

# Facettiertes Browsen von semantischen Daten auf einem mobilen Endgerät

Alexander Kleinen  
Institute for Computer Science  
University of Koblenz-Landau

## **Zusammenfassung**

In dieser Arbeit wird das MobileFacets System präsentiert, das ein bequemes facettiertes Browsen und Suchen von semantischen Daten auf einem mobilen Endgerät ermöglicht. Anwender bekommen in Abhängigkeit ihres lokalen Ortskontextes, weitreichende Informationen wie Orte, Personen, Organisationen oder Events dargeboten. Basierend auf der Theorie von Facetten, wird das facettierte Browsen zur Erkundung von strukturierten Datensätzen anhand einer Client Anwendung realisiert. Die Anwendung bedient sich dabei eines lokalen Servers, der für Anfragen der Clients, die Anbindung an externe Datenquellen und die Aufbereitung der strukturierten Daten zuständig ist.



## Diplomarbeit

vorgelegt von *Alexander Kleinen* im November 2010 zum Thema

*Facettiertes Browsen von semantischen Daten auf einem mobilen Endgerät*

zur Erlangung des Grades eines Diplom-Informatikers im Studiengang  
Informatik an der Universität Koblenz-Landau.

## Gutachter

*Prof. Dr. Steffen Staab*

*Dr. Ansgar Scherp*

Web Science and Technologies

Institut für Informatik

Fachbereich 4

Universität Koblenz-Landau

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine  
anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

*Koblenz, den 29. November 2010*

---

*Alexander Kleinen*



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Struktur der Arbeit . . . . .	2
<b>2</b>	<b>Szenarien</b>	<b>4</b>
2.1	Informationsbedarf im mobilen Bereich . . . . .	4
2.2	Szenario: Trivia . . . . .	6
2.3	Szenario: Point of Interest-Suche . . . . .	7
<b>3</b>	<b>Verwandte Arbeiten</b>	<b>9</b>
3.1	Facetten . . . . .	9
3.2	Taxonomien von Facetten . . . . .	11
3.3	Facettiertes Browsen . . . . .	13
3.4	Facetten-basierte Benutzeroberflächen . . . . .	14
3.5	Facetten-basierte Benutzeroberflächen im mobilen Bereich . . . . .	16
<b>4</b>	<b>Oberflächen-Entwurf</b>	<b>21</b>
4.1	Erste Entwurfsphase . . . . .	22
4.2	Zweite Entwurfsphase . . . . .	24
4.2.1	Szenario: POI-Suche . . . . .	24
4.2.2	Szenario: Trivia . . . . .	28
4.2.3	Verwendung von Tabs und Menüs . . . . .	30
4.2.4	Browsen über Facetten . . . . .	31
4.2.5	Darstellung der Facettenauswahl . . . . .	32
<b>5</b>	<b>Evaluation der Designstudie</b>	<b>35</b>
5.1	Vorgehen . . . . .	37
5.2	Testpersonen . . . . .	39
5.3	Aufgaben . . . . .	39
5.3.1	Szenarien POI-Suche & Trivia . . . . .	40
5.3.2	Detailfragen . . . . .	41
5.4	Auswertung und Analyse . . . . .	42
5.4.1	Diskussion des Szenarios: POI-Suche . . . . .	46
5.4.2	Diskussion des Szenarios: Trivia . . . . .	47
5.4.3	Diskussion der Detailfragen . . . . .	48
5.4.4	Weitere Anmerkungen . . . . .	51

<b>6</b>	<b>Anforderungsanalyse</b>	<b>53</b>
6.1	Funktionale Anforderungen . . . . .	53
6.1.1	Client . . . . .	53
6.1.2	Backend . . . . .	56
6.2	Nicht-funktionale Anforderungen . . . . .	57
<b>7</b>	<b>Architektur und Entwurf</b>	<b>58</b>
7.1	Komponenten . . . . .	59
7.1.1	Client . . . . .	59
7.1.2	Server . . . . .	61
7.1.3	Datenrepositories . . . . .	62
7.2	Suchen von Facetten und Instanzen . . . . .	64
7.3	Kommunikation . . . . .	69
<b>8</b>	<b>MobileFacets Prototyp</b>	<b>74</b>
8.1	Iterative Implementierung und Benutzerfeedback . . . . .	74
8.2	Client . . . . .	77
8.2.1	Suchen, Browsen und Interaktion mit Facetten und Instanzen . . . . .	79
8.2.2	Kartenansicht, Menü und Interaktion mit POIs und Events . . . . .	83
8.2.3	Ergebnisliste . . . . .	86
8.2.4	Fotoansicht . . . . .	88
8.2.5	Detailansicht . . . . .	89
8.2.6	Anwendungs-Einstellungen . . . . .	91
8.3	Server . . . . .	92
8.3.1	Prozesskette . . . . .	92
8.3.2	Integration der Datenrepositories und Extraktion von Facetten . . . . .	97
<b>9</b>	<b>Evaluation von MobileFacets</b>	<b>107</b>
9.1	Vorgehen . . . . .	107
9.2	Testpersonen . . . . .	107
9.3	Aufgaben . . . . .	108
9.4	Auswertung und Analyse . . . . .	111
9.4.1	Aufgabe A: POI-Browsen . . . . .	112
9.4.2	Aufgabe B: Personen Suche . . . . .	114
9.4.3	Aufgabe C: POI-Suche . . . . .	116
9.4.4	Aufgabe D: Event-Browsen . . . . .	119
9.4.5	Aufgabe E: Event-Suche und weiteres Browsen . . . . .	121
9.4.6	Angemessenheit der Arbeitsschritte . . . . .	123

9.4.7	Intuition . . . . .	124
9.4.8	Anwendungsunterstützung . . . . .	125
9.4.9	Erwartungskonformität . . . . .	127
9.4.10	Weitere Anmerkungen . . . . .	127
<b>10</b>	<b>Schlussfolgerung</b>	<b>131</b>
<b>11</b>	<b>Zusammenfassung und Ausblick</b>	<b>132</b>
	<b>Literaturverzeichnis</b>	<b>135</b>
	<b>Anhang</b>	<b>141</b>

# 1 Einführung

## 1.1 Motivation

Das facettierte Browsen komplexer Daten hat sich im Laufe der letzten 30 Jahre zu einer beliebten Technik entwickelt. Es unterscheidet sich von herkömmlichen Suchmechanismen, bei dem die Eingabe von Schlagwörtern in einem Suchfeld erfolgt. Des Weiteren gewinnen technische Verfahren zum facettierten Navigieren und Browsens innerhalb eines Informationsraums zunehmend an Bedeutung [1–3].

Es gibt bereits eine umfangreiche Literatur, die sich speziell mit dem facettierten Suchen und Browsen von komplexen, strukturierten Daten befasst. Die bestehenden Ansätze hierzu verlangen eine halb automatische Generierung von hierarchisch facettierten Metadaten über gut strukturierte Informationsräume [2] oder Methoden zur Ermittlung gewichteter Facetten aus verknüpften Daten [4]. Eine Lösung für das facettierte Browsen beliebiger und semantischer Daten auf einem mobilen Endgerät anzubieten, ist für diese Arbeit eine überaus wichtige Zielsetzung.

Aufgrund der Limitationen von mobilen Endgeräten, wie die der Display-Größe oder der eingeschränkten Interaktionsmöglichkeiten, bedarf es einer gut durchdachten Benutzeroberfläche. Neben dem Design muss festgelegt werden, welche Daten benötigt werden. Je nach Anwendungskontext und Qualität der Daten müssen unterschiedliche Anforderungen hinsichtlich der Gestaltung der Benutzeroberfläche gestellt werden. Bei einer Anwendung für das facettierte Suchen und Browsen über semantische Daten bedarf es einer flexiblen Benutzeroberfläche, in der Anwender keine Kenntnisse über die Struktur der unterliegenden Daten benötigen [5, 6]. Ein typisches Suchfeld zur Eingabe eines Schlüsselwortes ist in seiner Funktionalität nicht ausreichend genug, um eine Navigation und Erkundung von großen und komplexen Datensätzen wie Online-Kataloge oder digitale Bibliotheken vorzunehmen. Statt dessen sind Facetten-basierte Suchmechanismen geeignete und bewährte Techniken zur Erkundung solcher Datensätze [3].

Die Anzahl von mobilen Endgeräten mit Zugang zum Internet, wie beispielsweise Smartphones, ist in den letzten Jahren rapide angestiegen. Es wird vorausgesagt, dass die Anzahl der Internetzugriffe mittels eines mobilen Endgerätes, die der Zugriffe von normalen PCs oder Laptops weltweit übersteigen wird. Es wird erwartet, dass die gesamte Anzahl der PCs im Jahre 2013 auf 1.78 Milliarden ansteigt. Im Gegensatz dazu steigt die Anzahl der internetfähigen mobilen Endgeräte weit über 1.82 Milliarden [7]. Mobile Endgeräte bekommen immer leistungsfähigere Hardware, wie hochauflösende Kameras, großzügige Speichermöglichkeiten und schnelle CPUs. Weiterhin

steigen die Kommunikationsmöglichkeiten solcher Geräte. Mit Technologien wie 3G, GPS und WLAN, ausreichenden PIM<sup>1</sup>-Funktionalitäten und einer Vielzahl von Applikationen werden mobile Endgeräte kontinuierlich vielseitiger.

Aktuelle Techniken und Anwendungen im mobilen Bereich sind nicht hinreichend gut, um große Menge an semantischen Daten zu verwalten und verarbeiten. Darüber hinaus ist nicht nur die Verwaltung und Verarbeitung dieser Daten wichtig, sondern auch Techniken zur Erkundung von strukturierten Datensätzen. Gerade im mobilen Anwendungskontext stellt sich die Frage, wie gestaltet man flexible Benutzerschnittstellen um ortsgebundene Informationen sinnvoll darzustellen. Benutzer interessiert es, welche Informationen sich in der unmittelbaren Umgebung befinden [8]. Dazu können Information zählen, wie interessante Kirchen, Museen oder andere Sehenswürdigkeiten, die eine Geokoordinate besitzen. Darüber hinaus ist es interessant zu wissen, welche Personen einen semantischen Zusammenhang mit der Umgebung des Anwenders haben oder welche Events (Konzerte, Parties, Ausstellungen) in der nächsten Zeit stattfinden.

In Anbetracht der Tatsache, dass es eine steigende Zahl von internetfähigen mobilen Endgeräten gibt, müssen neue Ansätze für flexible und angemessene Benutzerschnittstellen erforscht werden. Keine der zurzeit verfügbaren Lösungen erfüllt die folgende Anforderung: Ein generische Benutzerschnittstelle für mobile Endgeräte für eine flexible Erkundung von strukturierten Daten.

## 1.2 Struktur der Arbeit

Die Arbeit gliedert sich grob in sieben Teilabschnitte: Im ersten Abschnitt werden die für diesen Kontext wichtigen Szenarien definiert. Die Szenarien beschreiben typische Anwendungsfelder im mobilen Umfeld. Im Anschluss wird eine ausführliche Definition von Facetten und Beispiele von gegenwärtigen Anwendungen geliefert. Im Zuge dieser Arbeit wird danach eine prototypischer Oberflächenentwurf erstellt, welcher die Grundlage des MobileFacets System bildet. Aus den Designentwürfen und den Szenarien wird in Teilabschnitt vier eine Designstudie erstellt, die die Ergebnisse des Oberflächenentwurfs validieren und die Grundlage für die Implementation des System darstellen. Aus dieser Designstudie werden die funktionalen sowie nicht-funktionalen Anforderungen an das System definiert. Im Anschluss wird ausführlich auf die Architektur und die Implementierung des System eingegangen. Funktionalitäten werden anhand von Screenshots im Detail er-

---

<sup>1</sup>PIM = Personal Information Manager, (engl. für persönliche Informationsverwaltung)

klärt. Außerdem wird die Server-Komponente erläutert, die das Backend des Systems bildet.

Der siebte Abschnitt ist eine Evaluation des MobileFacets Prototypen, der zeigen soll, wie intuitiv sich das System bedienen lässt.

Kategorie	Beispiel	Absolute Häufigkeit
Trivia	Wie und wann starb Bob Marley?	18,5%
Directions	Die Route zum nächsten Restaurant	13,3%
Point of Interests	Zeig mir alle Sehenswürdigkeiten in der Stadt	12,4%
Friend Info	Wo sind Thorsten und Christian?	7,6%
Shopping	Wieviel kostet das neue Album von Muse?	7,6%

Tabelle 1: Kategorien der Informationsbedürfnisse. Nach auftretender Häufigkeit sortiert. [9]

## 2 Szenarien

### 2.1 Informationsbedarf im mobilen Bereich

Für diese Arbeit ist es interessant und von Bedeutung, welche generellen Informationsbedürfnisse Benutzer im mobilen Bereich haben. In [9] wurde eine Tagebuchstudie durchgeführt, die zum Ziel hatte, signifikante Kategorien von Informationsbedürfnissen zu bestimmen. In der Studie konnten aus insgesamt 421 Tagebucheinträgen 16 verschiedene Kategorien herausgefunden werden. Die folgenden fünf Kategorien sind die häufigst auftretenden Informationsbedürfnisse [9].

**Trivia** Diese Klasse beschreibt typische Informationsbeschaffungen. Sei es von einer einzelnen Person oder einer Gruppe von Personen, die eine bestimmte Information beschaffen möchten. Ein einfaches Beispiel wäre die Suche nach einer bestimmten Information im Internet: Wie und wann starb Bob Marley? Der Aspekt der Informationsbeschaffung lässt sich weiterhin auf eine Gruppe von Personen ausweiten, in der über einen bestimmten Sachverhalt diskutiert wird, jedoch aber der korrekte Informationsaspekt im Internet nachgeschaut werden muss. Beispielsweise treffen sich Anbieter und Kunden eines Unternehmens in einem Restaurant und diskutieren über die verschiedensten Sachverhalte. Aus dieser sozialen Konversation entstehen Fragen, die im besten Falle mithilfe einer Suche im Internet beantwortet werden können. Diese Fragen können einen gewissen Bezug auf den jeweiligen Ortskontext haben. Wie viel Einwohner hat die Stadt? Wie hoch

ist der Fernsehturm? Das sind nur einige Beispiele, die einen bestimmten Ortsbezug besitzen. Die Beantwortung der Fragen kann anhand eines mobilen Endgerätes eines Beteiligten erledigt werden.

Im mobilen Bereich stellen sich Hürden, welche die Informationsgewinnung erschweren. Unzureichende oder schlechte mobile Internetanbindungen, sowie umständliche Interaktion mit mobilen Endgeräten sind nur einige Beispiele. Je nach Größe und Typ des Gerätes kann sogar eine einfache Google-Suche nach der obigen Frage zur Höhe des Fernsehturmes umständlich werden.

**Directions** Hier handelt es sich um die zweithäufigste Aufgabe, die im mobilen Bereich eine Rolle spielt. Ein Benutzer möchte von seinem Standort A zu einem bestimmten Standort B - und das möglichst genau und visuell ansprechend dargestellt. *Directions* lässt sich hier unter dem Begriff der mobilen Navigation einordnen.

**Point of Interest** POIs sind Orte, die für Nutzer eine Bedeutung haben. Sie stehen an dritter Stelle der häufigsten Aufgaben im mobilen Bereich. Ein einfaches Beispiel wäre, wenn ein Benutzer in einer fremden Stadt unterwegs ist und die lokalen Gegebenheiten kennenlernen möchte. Sei es die Auflistung der nächsten Restaurants und Bars oder kulturellen Informationen, die die Stadt zu bieten hat. POIs werden oft auf einer Karte visualisiert.

**Friend Info** Das ist eine Kategorie, in dem der soziale Aspekt eine wichtige Rolle spielt. "Wo befindet sich gerade meine Schwester?" ist nur ein Beispiel, welches hier aufgeführt werden kann.

**Shopping** Diese Kategorie deckt den direkten kommerziellen Part der Informationsbeschaffungen im mobilen Bereich ab. Hier möchte der Benutzer gezielt Informationen zu Preisen von Produkten erfahren. Weiterhin steht auch der Vergleich von Preisen aus verschiedenen Läden eine wichtige Rolle.

Für diese Arbeit spielen diese fünf Kategorien eine wichtige Rolle. Die Relevanz der Informationsbeschaffung sollte sich auch im Aspekt des Designs und der Implementierung widerspiegeln. Der mit Abstand häufigste Informationsbedarf liegt meist in einfachen Fragestellungen (Kategorie *Trivia*: "Wie und wann starb Bob Marley?"), wie in der Tabelle 2.1 zu sehen ist. *Directions* spielt im mobilen Bereich eine immer größere Rolle. Neuartige Geräte verfügen oft über umfangreiche Kommunikation- und Ortungsfunktionen, die es mobilen Applikationen leicht machen eine Navigation zu einem bestimmten Ziel zur Verfügung zu stellen.

