaltsobjekten nur geprüft werden, ob diesen Medienobjekte entweder über Relationen auf Medienobjekttypen (im Fall des Themenparks) oder durch Speicherung von Mediendateien im Dateibereich eines Inhaltobjektes Mediendateien (Bilder, Videos, etc.) zugeordnet sind. Referenzen auf solche verknüpften Medienobjekte können dann im MediaRSS-Format im AtomFeed-Stream zurückgegeben werden.

## 5. Fazit und Ausblick

Im Rahmen der Arbeiten zu SUI wurde für den Themenpark Umwelt eine erste Implementierung einer SUI-konformen Service-API für WebGenesis-basierte Systeme vorgenommen.

---

<sup>4</sup> <http://www.lubw.baden-wuerttemberg.de/servlet/is/16154/>

Aufgrund der starken semantischen Strukturierung des Themenparks lassen sich dabei über die Service-API Themenpark-Inhaltsobjekte passend zu vorgegebenen Inhaltsklassen abrufen und durch Zusatzinformationen wie Raumbezug oder Medienbezug ergänzen.

Diese Implementierung lässt sich wie angedeutet per Konfiguration auch auf andere WebGenesis-basierte UIS-Systeme wie FADO oder die Homepage-Systeme von UM und LUBW übertragen, indem über generische Mappingregeln eine Zuordnung zwischen SUI-Konzeptklassen und WebGenesis-Objekten vorgenommen wird. Idealerweise würden zukünftige WebGenesis-Systeme SUI-Inhaltsobjektklassen bereits als Standard-Ontologie in allen UIS-WebGenesis-Systemen vorhalten. Autoren könnten dann ihre Inhaltsobjekte direkt solchen Klassen zuordnen. Auf diese Weise ließe sich auch ein standardisierter Raumbezug für alle UIS-WebGenesis-Systeme realisieren. Hierzu würde man in der Standard-UIS-Ontologie den Raumbezug über Konzeptklassen abbilden, denen entsprechende Eigenschaften, wie Longitude und Latitude zugeordnet sind, die ein Autor nach Zuordnung dann auch in den Autorenformularen evtl. unter Nutzung eines Geodienstes setzen kann.

## 6. Literatur

- /1/ Bügel, U. et al. (2011): SUI für Umweltportale – Entwurf und prototypische Implementierung einer Architektur für die semantische Suche im Portal Umwelt-BW. In: Mayer-Föll, R., Ebel, R., Geiger, W.; Hrsg.: F+E Vorhaben KEWA – Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen, Phase VI 2010/11, Karlsruher Institut für Technologie, KIT Scientific Reports 7586, S. 21-32.
- /2/ Schlachter, T. et al. (2012): LUPO – Ein Baukasten für die Entwicklung von Umwelt- und Energieportalen. In diesem Bericht.
- /3/ OpenSearch-Spezifikation: <http://www.opensearch.org>, besucht am 29.05.2012.
- /4/ Grießmann, B. et al. (2011): Themenpark – Weitere Inhalte, Medien und Technologien beim Themenpark Umwelt. In: Mayer-Föll, R., Ebel, R., Geiger, W.; Hrsg.: F+E Vorhaben KEWA – Kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen, Phase VI 2010/11, Karlsruher Institut für Technologie, KIT Scientific Reports 7586, S. 129-136.
- /5/ Themenpark Umwelt: <http://www.themenpark-umwelt.baden-wuerttemberg.de>.
- /6/ Fielding, R. T. (2000): Architectural Styles and the Design of Network-Based Software Architectures, Dissertation, University of California, Irvine, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>, besucht am 01.06.2012.