In den folgenden zwei Abschnitten sind zuerst typische Fragestellungen in den Bereichen *Trivia*, *Directions* und *Point of Interests* aufgeführt. Aus diesem Mix an Fragestellungen wurden zwei Szenarien aufgestellt, in der diese typischen Fragen sinnvoll abgebildet werden. Es handelt sich dabei um Szenarien, die einen realistischen Hintergrund abbilden, zum anderen sind sie so gestaltet, dass sie einen Ausschnitt der zuvor angesprochenen Fragen beinhalten. Zu beachten ist, dass das Endsystem nicht nur auf diese Szenarien ausgerichtet ist. Die Kategorien *Friend Info* und *Shopping* können je nach verwendetem Datentopf auch für das Endsystem interessant und realisierbar sein. Die Szenarien stellen vielmehr ein Ausschnitt der gesamten Funktionalitäten aus Abschnitt 6 dar.

## 2.2 Szenario: Trivia

Die folgenden Fragen repräsentieren eine Auswahl aus dem Bereich Trivia (siehe Tabelle 2.1).

- Ich diskutiere gerade mit meinen Freunden über verschiedene Themengebiete
  - Wie viele Einwohner hatte Berlin im Jahre 1972? Wie viele Einwohner hat Koblenz in diesem Jahr?
  - Wie hoch ist der Berliner Fernsehturm? Was ist mit dem Koblenzer Fernmeldeturm?
  - Woran starb Bob Marley und wann war das genau? Und wie ist überhaupt John Lennon gestorben?
- Szenario

*Paul ist Mitarbeiter der Firma Software AG mit Sitz in Hamburg. Er ist sehr oft auf Dienstreisen, wie auch an diesem Tag, wo er geschäftlich mit ein paar Kollegen in Berlin ist, um einige Präsentationen bei einem potentiellen Kunden zu halten. Nachdem die Präsentationen sehr vielversprechend gelaufen sind, möchte Paul mit seinen Kollegen mittags in einem Lokal etwas essen. Bei einem ausgelassenen Gespräch kommt die Frage auf, wie hoch der Fernsehturm von Berlin doch sei. Jeder kennt ihn, doch keiner von den Kollegen weiß die korrekte Höhenangabe des Turmes. Nachdem die Frage geklärt ist, erinnert sich Paul zudem, dass er nächstes Wochenende auf einem Messebesuch der Koblenzer LocalBit ist. Paul fragt sich, wie hoch doch der Koblenzer Fernsehturm im Gegensatz zum Berliner Turm ist...*

In diesem Szenario spielt der Suchkontext des aktuellen Ortes eine wichtige Rolle. Jedoch wird das Szenario um den Aspekt der unabhängigen Ortssuche erweitert. Nachdem Paul Informationen zum nächstgelegenen Fernsehturm einsehen konnte, löst er sich von dem aktuellen Suchort Berlin ab und ändert den Suchkontext in das entfernte Koblenz. Dieses Szenario spiegelt eine typische Konversation wieder, in der bei vorhandenen Fragen neue Fragen aus dem jeweiligen Konversationskontext auftreten können. Dabei spielt der aktuelle Gesprächskontext (Höhe des Turmes) eine Rolle. Weiterhin lenkt die aktuelle Position (hier in der "fremden" Stadt Berlin) der Gesprächsteilnehmer maßgeblich die diskutierten Themengebiete.

## 2.3 Szenario: Point of Interest-Suche

Die folgenden Fragen repräsentieren eine Auswahl aus dem Bereich der Suche nach Point of Interests (POI).

- Ich möchte Informationen zu meiner unmittelbaren Umgebung
  - Wo befindet sich die nächste Pizzeria?
  - Gibt es Bars und Kneipen in meiner Umgebung?
  - Ich suche Informationen zu historischen Gebäuden in der Stadt.
  - Ich möchte Flickr-Bilder, die in meiner unmittelbaren Umgebung liegen, einsehen.
  - Ich möchte generell die POIs in meiner Umgebung einsehen.
  - Ich möchte die verfügbaren Facetten in meiner Umgebung einsehen.
  - Welche Events finden in der nächsten Zeit statt?

- Szenario

*Der IT-Berater Paul Müller von der Software AG ist ein Wochenende später geschäftlich in Koblenz mit der Bahn angekommen und möchte die IT-Messe LocalBit besuchen. Da Paul noch nie in Koblenz war, möchte er vom Bahnhof aus die Position der Rhein-Mosel-Halle erfahren, da dort die Messe stattfinden wird. Nachdem er erfolgreich den Weg zur Messehalle gefunden hat und schließlich den Messebesuch hinter sich gebracht hat, möchte Paul mehr über die Stadt Koblenz erfahren. Seine Rückreise ist erst Montags, so hat er noch den Sonntag Zeit, um seinen Durst nach kulturelle Informationen zu stillen. Er interessiert sich unter anderem für das Deutsche Eck und die Festung*

*Ehrenbreitstein und erfährt zudem, dass im folgenden Jahr die Bundesgartenschau in Koblenz stattfinden wird. Nachdem sein Durst nach kulturellen Informationen gestillt ist, knurrt sein Magen. Am Deutschen Eck möchte Paul die nächstgelegenen Restaurants ausfindig machen. Seine Wahl fällt auf ein italienisches Restaurant, welches nur einige Gehminuten entfernt liegt. Zufrieden und gesättigt macht Paul noch einen Abstecher zu einem nahegelegenen Café, bevor er sich auf den Weg zu seinem Hotel macht...*

Das Szenario spiegelt eine typische Suche nach lokalen Informationen wieder. Generell behandelt das Szenario typische Sightseeing Aspekte, sowie die Suche nach Point of Interests. Aus diesem Szenario lassen sich konkrete Arbeitspakete ableiten, die zum einen für das Oberflächendesign und zum anderen für die Durchführung der Evaluation eine wichtige Rolle spielen. Anhand des Szenarios kann man zwei voneinander unabhängige Fragestellungen von Paul sehen. Die Gewinnung von kulturellen Informationen auf der einen Seite, auf der anderen die Suche nach nächstgelegenen Lokalitäten um den Hunger zu stillen. Diese Aufteilung der unterschiedlichen Informationsbeschaffung spiegelt sich auch in der Durchführung der Evaluation wieder, die im Abschnitt 5 diskutiert wird.

## 3 Verwandte Arbeiten

Derzeit besteht ein Großteil der verfügbaren Suchinterfaces aus einem Suchfeld und einer Ergebnisliste, die typischerweise nach absteigender Relevanz sortiert ist. Dieses Suchparadigma zur Suche nach Informationen hat sich in einem großen Umfang durchgesetzt. Heutzutage benutzen viele Suchmaschinenanbieter wie Google, Yahoo oder Bing dieses bewährte Paradigma. Abgesehen von typischen Suchszenarien wie die "Suche nach Webseiten oder bestimmte Informationen auf Webseiten", bedarf es für die Erkundung und Navigation innerhalb von großen Informationskollektionen wie Online-Kataloge, digitale Büchereien oder semantische Daten einen anderen Ansatz.

Diese Informationskollektionen können durch hierarchisch angeordnete Datensätze kategorisiert sein. Jede Kategorie innerhalb dieser Hierarchie korrespondiert mit einer bestimmten Facette. Man stelle sich ein Beispiel einer digitalen Bücherei vor. Die "Faceted DBLP<sup>2</sup>" bietet die Möglichkeit durch gezieltes Auswählen einer Kategorie (Facette) die Suchmenge so weit einzuschränken, dass nur die relevanten Informationen angezeigt werden [10]. Facetten sind in dieser Bücherei beispielsweise das Datum der Publikation (1988 -1995), der Publikationstyp (Artikel, technischer Report) oder der Autor. Im Semantic Web Kontext können Facetten durch Prädikate in einer RDF Datenstruktur repräsentiert werden.

Es gibt eine Vielzahl an Literatur, die das Thema facettierte Browsen innerhalb von komplexen Datenstrukturen behandeln. In den folgenden Abschnitten wird zuerst eine Definition einer Facette erörtert. Im Anschluss werden unterschiedliche Taxonomien von Facetten diskutiert. Darauf hin werden Praxisbeispiele gegeben, die verschiedenen Ansätze des facettierte Browsens und Suchens im Desktop- sowie im mobilen Bereich zeigen.

### 3.1 Facetten

In der Literatur gibt es keine präzise und hinreichende Definition von Facetten. Das Wort *Facette* wird in unterschiedlichsten Kontexten verwendet und ungenau interpretiert [11]. Dennoch gibt es einige Charakteristiken, die in allen Definitionen identisch sind. Jede Facette besitzt einen Namen oder zumindest eine hinreichende Beschreibung [2], und jede Facette verweist auf eine Kategorie um Aspekte einer Informationskollektion zu beschreiben. Von Vickery [12] wurden Facetten anhand ihrer Eigenschaften analysiert. Er vergleicht das semantische Modell des menschlichen Gedächtnisses mit den Eigenschaften von Facetten [13] und konnte Facetten anhand der folgen-

---

<sup>2</sup><http://dblp.13s.de/>

den Kategorien definieren: *Attribute, Objekte, Orte, Prozesse, Eigenschaften, Substanzen, Zeiten* und *Empfänger*. Diese Facetten können in verschiedenen Anwendungsdomänen unterschiedlich benannt sein. Sein Ziel war es, eine Klassifikation von Facetten in essenzielle Subjekte vorzunehmen.

Darüber hinaus wird von Eyal Oren et al. [14] eine präzisere Definition einer Facette geliefert. Sie betrachten das facettenreiche Browsen als eine Zerschneidung von Informationsräumen und verwenden dabei die konzeptuellen Dimensionen dieser Daten. Diese konzeptuellen Dimensionen werden hier als Facetten definiert. In Bezug darauf, repräsentieren Facetten wichtige Informationscharakteristiken dieser Daten. Jede Facette kann durch mehrere Attribute eingeschränkt werden. Somit lässt sich anhand einer gezielten Auswahl dieser Attribute, die Ergebnismenge manipulieren. Diese Art der Facettentheorie lässt sich äquivalent auf die Navigation innerhalb von semi-strukturierten RDF Graphen übertragen. RDF Subjekte repräsentieren Informationsaspekte, RDF Prädikate werden als Facetten gesehen und RDF Objekte sind die Attribute [14].

Eine andere Ansicht ist, gegebene Klassen in Kategorien zu unterteilen. Beispielsweise lässt sich die Klasse *Person* in die Kategorien *Name, Alter, Beruf, Nationalität* und so weiter einteilen. Dabei wird der Begriff der Kategorie als Facette bezeichnet. Weiterhin kann man sagen, dass eine Facette eine Subkategorie einer Facette ist: *Karotte* ist eine Subfacette von *Gemüse* und ist eine Subfacette von *Zutat* [3] .

## Charakteristiken von Facetten

Die folgende Übersicht zeigt einige wichtige Charakteristiken von Facetten:

- Facetten besitzen einen Namen oder haben eine Beschreibung
- Facetten werden abhängig des Informationsraumes beschrieben [14]
- Facetten beschreiben Subjekte in unterschiedlichen konzeptuellen Dimensionen [3]
- Facetten sind in orthogonale Dimensionen strukturiert [15]
- Facetten sind Teil eines Informationsraumes und beschreiben eindeutig eine Klasse von Elementen [16]
- Facetten erlauben es effizient durch einen Informationsraum zu navigieren
- Mit der Verwendung von Facetten werden Informationsräume eingeschränkt.

## 3.2 Taxonomien von Facetten

In diesem Abschnitt geht es um die Taxonomien von Facetten. Das Wort Taxonomie kommt von dem altgriechischen *táxis* ('Ordnung') und *nomos* ('Gesetz') und taucht in seinem Ursprung als Klassifikation von lebenden Objekten auf. Aristoteles war der erste Taxonomist, der Organismen in Kategorien klassifizierte. So wurden beispielsweise Organismen in zwei Gruppen geteilt: Pflanzen und Tiere. Tiere wurden wiederum anhand ihrer Blutfärbung, rotes und nicht rotes Blut, klassifiziert [17].

In der heutigen Zeit wird der Begriff der Taxonomie allgemeiner aufgefasst. Man beschreibt mit einer Taxonomie ein hierarchisch angeordnetes Klassifikationssystem, wobei Objekte oder Abstraktionen in irgendeiner Form, wie beispielsweise in einem Baum, angeordnet sind [16]. Abbildung 1 zeigt schematisch einen Baum, in dem es an oberster Stelle einen Wurzelknoten gibt. Darunter befinden sich mehrere Kind-Knoten, die mit Ästen an den Wurzelknoten verknüpft sind. Jeder Kind-Knoten besitzt ein oder mehrere Blätter, wobei diese exakt einem Eltern-Knoten zugeordnet sind. Falls ein Kind-Knoten zu mehr als einem Eltern-Knoten zugeordnet werden kann, und die Pfade der Eltern-Knoten zu den Kind-Knoten disjunkt sind, spricht man von Polyhierarchie [18]. In einem kleinen Beispiel lässt sich das einfach veranschaulichen: Der ehemalige Präsident Ronald Reagan lässt sich als 'Blatt' zu den Knoten *Schauspieler* und *Präsident* zuordnen. Wenn diese beispielhafte Taxonomie eine Klassifikation hinsichtlich der Typus *Person* beschreibt, kann Ronald Reagan in zwei unterschiedliche Kind-Knoten eingeordnet werden. Er war Schauspieler als auch Präsident. Diese Polyhierarchien verursachen im Gegensatz zu strikten Hierarchien, in der jeder Kind-Knoten genau ein Eltern-Knoten besitzt, eine höhere Komplexität.

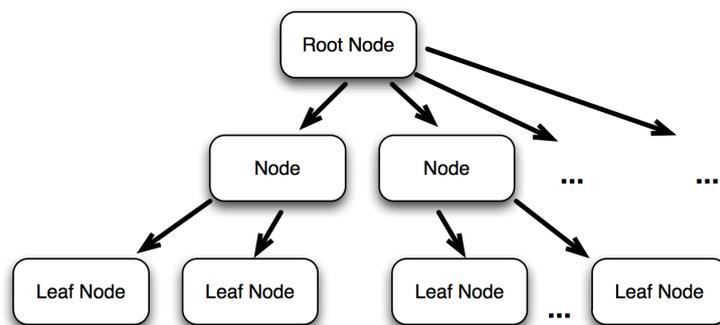


Abbildung 1: Hierarchische Taxonomie

Im Kontext dieser Arbeit werden Polyhierarchien betrachtet, da es durch-

aus sein kann, dass beispielsweise Point of Interests in mehrere Klassen zugeordnet werden können. Im Folgenden wird für die Begrifflichkeit "Klasse von Objekten", der Begriff der *Facette* weiterverwendet. Doch woher kommen diese "Wurzelknoten". In dieser Arbeit werden diese Wurzelknoten im weiteren Verlauf als High-Level Facetten, die darunter liegenden Kind-Knoten als Subfacetten und die Blätter als Instanzen definiert. Die Definition und Findung dieser High-Level Facetten stellt die Aufgabe dar und wird im Folgenden kurz an dem Begriff der Ontologie erläutert.

In den Informationswissenschaften beschreiben Ontologien eine formal geordnete Darstellung einer Menge von Konzepten, wobei diese Konzepte in bestimmten Beziehungen zueinanderstehen [19]. Eine Taxonomie einer Ontologie ist ein Spezialfall, wobei hier die 'is-a' Relationen aufgelistet sind. Weiterhin beinhaltet eine solche Taxonomie, die Beschreibung von Facetten zu ihren Subfacetten. Das folgende Beispiel soll eine Relation bei semantischen Daten verdeutlichen.

```
<owl:Class rdf:about="http://dbpedia.org/ontology/␣
    BasketballPlayer">
    <rdfs:label xml:lang="en">basketball player</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://dbpedia.org/␣
        ontology/Athlete"></rdfs:subClassOf>
</owl:Class>
```

Listing 1: Beispielontologie (OWL)

Das Beispiel zeigt einen Auszug aus der DBpedia-Ontologie<sup>3</sup>, in der die Beziehung zwischen der Facette *BasketballPlayer* und *Athlete* beschrieben ist. Die für diesen Kontext wichtigen Begriffe sind blau markiert. Die Aussagelogik in diesem Beispiel beschreibt eine hierarchische Anordnung der Klassen *BasketballPlayer* und *Athlete*, wobei ersteres eine *subClassOf* von *Athlete* ist.

Es stellt sich nun die Frage, welche Facetten als High-Level Facetten definiert werden, also Facetten, die in der Hierarchie-Ebene einer konzeptuellen Beschreibung von Informationen, an oberster Stelle stehen. In dieser Arbeit richtet sich die Definition der High-Level Facetten grundsätzlich an die Ontologie-Klassen der DBpedia. Das hat den Hintergrund, dass DBpedia einen Großteil der Informationen für das MobileFacets System bereitstellt. Die folgende Auflistung zeigt die für diese Arbeit relevanten High-Level Facetten:

- Place

---

<sup>3</sup>Die vollständige Ontologie lässt sich unter <http://wiki.dbpedia.org/Downloads351#dbpediaontology> einsehen.

- Person
- Organisation
- Work
- Species
- Event

Die Facetten *Place* und *Organisation* sind gerade für den mobilen Kontext interessant, da die darunter liegenden Ressourcen oft mit einer Geokoordinate versehen sind. Eine ortsbasierte Suche nach Informationen lässt sich somit einfach realisieren. Auch Personen können im mobilen Kontext von Bedeutung sein. Es können Fragestellungen wie "Welche Personen lebten in dieser Stadt" mithilfe dieser Facette und deren Instanzen beantwortet werden. Erwähnenswert ist zudem die Facette *Event*, da es für Anwender durchaus interessant sein kann, welche Events in der nächsten Zeit stattfinden.

### 3.3 Facettiertes Browsen

Zuerst muss unterschieden werden, inwiefern sich die normale Suche nach Informationen von einem facettierten Browsen unterscheidet. Man stellt sich ein Benutzer vor, der eine Bildersuche startet, um seine persönlichen Interessen zu befriedigen. Er möchte Street-Art Fotos in Koblenz und Umgebung finden. Standardisierte Suchmechanismen, wie das Eintippen eines Suchbegriffes mittels einer Suchmaschine wie der Google-Bilder Suche eignen sich nur bedingt für diese Aufgabe. Die Ergebnisansicht bietet keine Möglichkeit zur Filterung der Bilder hinsichtlich des Ortes oder des Aufnahmedatums. Nur durch geschickte Eingabe der Suchbegriffe lassen sich annähernd respektable Ergebnisse erzielen.

Ein Browsen und Suchen innerhalb von großen Informationskollektionen unterscheidet sich in den Ansätzen der normalen Schlüsselwort-basierten Suche. Eine Facetten-basierte Benutzeroberfläche erlaubt es Nutzern, ohne großes Vorwissen über das zugrunde liegende Datenschema, Informationen zu finden. Gerade wenn kein Vorwissen über die Struktur oder den Inhalt der Daten vorliegt, bedarf es einer anderen Suchtechnik. Die folgende Definition verdeutlicht dies:

*Basierend auf der Theorie von Facetten, ist das facettierte Browsen eine Technik zur Erkundung von strukturierten Datensätzen [20, 21].*

In Bezug auf das Beispiel der Bilder-Suche kann ein Nutzer im Falle des facettierten Browsens, durch den Datensatz navigieren. Jeder Navigationsschritt kann als Iteration zur Verfeinerung der Ergebnismenge angesehen werden. Dieser Mechanismus zur Verfeinerung der Ergebnismenge wird als Modifizierung der Original-Anfrage (Bilder-Suche in Koblenz) gesehen. Die selektierten Facetten agieren hier als Filter, indem nur die Ergebnisse zurückgeliefert werden, die eine Übereinstimmung mit diesen Facetten besitzen.

Typischerweise bestehen Facetten-basierte Benutzeroberflächen aus zwei Hauptbereichen: Ein Bereich für die Präsentation und die Interaktion mit Facetten, und eine Ergebnisansicht oder Liste von gefundenen Ergebnissen. Weitere Komponenten können Detailansichten zu selektierten Facetten und Instanzen sein oder eine simple Navigationshistorie [11]. Um eine hinreichend gute und qualitativ hochwertige Benutzeroberfläche für das facettrierte Browsen zu implementieren, müssen Fragen mit folgenden Aspekten betrachtet werden: Welche Datentypen werden für die Facetten sowohl als auch für die Instanzen genutzt? Wie werden die Facetten dem Benutzer präsentiert und können mehrere Facetten gleichzeitig selektiert werden? Das sind nur einige Fragen, die eine wichtige Rolle bei einem Design-Prozess spielen. Ab Kapitel 4 wird ausführlich auf das Design des MobileFacets-Prototypen eingegangen und es lassen sich auf Seite 21 weitere designkritische Punkte nachlesen.

### 3.4 Facetten-basierte Benutzeroberflächen

In diesem Abschnitt geht es um Facetten-basierte Benutzeroberflächen aus dem Desktopbereich.

**Flamenco** Das Flamenco Such-Framework [21] ermöglicht es Benutzern, flexibel durch große Informationssammlungen zu navigieren und einzelne Informationsaspekte zu erforschen. Das Ziel von Flamenco ist es, ein facettriertes Metadaten System zu generieren, das eine hierarchische Navigation innerhalb des System ermöglicht [3]. Dabei sollen Informationsräume flexibel, und ohne dass sich ein Nutzer in diesen Daten verloren fühlt, untersucht werden können. Auf der Webseite von Flamenco<sup>4</sup> werden einige Demos angeboten, in dem Benutzer beispielsweise die Nobelpreisträger der letzten 100 Jahre oder künstlerische Bilder des Kunstmuseums von San Francisco erkunden können.

Das Framework basiert auf den Arbeiten von M.A. Hearst [2, 3] und verwendet Algorithmen zur automatischen Generierung von hierarchischen

---

<sup>4</sup><http://flamenco.berkeley.edu/>

facettierten Metadaten, die für das Suchen und Browsen verwendet werden. Die Original-Metadaten werden aus den textuellen Beschreibungen (z.B. eine Kollektion von Nobelpreisträgern) von Artikeln extrahiert. Um eine gewisse Struktur aus den Informationskollektionen zu bekommen, die exakt den Inhalt dieser widergespiegelt, verwendet der Algorithmus eine Kategorisierung um Domänen zu erkennen. Dabei werden lexikalische Bibliotheken wie WordNet<sup>5</sup> verwendet. Dieser Vorgang setzt voraus, dass die eigentlichen Inhalte auch ausreichende Meta-Informationen bereitstellen. Im besten Falle sind diese noch kategorisiert, sodass eine einfache Facettenerkennung stattfinden kann.

Der Suchvorgang wird in diesen Informationskollektionen anhand von hierarchisch angeordneten Facetten realisiert. Beispielsweise sind die Bilder des Museums von San Francisco durch diese Facetten spezifiziert: Künstlername, Ort der Bildaufnahme, das Datum der Aufnahme und so weiter. Das User-Interface des Flamco-Frameworks ist in dem Beispiel Nobelpreisträger<sup>6</sup> zweigeteilt: Auf der linken Seite befindet sich die Möglichkeit, den Informationsraum durch gezieltes Navigieren innerhalb der angebotenen Facetten einzuschränken. Die rechte Seite zeigt die Ergebnismenge, in der zusätzlich die selektierten Facetten visuell dargestellt sind. Weiterhin kann eine Informationskollektion mittels einer Schlüsselwortsuche erkundet werden. Dass sich ein solches User-Interface, welches das facettierte Browsen und Suchen ermöglicht, auch intuitiv bedienen lässt, wurde in mehreren Usability-Studien geprüft [21].

**LENA** Hier handelt es sich um ein Fresnel Lens [22] basierter Linked-Data Browser, der eine facettierte Navigation innerhalb von verlinkten Daten ermöglicht und relevante Aspekte anhand des unterliegenden Wissens hervorhebt. LENA steht für **LE**ns based **NA**avigator und soll mithilfe von TripleRank ein qualitativ gutes und komfortables Browsen von semantischen Daten ermöglichen [4]. Im Gegensatz zu Flamenco bedient sich LENA von semantischen Datenquellen, wie beispielsweise DBpedia<sup>7</sup>. Im Semantic Web werden Informationen durch das Uniform Resource Description Framework (RDF) repräsentiert, dass es erlaubt Datenressourcen aus verschiedenen Quellen untereinander zu verlinken. Jede Ressource wird eindeutig über eine Uniform Resource Identifier (URI, engl. für "einheitlicher Bezeichner für Ressourcen") identifiziert, die auf eine einfache und erweiterbare Weise erzeugt wird. Sie spiegelt eine abstrakte oder auch physikalische Ressource

---

<sup>5</sup><http://wordnet.princeton.edu/>

<sup>6</sup><http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/nobel/Flamenco>

<sup>7</sup><http://dbpedia.org>

wieder [23]. Durch die eindeutig identifizierbare URI können sowohl semantische Relationen innerhalb von Datenquellen als auch Verlinkungen zwischen verschiedenen Datenquellen existieren.

Aufgrund der auftretenden Komplexität dieser Daten bedarf es gut definierten Suchstrategien, um nur die für den Nutzer relevanten Informationen anzeigen zu lassen. Die Relevanz einer semantischen Ressource wird von LENA durch TripleRank ermittelt. TripleRank ist eine Methode um RDF-Daten anhand eines RDF Graphen nach ihrer Relevanz anzuordnen. Dabei wird eine gewichtete Liste von Facetten errechnet, wobei jede Facette einen speziellen Informationsaspekt repräsentiert. Zu jeder Facette ist eine Liste von Ressourcen gelistet, die semantisch relevant mit dieser in Verbindung steht [24].

### 3.5 Facetten-basierte Benutzeroberflächen im mobilen Bereich

Ein einfaches und effizientes Verfahren um Informationen auf einem mobilen Endgerät zu erkunden, stellt einen gewissen Mehrwert für Anwender dar. Viele Anwendungen im mobilen Bereich stellen Funktionen bereit, wie das Suchen nach nachgelegenen Point of Interests oder bieten die Möglichkeit einer Navigationsführung an. Beispiele sind hier bekannte Systeme wie Navicon<sup>8</sup> und TomTom<sup>9</sup>. Doch wie sieht der Suchvorgang als solcher aus? Hier wird sich tendenziell an das Suchparadigma aus dem Desktopbereich orientiert, in der eine Suche über die Eingabe eines bestimmten Schlüsselworts definiert ist. Das setzt allerdings voraus, dass der Anwender ein hinreichendes Wissen mit sich bringt. Beispielsweise schwebt ihm ein konkreter Suchbegriff in Gedanken oder er weiß genau, welches Dokument er suchen möchte. Die Frage lautet nun, ob sich eine unpräzise Suchanfrage, bei der ein Anwender nicht genau weiß, was er suchen möchte, mithilfe eines einfachen Suchfeldes realisieren lässt. Gerade im mobilen Bereich ist es interessant zu wissen, "welche Informationen gibt es überhaupt in meiner unmittelbaren Umgebung"? Hier setzen mobile Lösungen an, die es Benutzern erlauben, komplexe Datenstrukturen auf einfache Art und Weise zu erforschen.

Charakterisierend für mobile Endgeräte sind die Eigenschaften, dass diese einen kleinen Bildschirm haben und dass es Restriktionen hinsichtlich der verfügbaren Bandbreiten gibt. Bei der Entwicklung einer mobilen Anwendung für das Suchen und Browsen von komplexen Datensätzen müssen demnach Prioritäten auf die Usability von solchen User-Interfaces gesetzt werden. Die

---

<sup>8</sup><http://www.navigon.com/portal/de/index.html>

<sup>9</sup><http://www.tomtom.com/>

folgenden Sektionen behandeln verschiedene Ansätze, wie man ein facetiertes Browsen auf einem mobilen Endgerät umsetzen kann.

**FaThumb** Ist ein Facetten-basiertes User-Interface für mobile Endgeräte [15], um eine Suche innerhalb von großen Informationskollektionen zu ermöglichen. Der Lösungsansatz liegt in der Umsetzung einer tastaturgesteuerten Navigation für hierarchisch angeordnete Facetten. Ihr Verständnis von Facetten basiert hauptsächlich auf den Arbeiten von Hearst et al. [25]. Sie verwenden die von den Inhalten gelieferten Metadaten als Attribute, um sie anschließend in orthogonal angeordnete Dimensionen (Facetten) zu strukturieren.

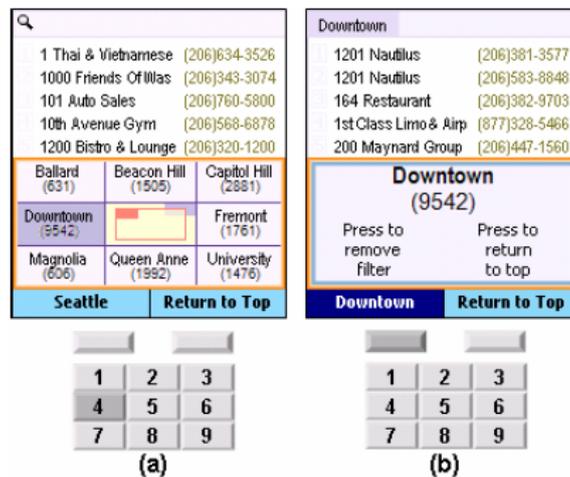


Abbildung 2: FaThumb (Screenshot)

Als Datenbasis der Anwendung wird die Datenbank von Yellow Pages<sup>10</sup> des Staates Washington genommen. Die verwendeten Attribute aus dieser Datenbank sind unter anderem Unternehmensnamen, Adressen, Telefonnummern und andere Kontaktdaten. Sie entwickelten einen Algorithmus, bei dem Attribute anhand ihrer Eigenschaft in verschiedene Facetten klassifiziert werden. Dabei spielen folgende Eigenschaften eine Rolle: Acht Top-Level Facetten mit einem Maximum von je acht Attributen auf jedem darunter liegenden Level.

Abbildung 2 zeigt einen Ausschnitt der FaThumb-Anwendung. Das User-Interface ist in vier Regionen strukturiert: Filter, Ergebnisse, Navigation über Facetten und eine aktive Region, welche farbig umrandet ist. Die Navigationsregion zeigt einen hierarchisch angeordneten Baum von Facetten,

<sup>10</sup><http://yellowpages.lycos.com/>

die aus dem Yellow Pages Datensatz kommen. Durch Anklicken einer Facette wird die Ergebnisliste oberhalb der Navigationsregion in Abhängigkeit dieser Facette gefüllt. Die Navigation durch den Benutzer erfolgt bei diesem System anhand der numerischen Tastatur des mobilen Endgerätes.

Anhand von einer Usability-Studie wurde herausgefunden, dass die Teilnehmer diese Art der Facettenavigation schnell angenommen haben.

**Dbpedia Mobile** Dieser Dienst ist eine ortsbasierte DBpedia-Client Anwendung für mobile Endgeräte, bestehend aus einer Kartenansicht und einem Fresnel Linked Data Browser. Dbpedia-Ressourcen werden anhand der Geokoordinate des Benutzers gefunden [26]. Somit ist es eine Kartenbasierte Sicht auf umliegende DBpedia-Ressourcen. Die Kartenansicht wird über den mobilen Browser des jeweiligen Endgerätes aufgerufen. Die Client-Anwendung an sich beschränkt sich nur auf die Ermittlung der Position via GPS. Derzeit wird nur der mobile Opera Browser in der Version 8 unterstützt. Das liegt daran, dass Opera eine ausreichende Unterstützung des Document Object Models 1 und 2 liefert, welche für die unterliegende Google-Maps API benötigt wird. Dbpedia Mobile ist mehr ein standortbezogener Dienst um Textinformationen sowie Fotos zu konsumieren, statt ein Facetten-basierter Browser um eine Navigation innerhalb von Informationskollektionen zu ermöglichen. Dennoch kann man sagen, dass es sich um bei diesen DBpedia-Ressourcen um Instanzen aus der Facette *Place* handelt.

**Mobile-cultural-heritage-guide** Dieses System fungiert als mobiler Reiseführer um Informationen rund um Amsterdam zu konsumieren. Das Suchen und Browsen von Informationen soll dabei mit minimalen User-Interaktionen möglich sein [27]. Deren Ansatz liegt in der Kombination zweier Wissensbasen: das Allgemeinwissen von Point Of Interests in der unmittelbaren Umgebung und das spezialisierte Wissen im Kontext der kulturellen Informationen. Erstere Informationen werden mittels Anbindung an die Linked Open Data Cloud realisiert, wo unter anderem Ressourcen von DBpedia abgefragt werden. Diese gefundenen Point of Interests werden durch weiteres Wissen in Form von Öl-Gemälden, Fotografien und Künstlerinformationen ausführlich erweitert. Dies wird realisiert durch einen Annotationsvorgang, bei dem die Objekte des MultimediaN E-Culture Projektes<sup>11</sup> semantifiziert und diese entstandenen RDF-Triple mit Hintergrundwissen von Thesauri und Schlüsselwortlisten annotiert werden. Die so entstandene RDF Repräsentation kann genutzt werden, eine Relation zwi-

---

<sup>11</sup>Unter anderem eine Datenkollektion des Instituts für Kunstgeschichte der Niederlande; siehe <http://e-culture.multimedien.nl/>

schen den Daten des E-Culture Projekts und den Ressourcen der DBpedia aufzubauen.

Die Interaktion mit Facetten wird realisiert, in dem die aktuelle Position des Nutzers ermittelt wird und nur die "Kultur-relevanten" Facetten dargeboten werden. Kultur-relevante Facetten sind Orte, Events, Kunstwerke und Personen. Basierend auf einer Auswahl einer gewissermaßen High-Level Facette, erscheinen Subfacetten wie Malereien, Bücher, Aktivisten und Künstler [2]. Der linke Screenshot der Abbildung 3 zeigt die Anwendung. Der Bildschirm ist in drei relevante Bereiche geteilt: Im oberen Segment wird eine Augmented Reality (deutsch: Erweiterte Realität) Ansicht verwendet, die das aktuelle Kamerabild des mobilen Endgerätes nutzt, um Informationen an das reale Bild visuell zu annotieren [28]. Darunter werden Metadaten zu dem angezeigten Objekt dargestellt. Facetten werden anhand der darunter liegenden zwei Leisten ausgewählt. Die bläulich markierte Leiste zeigt die verfügbaren High-Level Facetten an. Subfacetten lassen sich in der roten Leiste selektieren. Es fällt auf, dass die Möglichkeiten in diesem Design stark hinsichtlich der Anzahl der Facetten begrenzt sind. Lediglich vier High-Level und je vier Subfacetten lassen sich mit diesem Design selektieren.



Abbildung 3: Links: iPhone Applikation *Mobile cultural heritage guide: location aware semantic search* [29], Rechts: *mSpace Mobile: A Mobile Application for the Semantic Web* [27]

**mSpace Mobile** Diese Anwendung stellt ortsgebundene Informationen bereit, während man mit einem mobilen Endgerät unterwegs ist [27]. Der Suchkontext wird durch den physischen Standort des Anwenders bestimmt, welche durch die Geokoordinaten definiert sind. In Abhängigkeit der verfügbaren Daten aus der unmittelbaren Umgebung stellt das User-Interface fünf Kategorien bereit: z.B. Point of Interests, eine Kartenansicht oder Metainformationen zu Objekten. Im rechten Screenshot der Abbildung 3 sieht man, dass das User-Interface in sechs kleine Kacheln unterteilt ist, wobei jede Kachel einen Informationsaspekt repräsentiert. Diese Art der Bildschirmaufteilung soll das Beantworten von komplexen Fragen erleichtern, ohne dass Nutzer manuelle Eingaben tätigen müssen [30] [31].

Die oberen beiden Kacheln fungieren hier gewissermaßen als Navigation innerhalb der verfügbaren Facetten. Durch Auswählen einer Facette ändert sich der Inhalt der unteren Kacheln, wie beispielsweise die Informationen zur ausgewählten Facette oder den POIs der Karte. MSpace mobile lässt sich nur mit einem berührungsempfindlichen Stift bedienen. Hier muss man sich die Frage stellen, inwiefern eine solche Lösung mit einem Touchscreen-basierenden Endgerät zu realisieren ist.

## 4 Oberflächen-Entwurf

Bei dem Oberflächen-Entwurf stellt sich die Frage, wie der grundsätzliche Aufbau der Applikation aussehen soll, so dass das facetiierte Suchen und Browsen bequem und sinnvoll abgebildet wird. Gerade im mobilen Bereich ergeben sich wegen einer beschränkten Bildschirmgröße komplexe Anforderungen an die Gestaltung der Oberfläche. Zum einen müssen die Interaktionen des Benutzers mit der Applikation in einem typischen mobilen Umfeld (z.B. beim Gehen mit einer freien Hand) einfach zu bewerkstelligen sein, zum anderen ergeben sich komplexe Anforderungen in Bezug auf das facetiierte Suchen und Browsen. Wegen des kleineren Displays müssen neue Ansätze gefunden werden um über Facetten zu Browsen.

In der Literatur [1, 32] wurden folgenden Anforderungen definiert, die dazu da sind, um eine hinreichend gute Benutzeroberfläche für die facetiierte Navigation zu realisieren. Sie spielen bei dem Designentwurf eine tragende Rolle:

- Flexible Navigation innerhalb von Informationsbereichen
- Nahtlose Integration einer Schlüsselwort-Suche
- Fließender Übergang zwischen Verfeinerung und Erweiterung von Informationskollektionen
- Vermeidung von leeren Ergebnislisten; Keine Suche führt ins Leere [32]
- Nachvollziehbare Navigationsstrukturen
- Der Nutzer definiert seinen Suchweg frei [32]

Anhand dieser Anforderungen und den aus Kapitel 2 definierten Szenarien müssen die Entwürfe der Benutzeroberfläche ausgerichtet werden. Gerade der Zusammenschluss all dieser Anforderungen ist einer der Hauptaspekte dieses Designentwurfs, um den Benutzer ein ausreichendes Maß an Usability und die Vorteile des facetiierten Suchens und Browsen zu gewährleisten. Für eine ausreichend gute Qualität des Prototypen und der anschließenden Designstudie bedarf es eines gut durchdachten Designs. Im Folgenden wird auf die Entwicklung einer prototypischen Oberfläche eingegangen und Beispiele anhand von Mockups gezeigt. Die Entwicklung gliedert sich dabei in zwei Phasen.

## 4.1 Erste Entwurfsphase

Für das erste Design stellt sich die Frage, welche funktionalen Anforderungen sich in der Oberfläche widerspiegeln müssen. Da zu diesem Zeitpunkt des Designentwurfs noch keine konkreten funktionalen Anforderungen erhoben sind, wird versucht das typische facetthierarchische Browsen, wie beispielsweise aus wissenschaftlichen Arbeiten [1, 6, 10] oder von kommerziellen Webseiten wie Ebay<sup>12</sup> oder Amazon<sup>13</sup>, auf einem mobilen Endgerät umzusetzen. Dabei spielen die zwei Szenarien POIs-Suche und Trivia eine maßgebliche Rolle bei der Gestaltung der Oberfläche.

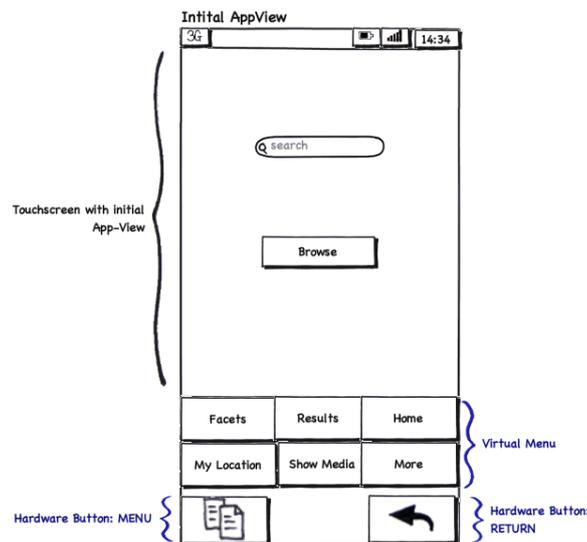


Abbildung 4: Erste Oberflächengestaltung beim Start der Applikation (Mockup)

Design-Entwürfe lassen sich entweder Papier-basiert oder computergestützt realisieren. Das erste Design für das Mockup-Prototyping wird mit Hilfe von der Software Balsamiq Mockups<sup>14</sup> erstellt. Als Einstiegspunkt der Applikation kann die Unterteilung in Suchen und Browsen sinnvoll sein. Abbildung 4 zeigt einen ersten Entwurf, in dem der Benutzer bei Programmstart die Möglichkeit hat, zwischen einer Suche und dem freien Browsen zu wählen. Der Entwurf orientiert sich an die prinzipielle Aufteilung

<sup>12</sup>[www.ebay.de](http://www.ebay.de)

<sup>13</sup>[www.amazon.de](http://www.amazon.de)

<sup>14</sup><http://www.balsamiq.com/products/mockups>

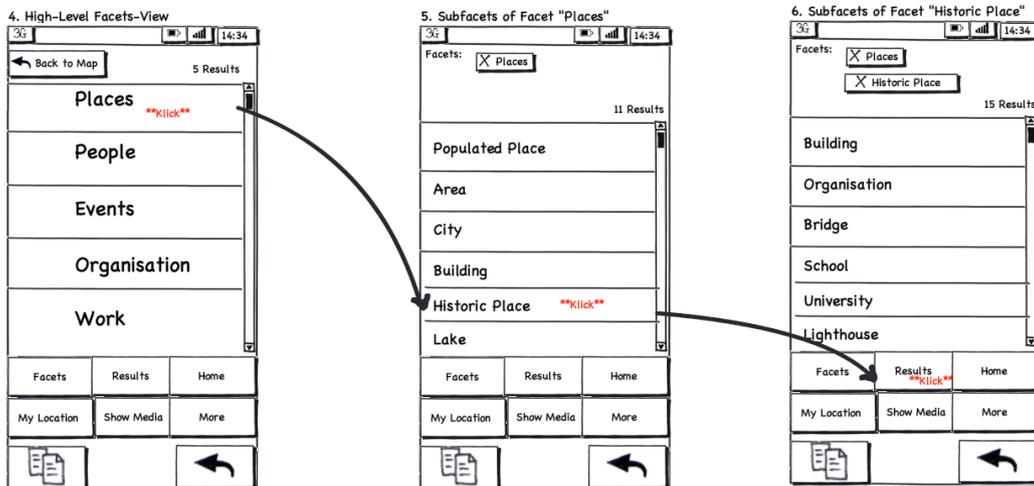


Abbildung 5: Browsen über Facetten (Mockup)

des Bildschirms von Google-Android basierenden Endgeräten. Der obere Bildschirmteil repräsentiert den Touchscreen für die Interaktion mit der Applikation. Die oberste Leiste zeigt grundlegende Betriebssystem-Funktionen wie Uhrzeit, Akku-Stand und Empfangsqualität. Unter dem Touchscreen befinden sich Hardware-Knöpfe, die unabhängig von der jeweiligen Applikation existieren. Der Mockup zeigt zudem ein virtuelles Menü, welches sich durch einen Hardwareknopf aktivieren lässt.

Zu beachten ist, dass das virtuelle Menü in diesem Beispiel scheinbar unter dem Touchscreen liegt. Jedoch ist es in der Praxis so, dass das Menü ein Teil des unteren Bildschirmbereiches einer Applikation überdeckt. Die Darstellung des Menüs unter dem Touchscreen wird der Übersicht halber dauerhaft "eingebledet".

Abbildung 5 zeigt den ersten Entwurf, wie man das Browsen über Facetten umsetzen kann. Die Aufteilung des Bildschirms ist in zwei Hälften gegliedert. Im oberen Teil befindet sich die sogenannte "Facettenansicht". Der untere Teil nimmt die Funktion des Browsens ein. Im linken äußeren Bild sieht man den Einstieg in das facettierte Browsen. Zu sehen sind die High-Level Facetten, die hier als Ausgangspunkt dienen. Durch einen Klick auf ein Listenelement öffnet sich eine neue Liste (zu sehen in der Mitte), in der alle Subfacetten enthalten sind. Durch einen weiteren Klick auf ein Listenelement öffnet sich wiederum eine neue Liste, in der die zugehörigen Subfacetten enthalten sind. Wie zu sehen ist, wird das Browsen mittels einfachen, scrollbaren Listen realisiert.

Der obere Bildschirmteil zeigt die zuvor ausgewählten Facetten als einfache

che Textelemente. Je nachdem welche Facette man auswählt, wird diese dynamisch zur oberen Liste hinzugefügt. Falls der Benutzer wünscht Facetten zu löschen, dienen dazu kleine Kreuze unmittelbar neben den Facetten.

## 4.2 Zweite Entwurfsphase

Während den ersten Mockups stellte sich heraus, dass für die Evaluation eine andere Darstellungsform sinnvoller wäre. Das Design wurde an die Google-Android Benutzeroberfläche angepasst und so wurde erreicht, dass die Oberflächenelemente wie beispielsweise Knöpfe, Menüs und Listen originalgetreu abgebildet sind. Das hat für die Evaluation zum einen den Vorteil, dass die Teilnehmer sich im Vorfeld ein konkretes Bild des lauffähigen Prototypen machen können. Zum anderen ist ein homogene Darstellungsform des Prototypen auf dem Papier, sowie letztendlich eine lauffähige Version auf einem Endgerät für den Verlauf der Evaluation von Vorteil.

Für diese zweite Stufe des Designprozesses kam Microsoft Visio<sup>15</sup> zum Einsatz. Für ein prototypisches Mockups-Design gibt es für das Android-Betriebssystem eine umfangreiche Sammlung von Designelementen<sup>16</sup>.

Abbildung 6 auf Seite 25 zeigt die grundlegende Designänderung. Nicht nur das Design der Mockups wurde in dieser Phase geändert, zudem handeln die nächsten Schritte des Entwurfs davon, die Szenarien aus Kapitel 2 abzubilden, sowie designkritische Elemente herauszukristallisieren.

### 4.2.1 Szenario: POI-Suche

Das Szenario POI-Suche handelt davon, ortsabhängige Informationen zu finden. Bei diesem Location-based Kontext muss man sich die Frage stellen, inwiefern man gefundene Point of Interests visuell darstellen möchte. Im mobilen Bereich wird häufig von einer virtuellen Kartenansicht Gebrauch gemacht, wie beispielsweise Google-Maps<sup>17</sup>. Als Einstiegspunkt für die Applikation bietet sich eine solche Kartenansicht an, siehe Abbildung 6. Im linken Bild sieht man auf der Kartenansicht ein kleines Dialogfenster, welches signalisiert, dass nach der aktuellen Position gesucht wird. Mittels eingebauten GPS-Empfänger lässt sich die genaue Position des Benutzers ermitteln, so dass die Kartenansicht auf den Standort zentriert werden kann. Im Hintergrund kann die Applikation nach umliegenden und verfügbaren Facetten suchen. Dieser Vorgang wird in dem rechten Bild als erfolgreich gemeldet. Hier hat

---

<sup>15</sup><http://office.microsoft.com/en-gb/visio/>

<sup>16</sup><http://www.artfulbits.com/Android/Stencil.aspx>

<sup>17</sup><http://maps.google.de/>

der Benutzer die Möglichkeit nach den gefundenen Facetten zu Browsen oder dies nicht zu tun. Hier repräsentiert die rote Hand einen Touchscreen Klick.

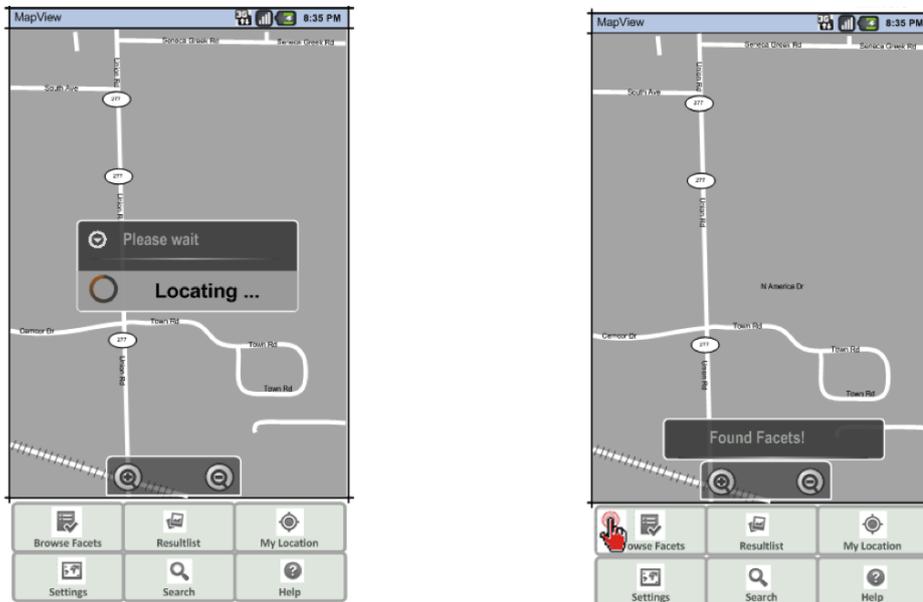


Abbildung 6: Szenario: POI-Suche, Anwendungsstart (Mockup)

Durch den Klick auf "Browse Facets" kann der Benutzer, wie in Abbildung 7 auf Seite 26 gezeigt, mit Facetten interagieren. Für den Entwurf stellte sich die Frage, wie man eine facetten Navigation auf einem mobilen Endgerät realisieren kann. Die Navigation erfolgt, wie schon in Beispiel 5 gezeigt, über dynamisch erzeugte Listen. Als Einstiegspunkt dienen auch hier die High-Level Facetten. Als nützliche Zusatzinformation wird jeweils die Anzahl der Subfacetten angezeigt. Der obere Bildschirm beinhaltet zwei Funktionen. Zum einen kann der Benutzer zurück zu Kartenansicht, zum anderen lassen sich mittels des Knopfes "Filters" zuvor ausgewählte Facetten einsehen. "Filters" wird hier synonym für den Begriff der Facette verwendet<sup>18</sup>. Die Verfeinerung der Ergebnismenge erfolgt durch Anklicken eines gewünschten Listenelementes. Je nachdem welche Facette der Benutzer auswählt, ändert sich die Liste dementsprechend<sup>19</sup>.

Durch Anklicken des Knopfes "Filter" öffnet sich ein Dialogfenster (Abbildung 8), in der die zuvor ausgewählten Facetten angezeigt werden. Die Checkboxen auf der rechten Seite signalisieren, dass in dem Beispiel

<sup>18</sup>Der Begriff "Filters" für die Auswahl an Facetten orientiert an sich die *Faceted Wikipedia Search*, <http://dbpedia.neofonie.de>

<sup>19</sup>siehe rechte Abbildung: abhängige Subfacetten zur Facette Places

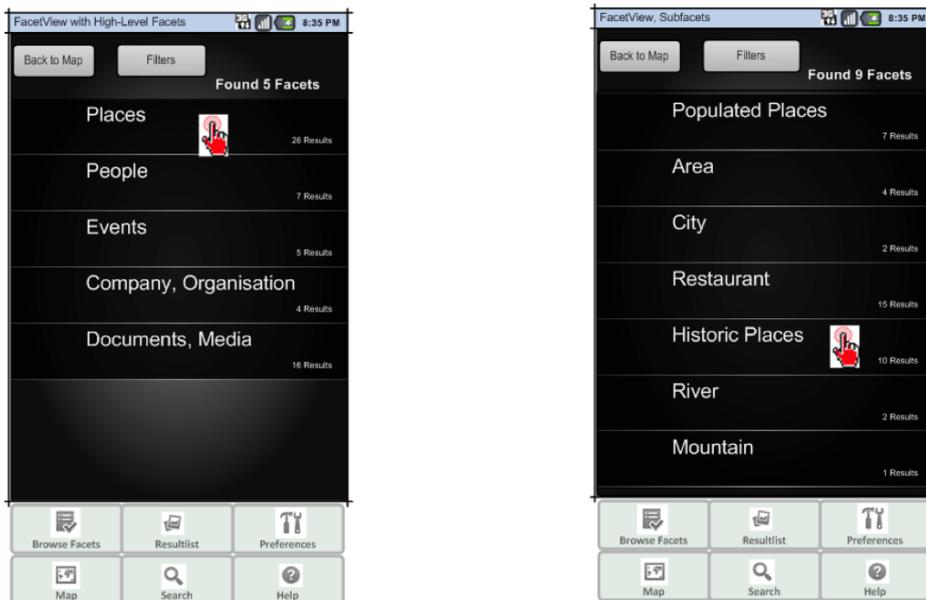


Abbildung 7: Variante 1: Browsen über Facetten (Mockup)

*Places* und *Populated Places* ausgewählt sind. Das ist nur eine Möglichkeit der Darstellung von ausgewählten Facetten. In Abschnitt 4.2.5 Darstellung der Facettenauswahl finden sich noch andere Varianten. Für die Evaluation des Szenarios POI-Suche wird diese Darstellungsweise aus Abbildung 8 den Teilnehmern gezeigt.

In dem Szenario POI-Suche Szenario liegt der Fokus auf der Suche nach ortsabhängigen Informationen. Für die Facette *Places* bietet sich eine Karte an, um Ergebnisse visuell darzustellen. Facetten *People*, *Organisation* und *Documents* besitzen in der Regel keinen konkreten Ortsbezug, somit ließe sich hier eine Listendarstellung der gefundenen Ergebnisse realisieren. Abbildung 9 zeigt einen Kartenausschnitt mit gefundenen Ergebnissen. Ergebnisse, in diesem Kontext als Point of Interests definiert, werden durch kleine Hinweissymbole an den entsprechenden Positionen angezeigt. Damit der Benutzer die Übersicht nicht verliert und auf den ersten Blick sieht, um welche Information es sich handelt, wird neben dem Symbol der Name oder Titel der Information eingeblendet.

Die Navigationsführung der Applikation ist in diesem Szenario mit Hilfe von Menüs realisiert. Im unteren Bildschirmteil der Mockups befindet sich ein virtuelles Menü, anhand dessen der Benutzer die einzelnen Funktionen der Applikation abrufen kann. In Abbildung 9 sieht man insgesamt sechs Menüeinträge. *Browse facets* dient dazu, um den Bildschirm aus Abbildung 6 aufzurufen. *Resultlist* ist eine andere Darstellungsart von gefundenen



Abbildung 8: Darstellung der ausgewählte Facetten (Mockup)



Abbildung 9: Kartensicht mit POIs (Mockup)

Ergebnissen. Hier lässt sich im Gegensatz zur Kartenansicht eine einfache Liste vorstellen. Ergebnisse werden nach absteigender Häufigkeit oder Priorität in einer Liste dargestellt. *My Location* dient dazu, die Kartenansicht auf diese Position des Benutzers zu zentrieren. Im Menüpunkt *Settings* lassen sich einfache Programmeinstellungen vornehmen. *Search* ruft die Funktion der Suche nach Facetten oder Inhalten auf. In Szenario "Trivia" wird diese Funktion gezeigt und näher diskutiert. Über den Menüeintrag *Help* kann sich der Benutzer über die angebotenen Funktionen der Applikation informieren.

#### 4.2.2 Szenario: Trivia

Im Gegensatz zum ersten Szenario löst sich Paul von der reinen ortsabhängigen Informationsgewinnung und ändert den Suchkontext Berlin nach Koblenz. Signifikant für dieses Szenario ist die Verwendung einer Suchfunktion und eine andere Philosophie der Navigation. Abbildung 10 illustriert die Suchfunktion der Applikation. Zu sehen ist ein typisches Eingabefeld, in dem der Benutzer mittels der virtuellen Tastatur des Gerätes Eingaben tätigen kann. Die Suchfunktion bietet in diesem Beispiel eine Live-Vorschau. Nach der Eingabe der ersten beiden Buchstaben öffnet sich eine Liste mit vorgeschlagenen Resultaten. Rückblickend auf das erste Szenario verwendet dieser Mockup kein virtuelles Menü. Lediglich die obere Tableiste dient zur Navigation. Zu sehen sind die Reiter Suchfunktion, die Kartenansicht, die Facettenansicht und eine Ergebnisliste. Die Kartenansicht ist analog zum ersten Szenario gestaltet (siehe Abbildung 6).

Abbildung 11 illustriert einen anderen Ansatz der Darstellung von ausgewählten Facetten. Die Dialogform aus Szenario POI-Suche wird hier mit einer Liste im oberen Bildschirmbereich ersetzt. Dabei stehen zusammengehörige, also in diesem Beispiel hierarchisch strukturierte Facetten jeweils in einer Zeile. Zu sehen ist die High-Level Facette *Places* mit der zugehörigen Auswahl Berlin. Zu beachten ist, dass die Applikation bei Programmstart die Ortsfacette als Ausgangspunkt für weiteres Browsen annimmt. Das heißt alle andere Facetten, wie beispielsweise Personen und Events, sind unmittelbar abhängig von der Ortsfacette. Weiteres zum Verständnis von Facetten befinden sich in Kapitel 3, Verwandte Arbeiten.

Der Knopf unmittelbar neben den Facetten kann verwendet werden, um zuvor ausgewählte Facetten wieder zu löschen. Somit wird die Ergebnismenge wieder vergrößert. Der untere Bildschirmbereich dient dazu, um über Facetten zu Browsen. Analog zum Szenario POI-Suche wird hier eine einfache Listenansicht verwendet, die dynamisch generiert wird.

Falls der Benutzer die Ergebnismenge einsehen möchte, bietet sich der rechte Tabreiter "Resultlist" an. Im Gegensatz zur Kartenansicht werden die

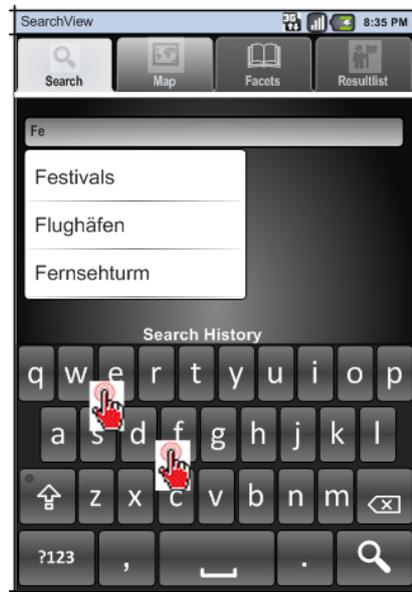


Abbildung 10: Darstellung der Suchfunktion (Mockup)

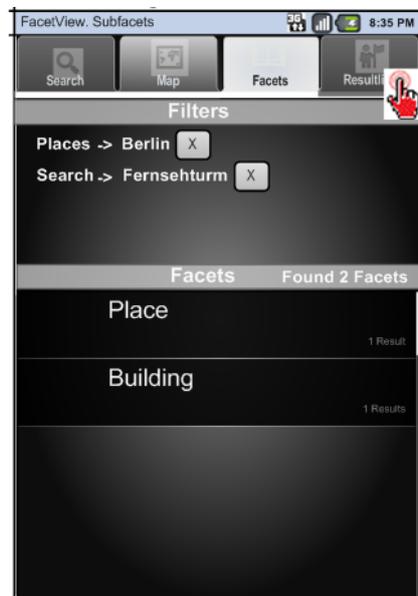


Abbildung 11: Szenario Trivia: Darstellung der ausgewählten Facetten und das Browsen über Facetten (Mockup)

Ergebnisse in einer Liste dargestellt. In Abbildung 12, linkes Bild, sieht man die Ergebnisliste. Die Ergebnisliste enthält hier lediglich ein Element, welches durch die Suchbegriffe Fernsehturm und Berlin errechnet wurde. Das heißt, es wird nach Instanzen gesucht, die zu beiden Suchbegriffen (Fernsehturm und Berlin) logisch verknüpft passen. Das Listenelement zeigt den Titel und beispielsweise die erste Zeile aus einem Wikipedia-Artikel. Durch einen Klick auf den Fernsehturm gelangt der Benutzer auf eine Detailansicht in der weitere Informationen angezeigt werden. Mithilfe des Knopfes "Options" können darüber hinaus, noch weiterführende Ressource-Informationen aufgerufen werden sind. Hier lassen sich Webseiten, weitere Bilder und andere Medien anzeigen.

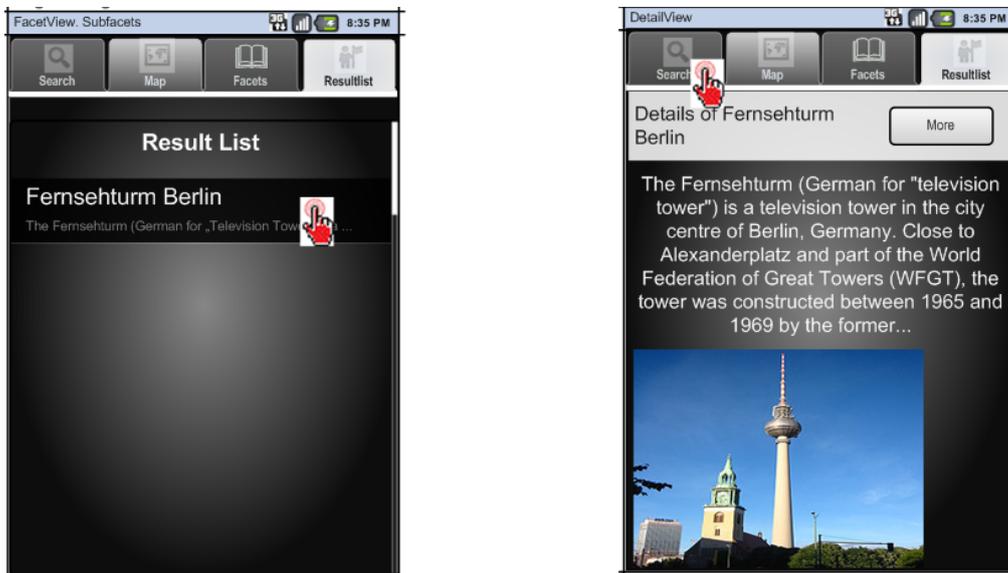


Abbildung 12: Szenario Trivia: Ergebnisliste und Detailansicht (Mockup)

Die vollständigen Mockups der Szenarien POI-Suche und Trivia befinden sich auf der beigelegten CD.

#### 4.2.3 Verwendung von Tabs und Menüs

Nach dem Design der Szenarien kam die Frage auf, welche designkritischen Elemente für die Evaluation von Bedeutung sind. Bei der Verwendung von Tableisten und Menüs ist es interessant zu wissen, inwiefern die Teilnehmer eine Präferenz bei Applikationen mit oder ohne Tableisten haben. Dazu wurde ein direkter Vergleich bei den Szenarien durchgeführt. Beim Szenario POI-Suche wurde auf eine Tableiste völlig verzichtet, während beim Szenario

Trivia Tableisten eine wichtige Rolle für die Interaktion und Navigation einnahmen. Menüs haben die Eigenschaft, dass der Bildschirmbereich nicht eingeschränkt wird. Es bleibt mehr Platz für die Kartenansicht oder es können mehr Elemente in der Listenansicht angezeigt werden. Tableisten nehmen hier einen festen Teil des Bildschirms ein. Jedoch muss beachtet werden, dass die nötige Anzahl der Schritte zur Ausführung einer Interaktion sich zwischen Menüs und Tableisten unterscheidet. Als Beispiel möchte der Benutzer nach Programmstart über die verfügbaren Facetten browsen. Eine Navigation per Menü aus Abbildung 6 auf Seite 25 bedarf zwei Schritte um zum gewünschten Ziel zu kommen. Die Aktivierung des Menüs per Hardwareknopf und das Anklicken des entsprechenden Menüpunktes. Bei der Tableisten-Variante lässt sich die Funktion einfach über die Tableiste mit nur lediglich einem Klick anwählen.

#### 4.2.4 Browsen über Facetten

Die Interaktion mit Facetten nimmt in der Applikation einen sehr hohen Stellenwert ein. Die Beantwortung von komplexen Fragen wie " Welches Hochhaus in New York hat mehr als 50 Stockwerke und wurde vor 1980 erbaut?" gegenüber Wikipedia Einträgen stellen höhere Anforderungen an das User-Interface als beispielsweise Schlüsselwort-basierte Suchmechanismen. Das System sollte die Navigation auf heterogenen Informationsräumen bereitstellen, in dem der Benutzer diesen Informationsbereich durch seine Interaktionen in Teilmengen zerlegen kann [33]. Durch hierarchisch angeordnete Facetten lassen sich Teilmengen durch gezielte Auswahl von Schlüsselwörtern bilden. Die erste Variante lässt sich in Abbildung 7 auf Seite 26 einsehen. Der Benutzer verfeinert die Grundgesamtheit des Informationsraumes, in dem er eine Hierarchie-Ebene absteigt. Die Hierarchie wird hier definiert durch die Facetten mit ihren zugehörigen Subfacetten. In diesem Beispiel wählt er die Facette *Places* aus und gelangt somit zu den Subfacetten *Populated Places*, *Area*, *City* und so weiter. Diese Variante hat den Vorteil, dass der Benutzer durch eine einfache Liste ohne großen Informationsoverhead nicht überfordert wird. Nachteilig bei dieser Variante ist, dass der Benutzer leicht die Übersicht verlieren kann, falls die Anzahl der Subfacetten sehr groß ist. Beispielsweise eine Liste mit über 50 Einträgen hat zur Folge, dass der Benutzer sehr lange nach unten scrollen müsste, um alle Listeneinträge einsehen zu können. Eine Sortierung der Listeneinträge nach einem bestimmten Kriterium, wie etwa die Entfernung zu den Ressourcen, kann hier die Usability steigern. Weiterhin spielt die Anzahl der Hierarchie-Ebenen eine wichtige Rolle. Angenommen die High-Level Facette *Places* enthält mehr als fünf Hierarchie-Ebenen, so kann der Benutzer schnell den

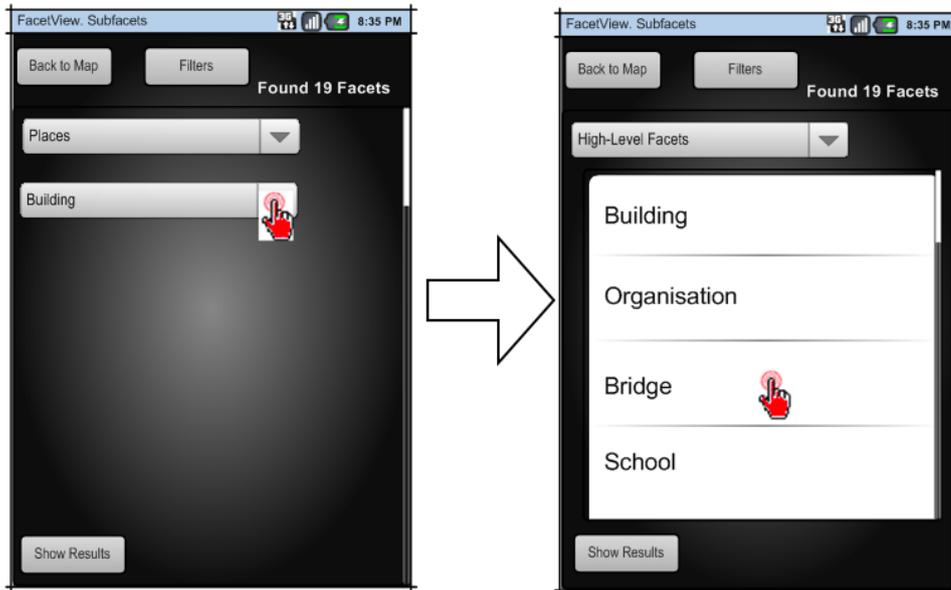


Abbildung 13: Variante 2: Browsen über Facetten (Mockup)

Überblick verlieren, in welcher Ebene er sich gerade befindet.

Der nächste Abschnitt *Darstellung der Facettenauswahl* zeigt einige Beispiele, um zuvor ausgewählte Facetten sinnvoll dem Benutzer zu präsentieren. Abbildung 13 zeigt die Facettenauswahl anhand eines Drop-Down Menüs. Durch Auswahl der High-Level Facette öffnet sich ein zusätzliches Drop-Down Menü, in dem zugehörige Subfacetten ausgewählt werden können. Vorteil ist hier, dass die Übersicht über die aktuelle Hierarchie-Ebene bewahrt werden kann.

Variante 3 hingegen (Abbildung 14) orientiert sich an mSpace Mobile [27]. Der Bildschirmbereich wird in kleine Teilbereiche zugeschnitten in dem das Browsen über Facetten realisiert wird. Links oben können die High-Level Facetten ausgewählt werden. Die Listen rechts und unten werden in Abhängigkeit der zuvor ausgewählten Facetten dynamisch gefüllt.

#### 4.2.5 Darstellung der Facettenauswahl

Bei der Darstellung der Facettenauswahl geht es um das Design der ausgewählten Facetten. Der Aspekt ist insofern interessant, da bei komplexen Anfragen die Übersicht beim Browsen verloren gehen kann. Dem Benutzer muss es jederzeit ersichtlich sein, in welchem Informationskontext er sich gerade befindet. Gerade bei der Navigation in multiplen Hierarchien muss der Benutzer den Überblick behalten können und sich nicht verloren fühlen [34].



Abbildung 14: Variante 3: Browsen über Facetten (Mockup)

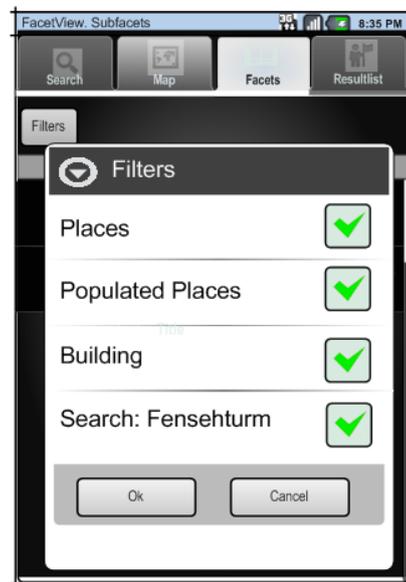


Abbildung 15: Darstellung der ausgewählten Facetten (Mockup)

Abbildung 15 illustriert zwei Entwürfe. Beim linken Entwurf wird die vorhandene Hierarchie visuell dargestellt. *Places* ist hier die High-Level Facette. Der Pfeil ausgehend von *Places* zeigt auf *Populated Places* und soll darauf hindeuten, dass *Populated Places* eine Subfacette von *Places* ist. Der Knopf mit einem Kreuz an jeder Zeile ist dafür da, um die ausgewählten Facetten wieder zu entfernen. Die rechte Abbildung verzichtet auf die Darstellung der Facetten-Hierarchie und zeigt alle ausgewählten Facetten in einer Liste an. Neben jedem Listenelement befindet sich eine Checkbox, um die jeweiligen Facetten wieder abzuwählen. In diesem Beispiel muss der Benutzer einen zusätzlichen Dialog aufrufen, welches bei der linken Variante entfällt, da die Darstellung der ausgewählten Facetten ein fester Bestandteil des Bildschirms einnimmt.

## 5 Evaluation der Designstudie

Das folgende Kapitel hat zum Ziel eine Designstudie des Prototypen durchzuführen. Im Vordergrund stehen keine technischen Aspekte, sondern es geht vielmehr um designkritische Entscheidungen von User-Interface Elementen, die letztendlich die Usability des Prototypen maßgeblich beeinflusst. Um am Ende dieser Designstudie hilfreiche und konkrete Ergebnisse zu gewinnen, müssen die Ziele hinsichtlich der zu evaluierenden Aspekte genau definiert werden.

In Bezug auf eine Software-Evaluation können die Ziele auf drei einfache Fragen reduziert werden [35]: Welches System ist besser? Wie gut ist das System? Warum ist es schlecht? Die erste Frage zielt auf den direkten Vergleich von eigenen, sowie Ansätzen von alternativen Systemen. Die Entscheidungsfindung basiert einerseits auf technische Aspekte (Welche Funktionen bietet das System X, welche das System Y), andererseits auf die individuelle Einschätzung und Beurteilung von Testpersonen. Bei der zweiten Frage geht es um die Klassifikation des Systems anhand einer Qualitätsskala. Jedoch muss hier berücksichtigt werden, dass frühe Prototypen nur schwer hinsichtlich ihrer späteren Funktionalitäten in Sachen Qualität beurteilt werden können. Die Frage warum ein System schlecht oder ein Teilaspekt nicht gut genug ist, zielt darauf ab, Schwächen des Systems herauszufinden [36]. Die Fragen können nur in unterschiedlichen Stadien der Softwareentwicklung sinnvoll beantwortet werden. Der direkte Vergleich des Prototypen mit alternativen Systemen, sowie die Qualitätsmessung ist nur bei nahezu fertigen oder zumindest hinreichend entwickelnden Systemen und Prototypen realisierbar. Diese Art der Evaluation wird unter dem Begriff der *summativen Evaluation* untergeordnet. Bei der Entwicklung des Prototypen dieser Diplomarbeit muss fortlaufend ein iterativer Prozess stattfinden, bei dem kritische Schwachstellen des Systems aufgedeckt und die entsprechenden Teile modifiziert werden. Die Art der Evaluation wird unter dem Begriff der *formativen Evaluation* eingeordnet [35]. Die primäre Zielgruppe der Evaluationsteilnehmer kann aus gewöhnlichen Handy-Nutzern bestehen. Weiterhin können auch sogenannte Experten-Benutzer, die umfangreiches Wissen auf dem Gebiet von mobilen Endgeräten mitbringen, als Teilnehmer fungieren. Auf die Methodik der Evaluation und Auswahl der Teilnehmer wird in Abschnitt 5.1 näher eingegangen.

Die Rolle der Evaluation ist die Verbesserung beziehungsweise Optimierung von designkritischen Elementen, die Aufdeckung von Usability-Schwachstellen und die Verbesserung Programm-spezifischer Abläufe.

## Usability

Bei der Evaluation von Softwaresystemen stellt sich zu erst die Frage, nach welchen Kriterien das System evaluiert werden soll. Typische messbare Kategorien von Software sind unter anderem Effizienz, Qualität, Performance, Funktionalität und Usability [37]. In weiten Teilen, ausgenommen der Usability, zielen die anderen Kategorien auf technische Aspekte von Softwaresystemen ab [36]. Die Effizienz und Qualität von Software lässt sich nur sinnvoll bei funktionstüchtigen beziehungsweise implementierten Systemen messen. Unter dem Gesichtspunkt dieser Designstudie stellt sich die Frage, welche messbaren Kategorien für den späteren Entwicklungsverlauf interessant und vor allem notwendig sind.

Im Desktopbereich gibt es eine Vielzahl von facettierten Browsern (diskutiert in Kapitel 3). Sei es für den Konsumbereich ausgerichtete Systeme wie Ebay<sup>20</sup> oder Amazon<sup>21</sup> oder aus wissenschaftlichen Arbeiten hervorgegangene System wie The Flamenco Search Interface Project<sup>22</sup>. Im mobilen Bereich ist die Anzahl der facettierten Browser überschaubar. Bekannte Systeme sind mSpace-Mobile oder Mobile cultural heritage guide. Dieser Mangel an Systemen im mobilen Bereich bedarf einer genauen Untersuchung bei der Gestaltung von User-Interfaces.

Die Frage nach der Usability von mobilen User-Interfaces ist wichtig, da sie Aufschluss darüber gibt, wie effizient und vor allem benutzerfreundlich der Prototyp der Diplomarbeit ist. Mobile Endgeräte verfügen über spezielle Eigenschaften, einerseits in ihrer zugrunde liegenden Natur, wie physikalische Gegebenheiten (Größe, Gewicht, Bedienbarkeitsformen), sowie Eigenschaften hinsichtlich des Umgangs mit ihnen. Die Software sollte so gestaltet sein, dass eine hohe Gebrauchstauglichkeit und optimale Bedienung auch im mobilen Umfeld garantiert werden kann.

Die Entstehung eines qualitativ guten Designs soll dabei ein fortlaufender Prozess sein, in dem zwischen initialen Designs und letztendlich der fertigen Entwicklung Rückkopplungen [38] auftreten. Prozessphasen sind die Entwicklung von ersten Screen-Designs, die Usability Evaluation und anschließende Feedback-Gespräche mit den Beteiligten. Durch fortlaufende Gespräche mit den Testpersonen nach der Evaluation kann gewährleistet werden, dass gewonnene und umgesetzte Erkenntnisse bestätigt werden können.

---

<sup>20</sup><http://www.ebay.com>

<sup>21</sup><http://www.amazon.com>

<sup>22</sup><http://flamenco.berkeley.edu/demos.html>

## 5.1 Vorgehen

In diesem Abschnitt wird das Vorgehen dieser Designstudie und Evaluation diskutiert und die einzelnen Schritte detailliert beschrieben. Für die Gestaltung der Evaluation und deren Durchführung stellte sich die Frage, welche Herangehensweise und Methodik für diese Thematik die beste ist. Für die Evaluation von Softwaresystemen gibt es eine Vielzahl von Techniken, die zum Einsatz kommen. Man unterscheidet in deskriptive und vorausschauende Evaluations-Techniken [36]. Erstere Technik kann verwendet werden, um den aktuellen Status der Softwareentwicklung aufgrund von definierten Fragen und Aufgaben zu evaluieren. Die deskriptive Evaluation kann auf verschiedene Weise durchgeführt werden. Zum einen gibt es sogenannte Verhaltens-basierende Methodiken, in der der Benutzer mit dem Prototypen interagiert und gewisse Daten produziert. Das "think aloud protocol" ist einen der Methoden die zum Ziel hat, die Benutzerfreundlichkeit eines Systems herauszufinden [39]. Der Benutzer wird vor einen interaktiven Prototypen gesetzt, der noch nicht vollständig implementiert sein muss, und muss typische Aufgaben lösen. Es wird den Benutzer gefragt, was er sieht, denkt und fühlt, während er mit dem System arbeitet. Das Gesagte wird meist auf Video oder Audio gespeichert um später Designentscheidungen aufgrund der Aussagen der Teilnehmer zu treffen [40]. Diese Vorgehensweise setzt jedoch voraus, dass zumindest ein einfach implementierter Prototyp vorhanden sein sollte.

Für diese Evaluation bietet sich der so genannte Aufgaben-getriebener Designprozess an, da ein Prototyp erst im Anschluss der Evaluation implementiert wird. Im nächsten Paragraphen wird diese Vorgehensweise diskutiert.

### **Aufgaben-getriebener Designprozess**

Die Evaluationsmethodik, die hier verwendet wird, lässt sich unter dem Aufgaben-getriebenen Designprozess einordnen [39]. Der Prozess ist anhand spezieller Aufgaben an das System, die Benutzer während der Entwicklung bewältigen müssen, strukturiert. Anhand dieser Aufgaben, die schon in einer sehr frühen Prozessphase festgelegt werden, werden Designentscheidungen auf Basis der Aussagen der Teilnehmer durchgeführt. Doch wie werden diese Aufgaben ohne vorhandenen Prototypen von den Teilnehmern gelöst? Für diese Art der Evaluationsmethodik bieten sich Mockups an. Ein Mockup bezeichnet einen Prototypen der Benutzerschnittstelle. Mit Hilfe von Mockups lassen sich Anforderungen an die Benutzeroberfläche, sowie die Abbildung von programmiertechnischen Funktionen in einer frühen Entwicklungsphase visuell darstellen. Weiterhin bieten diese Mockups einen hohen Grad an

Flexibilität bei kurzfristigen Veränderungen. Designentwürfe lassen sich bequem und einfach umsetzen und können anschließend mit den Teilnehmern diskutiert und verglichen werden.

Für Designstudien bieten sich zwei Typen von Mockups an: ein rudimentär implementierter Prototyp oder ein Papier-basierter gedachter Durchlauf des Programms. Ersteres setzt voraus, dass im Vorfeld konkrete Vorstellungen der Benutzeroberfläche existieren. Zudem muss der Prototyp in der Funktionsvielfalt schon so hinreichend entwickelt sein, dass die vorgestellten Szenarien realistisch abgebildet werden können. Zweites ist für diese Arbeit relevant.

Die Aufgaben, die die Basis des Designprozesses bilden, wurden im Kapitel 2 definiert. Im folgenden wird auf die einzelnen Prozessphasen eingegangen.

Die Grundlage des Aufgaben-getriebenen Designprozesses bilden die Szenarien und ihre zugehörigen repräsentativen Aufgaben. In Kapitel 2 wurden zwei typische Szenarien herausgearbeitet, die für diese Diplomarbeit mit seinem Anwendungskontext von Relevanz sind. Die Szenarien Trivia und POIs-Suche stellen zwei unterschiedliche Anforderungen an den zu implementierenden Prototypen. Einerseits müssen aus den Ergebnissen der Designstudie funktionelle Anforderungen definiert werden (diskutiert in Abschnitt 6), andererseits muss ein starker Fokus an die Benutzeroberfläche gesetzt werden. Funktionen wie das Browsen und Suchen nach Facetten müssen sich sinnvoll in dem User-Interface widerspiegeln.

Das Identifizieren der Endbenutzergruppe ist der zweite Schritt des Aufgaben-getriebenen Designprozesses. Bei vielen prototypischen Implementationen im wissenschaftlichen Bereich besteht die Endbenutzergruppe nicht zwangsweise aus normalen Benutzern. Die Akzeptanzgrenze der Nutzer im wissenschaftlichen Bereich ist oft höher als bei Benutzern, die zum Beispiel solche Systeme im Alltag einsetzen möchten. Das Ziel dieser Arbeit ist es den Prototypen soweit zu implementieren, so dass ein gewisser Grad an Usability erreicht wird. Es muss davon ausgegangen werden, dass auch normale Endbenutzer (z.B. die ihr Handy für den täglichen normalen Gebrauch verwenden), die vielleicht nicht Experten auf dem Gebiet von mobilen Endgeräten sind, problemlos mit dem Prototypen umgehen können.

Anhand der Szenarien aus Kapitel 2 und der Definition der Endbenutzergruppe, müssen repräsentative Aufgaben hinsichtlich der Benutzeroberfläche erstellt werden. In Abschnitt 5.3 geht es um die spezifischen Teilaufgaben. Sie beschreiben einerseits die Szenarien im Detail und repräsentieren grundlegende Funktionalitäten des Systems. Der erste große Aufgabenblock besteht aus der Durchführung der beiden Szenarien. Dabei werden aus den Szenarien einzelne Aufgaben abgeleitet, die in der Durchführung der

Evaluation von den Teilnehmern bewältigt werden. Der zweite Aufgabenblock befasst sich vielmehr um einzelne designkritische Elemente in Bezug auf die Verwendung von Oberflächenkomponenten. Dabei müssen die Teilnehmer entscheiden, welche subjektiven Vorlieben sie haben und welche Komponenten ihrer Meinung nach mehr Sinn machen.

Der letzte Schritt befasst sich mit der Durchführung und Auswertung der Evaluation; Also der Präsentation der Szenarien und ihren zugehörigen Aufgaben, die Beantwortung von Fragen und die Auswertung dieser.

## 5.2 Testpersonen

Die Durchführung der Evaluation erfolgte in einer kleinen Gruppe mit fünf Teilnehmern. Dabei handelt sich um Teilnehmer, die zwischen 26 und 35 Jahren alt sind. Vier der fünf Teilnehmer sind männlich. Der Hintergrund für die Wahl einer solchen kleinen Gruppe liegt in der Methodik der Durchführung. Die Präsentation der Mockups und deren Aufgaben erfolgte sehr ausführlich. Im Anschluss der Präsentation konnte somit ausführlich auf Kommentare und Verbesserungsvorschläge der Teilnehmer eingegangen werden. Bei dieser Evaluation spielt die klassische Generierung von großen Datenmengen eine untergeordnete Rolle. Umfangreiche Kommentare, Anmerkungen und Verbesserungsvorschläge haben hier eine größere Bedeutung, da diese für weitere Designentwicklungen sehr hilfreich sind.

Fünf Testpersonen aus der Universität Koblenz aus dem Bereich Informatik nahmen bei dieser Evaluation teil. Bei zwei der Teilnehmer handelt es sich um Personen aus dem Bereich der Computervisualistik. Im Bereich der Usability und Gestaltung von Benutzeroberflächen haben sie fundiertes Wissen mitgebracht und konnten in der Diskussion hilfreiche Kommentare zu einzelnen Designentscheidungen geben. Lediglich ein Teilnehmer schätze sich als Experte im Bereich von mobilen Endgeräten ein. Maßgeblich für die Evaluation war aber nicht ein Expertenwissen im Bereich der mobilen Endgeräte, vielmehr spielte ein gutes Verständnis im Bereich der Usability von Benutzeroberflächen eine Rolle.

## 5.3 Aufgaben

In diesem Abschnitt geht es um die Teilnehmeraufgaben, die im Rahmen der Evaluation durchgeführt werden. Die Szenarien aus Kapitel 2 bilden die Grundlage für den ersten Aufgabenblock. Anhand der Szenarien können einzelne Schritte zur Bewältigung der Gesamtaufgabe, die beispielsweise im ersten Szenario *POI-Suche* das Suchen nach umliegenden Sehenswürdigkeiten umfasst, definiert werden. Doch wie werden diese Aufgaben letztendlich von

den Teilnehmern bewältigt? Die Aufgaben werden nicht mittelbar von den Teilnehmern selbst gelöst, vielmehr müssen die Teilnehmer anhand von präsentierten Mockups das Szenario verstehen und die einzelnen Schritte nachvollziehen.

Der zweite Aufgabenblock befasst sich mit Detailfragen zum User-Interface. Dabei spielen Design-kritische Elemente, wie die Verwendung von Tableisten oder Menüs eine wichtige Rolle. Hier bekommen die Teilnehmer verschiedene Designbeispiele gezeigt anhand deren sie entscheiden, welche die Beste und Sinnvollste ist.

### 5.3.1 Szenarien POI-Suche & Trivia

**POI-Suche** Rückblickend auf das erste Szenario *POI-Suche* aus Abschnitt 2 handelt es sich hierbei um eine typische Suche nach Orts-bezogenen Informationen. Aus dem Szenario lässt sich eine logische Trennung zweier Aufgabenpakete feststellen: die Suche nach vorhandenen kulturellen Informationen in der unmittelbaren Umgebung und die Suche nach nächstgelegenen Restaurants. Im Folgenden sind die Aufgaben des Szenarios *POI-Suche* aufgeführt.

#### Aufgaben 1

1. Sehenswürdigkeiten der Stadt herausfinden
2. Detailinformationen der Festung Ehrenbreitstein erfahren
3. Bilder zur Festung anschauen

#### Aufgaben 2

1. Ein Restaurant finden

**Trivia** Das zweite Szenario *Trivia* repräsentiert im Gegensatz zum ersten Szenario (POI-Suche) eine typische Alltagssituation, in der willkürliche Fragen bei Gesprächen unter Freunden oder Kollegen auftauchen können. Das Szenario lässt sich im Kontext "Informationsbedarf bei mobilen Anwendungen" unter *Trivia* (siehe 2.1 auf Seite 4) einordnen.

#### Aufgaben 1

1. Gebäude der Stadt finden
2. Detailinformationen über die Höhe des Fernsehturmes von Berlin finden

## Aufgaben 2

1. Kontext Ort „Berlin“ nach Ort „Koblenz“ ändern
2. Detailinformationen über die Höhe des Turmes in Koblenz finden
3. Die Position des Turmes einsehen

### 5.3.2 Detailfragen

Der zweite Aufgabenblock behandelt, unabhängig von einem konkreten Szenario, designkritische Elemente in der User-Interface Gestaltung. Dabei wurden anhand der ersten Mockups drei verschiedene Aspekte herausgefunden, die hinsichtlich der Interaktion mit der Applikation, sowie deren Nachvollziehbarkeit eine wichtige Rolle spielen. Im Folgenden sind die Detailspekte aufgeführt:

#### 1. Verwendung von Tabs / Menüs

Hier geht es um die Frage nach der generellen Strukturierung der Applikation. Also inwiefern der Benutzer Schritt für Schritt seine Interaktion ausführen muss, um an die gewünschte Information zu gelangen. Gerade die Verwendung von Tableisten oder Menüs können die grundlegende Programmstruktur erheblich beeinflussen. Benutzeraufgaben hier sind unter anderem, die individuellen Vorlieben bei der Verwendung von Tabs oder Menüs der Teilnehmer herauszufinden. Bei dem Mockup-Design von Szenario POI-Suche und Trivia wurde jeweils die Navigation per Menü und die Navigation per Tableisten realisiert. Somit bietet sich für die Teilnehmer ein direkter Vergleich dieser beiden Designelemente.

#### 2. Darstellung der Facettenauswahl

Die allgemeine Darstellung von Facetten stellt eine wichtiger Aspekt bei der Gestaltung des User-Interfaces dar. Die Teilnehmer bekommen hier verschiedene Designansätze gezeigt und müssen entscheiden, welche der Ansätze ihrer Meinung nach der sinnvollste ist. Weiterhin sind Fragen zum allgemeinen Verständnis von Facetten in diesem Aufgabenblock enthalten.

#### 3. Browsen und Suchen

Hier geht es um die Frage, wie der Benutzer mit den Facetten interagieren kann. Als Ausgangspunkt wird zum einen das System der Universität Amsterdam (Mobile cultural heritage guide<sup>23</sup> [29]) und

---

<sup>23</sup>Mobile cultural heritage guide: location-aware semantic search

der Applikation mSpace Mobile [27] genommen, zum anderen eigene Lösungsansätze aus Kapitel 4 gezeigt. Aufgabe der Teilnehmer ist es, verschiedene Ansätze hinsichtlich Benutzbarkeit und Nachvollziehbarkeit zu vergleichen. Weiterhin spielt hier neben der allgemeinen Interaktion mit den Facetten, die Suche nach deren eine wichtige Rolle.

## 5.4 Auswertung und Analyse

Im Folgenden der komplette Fragenkatalog, der für die erste Evaluation verwendet wurde. Im Anschluss werden wichtige Aspekte analysiert und diskutiert. Der Fragenkatalog ist in mehrere Blöcke unterteilt: Allgemeine Fragen zu den Teilnehmern, Fragen zum den Szenarien POI-Suche und Trivia, sowie Detailfragen zur Verwendung von Tableisten und Menüs, das Browsen über Facetten und die Darstellung und Verständnis von ausgewählten Facetten<sup>24</sup>. Der letzte Fragenblock bildet eine Art Wunschliste, die den Funktionsumfang der Applikation erweitern soll.

### Fragenkatalog der Designstudie

Nr.	Allgemeine Fragen
1.1	Alter
1.2	Geschlecht
1.3	Wie gut kennen Sie sich allgemein mit mobilen Endgeräten aus?
1.4	Besitzen Sie ein mobiles Endgerät, wenn ja wie viele?
1.5	Welches Endgerät besitzen Sie?
1.6	Wie oft am Tag benutzen Sie Ihr Gerät? (in Minuten)
1.7	Ich weiss über die neuesten Handyentwicklungen im Bezug auf Technik bescheid
1.8	Mit welchen Interaktionsformen haben Sie bereits Erfahrung?

<sup>24</sup>In den Fragen werden Bezüge zu bestimmten Mockups gemacht. Die zugehörigen Mockups lassen sich in der beigelegten CD einsehen. Der Übersicht halber sind diese hier nicht aufgeführt.

<b>Nr.</b>	<b>Detailfragen zur POI-Suche</b>
2.1	Konnten Sie die einzelnen Schritte in Bezug auf das Suchen nach Points of Interests nachvollziehen?
2.2	Beurteilen Sie: Ist die Navigation über das virtuelle Menü nachvollziehbar?
2.3	Finden Sie im allgemeinen die Navigationsstruktur der Applikation nachvollziehbar?
2.4	Das Browsen über Facetten wird mit einzelnen Listen realisiert. Können Sie diese Navigationsform nachvollziehen?
2.5	Die Kartenansicht zeigt gefundene Ressourcen als kleine Hinweissymbole mit jeweiligem Titel. Könnte die Übersicht verloren gehen, falls sehr viele Ressourcen gefunden werden?
2.6	Beim Anklicken einer Ressource auf der Kartenansicht öffnet sich ein Dialog um Details anzuzeigen oder nicht. Würden Sie eine direkte Detailansicht der Ressource präferieren? (ohne Umweg über einen Dialog)
2.7	Ist Ihnen ersichtlich, dass Facetten und Ressourcen nur in der umliegenden Umgebung gesucht werden?
2.8	Bei dem virtuellen Menü befinden sich kleine Symbole neben dem Text. Finden Sie Verwendung von Symbolen im allgemeinen hilfreich?
<b>Nr.</b>	<b>Allgemeine Fragen zum Szenario: POI-Suche</b>
2.9	Ist das gezeigte Szenario „Sightseeing“ realistisch und authentisch?
2.10	Haben Sie selbst schon einmal eine ähnliche Situation erlebt?

<b>Nr.</b>	<b>Detailfragen zu Trivia</b>
3.1	Konnten Sie die einzelnen Schritte zur Suche nach Informationen zu Fernsehtürmen nachvollziehen?
3.2	Finden Sie im allgemeinen die Navigationsstruktur zur Suche nach ortsunabhängigen Informationen angemessen?
3.3	Im Gegensatz zum Szenario 1 wurde hier mit einer Tableiste anstatt eines Menüs gearbeitet. Beurteilen Sie: Vereinfacht die Verwendung von Tableisten die Navigation?
3.4	Erachten Sie eine Live-Vorschau bei der Suche im allgemeinen für eine hilfreiche Funktion?
3.5	In Schritt 8 werden gefundene Facetten angezeigt. Haben Sie hier eher eine Ergebnisliste erwartet?
3.6	Der Location Kontext wird in den Schritten 11-12 über eine Suche von Berlin nach Koblenz geändert. Finden Sie diese Vorgehensweise kompliziert?
3.7	Im Tab „Facets“ wird neben den Facetten noch die zuvor ausgewählten Facetten am oberen Bildschirmrand angezeigt. Halten Sie das für sinnvoll?
3.8	Wünschen Sie sich eine Historie der zuletzt gesuchten Begriffe?
<b>Nr.</b>	<b>Allgemeine Fragen zum Szenario: Trivia</b>
3.9	Ist das gezeigte Szenario „Social Lunch“ realistisch und authentisch?
3.10	Haben Sie selbst schon einmal eine ähnliche Situation erlebt?
<b>Nr.</b>	<b>Verwendung von Tabs</b>
4.1	Bevorzugen Sie allgemein eine Applikation mit Tableisten? (1.1,1.3)
4.2	Bewahren Tableisten ihre Meinung nach die Übersicht einer Applikation?
4.3	Tableisten verkleinern zum Beispiel die Kartenansicht. Empfinden Sie das als einen Nachteil?
4.4	In Beispiel 1.3 könnte man noch das virtuelle Menü ausklappen. Die Tableiste wäre somit nicht mehr sichtbar. Würden Sie das als Störend empfinden?
4.5	Bevorzugen Sie im allgemeinen eine Tableiste am oberen oder unterem Bildschirmrand?
4.6	Welche der folgenden Funktionen sollten per Tabs erreichbar sein? Kreuzen Sie jeweils ja oder nein an!

<b>Nr.</b>	<b>Darstellung und Verständnis von Facetten</b>
5.1	Ist es hilfreich, wenn die zuvor ausgewählten Facetten als fester Bestandteil auf dem Bildschirm einzusehen sind? (2.1 , 2.3)
5.2	Bei den Beispielen 2.2, 2.4 und 2.5 lassen sich die Facetten Places, Populated Places und Building jeweils unabhängig voneinander abwählen(bzw. löschen). Ist es ersichtlich, dass Populated Places und Building Subfacetten von Places sind?
5.3	Was denken Sie wie die Ergebnismenge von Beispiel 2.5 aussieht? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.4	Was denken Sie was passiert, wenn man bei Beispiel 2.5 die Facette Populated Places deaktiviert? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.5	Beispiel 2.1 und 2.3 zeigen die Facetten in einer hierarchischen Ansicht. Was denken Sie passiert, wenn man den Knopf neben der Facette Building anklickt? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.6	Anhand der gezeigten Beispiele: Für welche Darstellung von Facetten würde Sie sich entscheiden?
<b>Nr.</b>	<b>Browsen über Facetten</b>
6.1	In Beispiel 3.1 werden Facetten und ihre zugehörigen Subfacetten farblich getrennt. Beurteilen Sie: Ist eine solche farbliche Trennung hilfreich für die Navigation?
6.2	Finden Sie die räumliche Aufteilung des Bildschirms ansprechend? (3.2)
6.3	Applikation 3.2 wird über einen Stift bedient. Könnten Sie sich vorstellen eine solche Applikation mittels eines Touchscreens zu bedienen?
6.4	Finden Sie die Listendarstellung der verfügbaren Facetten als eine angemessene Darstellungsform? (3.3)
6.5	In Beispiel 3.3 werden Facetten und ihre zugehörigen Subfacetten über Listen realisiert, die den kompletten Bildschirm einnehmen. Könnte man hier schnell die Übersicht verlieren?
6.6	Finden Sie ein Dropdown-Menüs als eine angemessene Darstellungsform? (3.4)
6.7	Könnte man bei Beispiel 3.5 schnell den Überblick verlieren?
6.8	Beurteilen Sie: Ist die Schriftgröße in der Listenansicht bei Beispiel 3.5 in Ordnung?
6.9	Anhand der gezeigten Beispiele: Für welche Form des „Browsen über Facetten“ würden Sie sich entscheiden? Kreuzen Sie an!

Nr.	Funktionsumfang der Applikation
7.1	Wünschen Sie sich eine Kartenansicht um Informationen anzeigen zu lassen?
7.2	Wünschen Sie sich eine Ergebnisliste der gefundenen Informationen?
7.3	Wünschen Sie sich eine Navigationsfunktion?
7.4	Wünschen Sie sich eine Augmented Reality View? (Siehe Mobile Cultural Heritage)
7.5	Wünschen Sie sich eine Historie der zuletzt gesuchten Begriffe?
7.6	Wünschen Sie sich eine Favoritenverwaltung?

#### 5.4.1 Diskussion des Szenarios: POI-Suche

Das Szenario POI-Suche wurde von allen Teilnehmern als realistisch und authentisch eingestuft. Vier der insgesamt fünf Teilnehmer haben selbst schon mal eine ähnliche Situation erlebt. Das deutet darauf hin, dass so ein Szenario typische Alltagssituationen widerspiegelt. Generell kann man sagen, dass die Suche nach ortsbasierenden Informationen eine sehr wichtige Funktion des zu implementierenden Prototypen sein wird.

Die Navigationsstruktur des ersten Szenarios wurde im Gegensatz zum zweiten Szenario mithilfe eines Menüs realisiert. Die Fragen 2.2 und 2.3 zielten darauf ab, ob die Teilnehmer die Navigationsstruktur innerhalb der Applikation nachvollzogen wurde. Es ging darum herauszufinden, ob die Navigationsschritte schlüssig sind. Weiterhin sollten sich die Teilnehmer in der Applikation nicht "verloren" fühlen. Die Navigation per Menü wurde von der Hälfte der Teilnehmer als mittelmäßig und von der anderen Hälfte ein wenig besser eingestuft. Die allgemeine Navigationsstruktur (unabhängig des Menüs) wurde insgesamt von den Teilnehmern während der Präsentation, sowie in der anschließenden Diskussion als unpraktisch wahrgenommen.

Die Listendarstellung für das Browsen über Facetten wurde von den Teilnehmern als eine gute Darstellungsform erachtet. Jedoch stellte sich heraus, dass die Navigation innerhalb der Facetten, wenn man sich hierarchische Facetten vorstellt, nicht optimal nachvollzogen wurde. Hier muss man bei der Implementierung darauf achten, dass eine hinreichende optische Visualisierungsform vorhanden ist, so dass der Benutzer eindeutig weiß, in welcher Facette er sich gerade befindet.

Die Kartenansicht als Visualisierungsform von gefundenen Ergebnissen wurde von den Teilnehmern positiv aufgenommen. Jedoch waren sich alle einig, dass bei zu vielen Informationen auf der Karte die Übersicht schnell verloren gehen kann. Ein Clustering der Ergebnisse beziehungsweise eine andere Darstellungsform dieser ist von Nöten. Dass die Applikation nur Facetten

und Ergebnisse in der Umgebung sucht, wurde von vier der fünf Teilnehmer als ersichtlich empfunden. Lediglich eine Person konnte dies nicht bestätigen. Weiterhin stimmten 75% Teilnehmer voll zu, dass beim Anklicken eines Ergebnisses auf der Kartenansicht der Umweg über einen Dialog unnötig wäre.

Symbole als Hilfsmittel bei der Menüführung wurde insgesamt als ein hilfreiches Designelement gesehen. Das lässt darauf schließen, dass die Menüsymbole die zugrunde liegende Funktion visuell gut repräsentieren.

#### **5.4.2 Diskussion des Szenarios: Trivia**

Das zweite Szenario behandelte das Thema Social-Lunch und wurde von 75% der Teilnehmer als realistisch und authentisch eingestuft. Jedoch hat im Gegensatz zum ersten Szenario nur etwa die Hälfte eine ähnliche Situation erlebt. Bei der anderen Hälfte der Teilnehmer traf dies kaum zu. Für den späteren Prototypen gehört dennoch die grundsätzliche Suche nach ortsunabhängigen Informationen zu einer sehr wichtigen Funktion. In Anbetracht dessen, dass nur die Hälfte der Teilnehmer eine ähnliche Situation hatten, sieht man an der Studie aus [9], dass bei sozialen Konversationen häufig Interessenfragen auftreten können. Die Realisierung dieser Suche wird anhand einer sinnvollen Darstellung der Facetten durchgeführt, die in Abschnitt 5.4.3 diskutiert wird.

Die allgemeine Nachvollziehbarkeit der einzelnen Schritte zur Suche nach ortsunabhängigen Informationen ist tendenziell schlechter ausgefallen. Im Gegensatz zum Szenario POI-Suche, wurden die einzelnen Navigationsschritte insgesamt schlechter nachvollzogen. Das kann an der völlig anderen Navigationsphilosophie über Tableisten statt über ein Menü liegen.

Bei der Suche nach Informationen mithilfe eines Eingabefeldes war eine Live-Vorschau der zu suchenden Begriffe integriert. Dabei handelt es sich um eine Funktion, die beim Eintippen eines Suchbegriffes passende Treffer im Voraus liefert. Der Benutzer kann somit einerseits das Suchwort schneller finden, andererseits kann sie sich als hilfreich erweisen, wenn der Benutzer den Suchbegriff nur vage oder unvollständig kennt. Diese Live-Vorschau wurde von der Mehrheit der Teilnehmer als eine sehr hilfreiche Funktion erachtet. Lediglich eine Person empfand diese Funktion als nutzlos. Für den Prototypen muss herausgefunden werden, inwiefern sich eine sinnvolle Live-Vorschau auf semantischen Daten realisieren lässt. Eine nennenswerte Frage war, die Frage nach dem Wechsel des Ortskontextes von Berlin nach Koblenz. Hier stellte sich heraus, dass über 75% der Teilnehmer die vorgestellte Herangehensweise als kompliziert angesehen haben. Hier besteht für die Implementation Verbesserungsbedarf.

Das Manko im ersten Szenario, dass die Navigation innerhalb der Facetten nicht ersichtlich genug war, wurde im zweiten Szenario in der Hinsicht verbessert, dass ausgewählte Facetten am oberen Bildschirmrand dargestellt werden. 50% halten das für eine sinnvolle und 25% sogar für sehr sinnvolle Lösung. Das spiegelt auch die Erfahrungen im Umgang mit Facetten aus dem Desktopbereich wieder [2, 32]. Mehr zum Thema in Kapitel 3. Die Kombination von gefundenen Ergebnissen und deren Darstellung und Umgang ist eine der Schlüsselfragen im Designprozess. Für den Benutzer muss direkt ersichtlich sein, in welcher Facettenhierarchie er sich befindet und welche Facetten er zuvor ausgewählt hat. Gerade bei einer großen Anzahl von Facetten kann der Benutzer schnell den Überblick verlieren. Daher muss für den Prototypen eine sinnvolle Integration der Facettenansicht berücksichtigt werden. Der Fragenblock *Darstellung und Verständnis von Facetten* aus dem Abschnitt Detailfragen 5.4.3 behandelt diesen Aspekt ausführlicher.

Die Suche nach Informationen über ein Suchfeld sahen alle Teilnehmer als eine sinnvolle und gängige Methode. Jedoch kann man sagen, dass eine Historie über die zuletzt gesuchten Begriffe nicht gewünscht wurde. Zur Suche ist weiterhin anzumerken, dass ein extra Tab- beziehungsweise Menüeintrag als umständlich gesehen wurde. Hier bedarf es für den Prototypen eine Integration der Suchmaske in den Bereich des Browsens.

### 5.4.3 Diskussion der Detailfragen

Bei den Detailfragen ging es vielmehr um Entscheidungen bei designkritischen Elementen des User-Interfaces, anstatt das allgemeine Verständnis um Navigationsstrukturen oder die Nachvollziehbarkeit von Aktionen. Den Teilnehmern wurde ein konkreter Vergleich der verschiedenen Vorschläge gezeigt, so dass sie das nach ihrer Ansicht beste Verfahren ermitteln konnten.

### Verwendung von Tableisten

Im ersten Detailblock ging es um die Frage nach der Verwendung von *Menüs* aus Szenario POI-Suche oder die Verwendung von *Tableisten* aus dem Szenario Trivia. Die Tendenz ist hier zweigeteilt. Die Hälfte der Teilnehmer erachten die Verwendung von Tableisten nicht für sehr sinnvoll, während die andere Hälfte diese Art der Darstellungsform für sinnvoll und hilfreich sehen. Die Übersicht einer Applikation kann mittels Tableisten laut den Teilnehmern nur mittelmäßig bewahrt werden. Die Mockup-Designs zeigen zwei verschiedene Varianten über die Platzierung der Tableiste innerhalb des Bildschirmbereiches. 25% finden die Anordnung der Tableiste am unteren Bildschirmrand besser als am oberen Bildschirmrand. Die Tendenz zum obe-

ren Bildschirmrand kann zustande kommen, da ein ausgeklapptes virtuelles Menü eine untere Tableiste verdecken würde, welches auch von 75% der Teilnehmer als störend empfunden wird.

Für den zu implementierende Prototypen wird eine Kombination beider Navigationsformen eine tragende Rolle spielen. Die Navigationsstruktur muss einfach und klar gestaltet werden. Weiterhin sollten die Menüeinträge nur die wichtigsten Funktionen beinhalten. Da Tableisten die Übersicht einer Applikation bewahren, wird für die Implementation eine obere Tableiste mit voraussichtlich folgenden Einträgen realisiert werden:

- Visualisierungansicht: (Dynamischer Darstellung je nach Inhalt und Facettentyp, z.B. Ort, Events, Personen,...)
- Facettenansicht: Hier findet die Interaktion des Benutzers mit zuvor ausgewählten Facetten statt
- Browse-Ansicht: Hier kann der Benutzer nach Facetten oder Inhalten suchen, sowie die Datenmenge durch gezielte Auswahl von Facetten einschränken

Die geringe Anzahl von drei Tabeinträgen führt dazu, dass die Komplexität der Navigation innerhalb der Applikation abnimmt. Dem Benutzer sollte es so ersichtlicher sein, welche Funktionen ihm zur Verfügung stehen.

## **Darstellung und Verständnis von Facetten**

Bei dem Fragenblock *Darstellung und Verständnis von Facetten* kam in der Diskussion mit den Teilnehmern heraus, dass viele das Verständnis der hierarchischen Facetten nicht vollständig nachvollziehen konnten. Hier ist es wichtig, dass Facetten mit ihren zugehörigen Subfacetten sinnvoll dargestellt werden müssen. Neben der Darstellung der ausgewählten Facetten spielt die Interaktion mit solchen eine wichtige Rolle. Bei der Darstellungsform für den Prototypen muss überlegen, welche der drei in Abbildung 16 auf Seite 50 gezeigten Darstellungsweisen von hierarchischen Facetten die Bessere ist.

Die erste Darstellung führte zu Verwirrung, da zu viele Informationen dargestellt werden. Je nach Facettentiefe von *Place* kann die Liste sehr lang werden. Wenn zusätzlich noch eine weitere Facette *Person* mit zugehörigen Subfacetten vorhanden ist, kann der Bildschirmplatz knapp werden. Die zweite Darstellung blendet Zwischenfacetten mittels einer Klammer aus. Diese Art der Darstellung bewahrt die Übersicht und gleichzeitig sieht der Benutzer, in welcher High-Level Facette er sich befindet. Die dritte Darstellung zeigt nur die unterste Facettenebene und lässt dem Benutzer

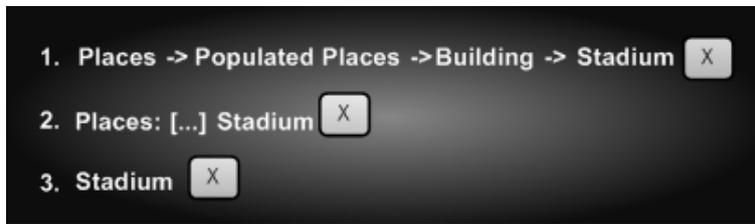


Abbildung 16: Varianten um ausgewählte Facetten darzustellen

keinen großen Interpretationsspielraum. Die Teilnehmer haben die erste Darstellungsform in der Hinsicht interpretiert, dass nach *Places*, *Populated Places*, *Building* und *Stadium* gesucht wird. Jedoch wird im Endeffekt nur nach *Stadium* gesucht. Dieser Missstand wird in der letzten Darstellungsart behoben.

Neben der Darstellung, die gerade diskutiert wurde, spielt die Interaktion mit diesen ausgewählten Facetten eine wichtige Rolle. Wie interagiert der Nutzer mit diesen und welche Auswirkung hat das auf die Ergebnismenge? Alle drei Beispiele besitzen einen kleinen Knopf rechts neben dem letzten Eintrag. Er dient dazu, um ausgewählte Facetten zu löschen.

Die Listendarstellung aus Frage 5.6 wurde von einer großen Mehrheit der Teilnehmer als ganz schlechte Visualisierungsform für hierarchisch angeordnete Facetten gesehen. Für die Implementation ist es wichtig, dass zum einen zusammengehörige Facetten visuell sinnvoll - und zum anderen, dass das Abwählen und Hinzufügen von Facetten logisch dargestellt wird.

### Browsen über Facetten

Die nächsten Detailfragen beschäftigten sich mit dem allgemeinen *Browsen über Facetten*. Als vorhandene Basis wurden die aus der Abbildung 3 auf Seite 19 gezeigten Systeme genommen. Zudem wurden eigene Ansätze gezeigt. MSpace Mobile kam mit seiner unübersichtlichen Bildschirmaufteilung sehr schlecht davon. Der Bildschirm ist hier in sechs kleine Kacheln unterteilt, in der jede Kachel eine gewisse Funktion repräsentiert (Facetten und Subfacetten, Kartenansicht, Detailansicht, usw. ). Diese feine Unterteilung des Bildschirms empfanden die Teilnehmer als unpraktisch und ist zudem für eine Touchscreen-basiertes System nur schwer bedienbar. Die Listendarstellung aus Beispiel 5 kam bei über 75% Teilnehmer gut an. Eine Liste mit ausreichend großen und gut lesbaren Listenelementen ist für den zu implementierenden Prototypen sicher eine gute Alternative. Jedoch stellte sich bei der anschließenden Diskussion heraus, dass man bei der

Listendarstellung den Überblick verlieren kann.

#### 5.4.4 Weitere Anmerkungen

Es kamen einige Unklarheiten in der Terminologie der verwendeten Begriffe auf. In den Mockups wurde der Begriff *Facette* und *Filter* teilweise für die gleichen Sachverhalte verwendet, welches bei den Teilnehmern zu Verwirrung führte. Es kam die Frage auf, warum man einen *Filter* abwählen muss, um Ressourcen zu finden. Hier besteht ein Verbesserungsbedarf hinsichtlich der zu verwendeten Begriffe. Für den Prototypen wird der Begriff der *Facette* festgelegt. Weiterhin muss für den Benutzer klar ersichtlich sein, inwiefern der Begriff *Facette* Auswirkung auf die Ergebnismenge hat.

Bei der Kartenansicht wurde bemängelt, dass die Übersicht verloren gehen kann, falls sehr viele Ressourcen gefunden werden. Gerade wenn der Benutzer aus der Karte herauszoomt, überlappen sich dicht besiedelte Ressourcen. Eine Lösung ist ein visuelles Clustering, in dem kleine Gruppierungen von Ressourcen vorgenommen werden. Weiterhin wurde auf der Kartenansicht ein kleiner Hinweistext "Found facets" gezeigt, falls Facetten in der Umgebung gefunden wurden. Die Nachricht, dass "Facetten" gefunden wurden, war für die Teilnehmer ungewöhnlich. Besser wäre eine Hinweismeldung "Found POI".

Bei der Detailansicht von Ressourcen kam die Frage auf, warum ein "More" Knopf gedrückt werden muss, um sich zum Beispiel Fotos zu dieser Ressource anzeigen zu lassen. Hier kann eine Integration von Medien wie Fotos und Videos in die Detailansicht vorgenommen werden. Alle wichtigen Informationen zu einer Ressource werden in einer einzelnen Ansicht für den Benutzer zur Verfügung gestellt.

Ein wichtiger Diskussionspunkt war die generelle Unterscheidung von "Search" und "Browse Facets". Die Suchfunktion wurde im zweiten Szenario Trivia für die Suche nach Ressourcen verwendet. Generell muss es dem Benutzer bei einer Suchfunktion ersichtlich sein, nach welchen Informationstypen gesucht wird. Ob es eine Suche nach Facetten oder nach Ressourcen ist, muss klar definiert werden. Für den Prototypen dieser Arbeit besteht der Anspruch, dass eine Suche nach Facetten, sowohl auch eine Suche nach Inhalten implementiert wird. Bei der Darstellung der Ergebnisse kann man sich eine integrierte Ansicht vorstellen, in der die gefundenen Facetten sowohl auch die Instanzen derer dargestellt werden. Die Suchfunktion als Solche wurde in den Mockups mittels eines typischen Suchfeldes realisiert, welches unabhängig von der Ansicht "Browse Facets" realisiert war. Für den Prototypen muss man eine etwaige Integration des Suchfeldes in die Browse-Ansicht in Betracht ziehen.

Um eine neue zusätzliche Facette auszuwählen, war in den Mockups keine Möglichkeit vorhanden. Für das spätere Design sowie der Implementation würde sich hier eine "Add facets" Funktionalität anbieten. Beispielsweise hat ein Benutzer die Orts-Facette hinzugefügt und möchte danach noch nach Personen browsen. Mehrauswahl von High-Level Facetten spielt hier eine wichtige Rolle.

Viele haben die einfache Listendarstellung beim Browsen über Facetten als eine gute Visualisierungsform gesehen. Jedoch hat diese den Nachteil, dass zum Beispiel Subfacetten nicht sofort eingesehen werden können. Man muss erst eine Hierarchie-Ebene absteigen, um sich die Subfacetten anzeigen zu lassen. Hier würde sich eine Vorschau-Ansicht in Form eines kleinen Pop-Ups anbieten, in der eine kleine Übersicht der Subfacetten angeboten wird.

## 6 Anforderungsanalyse

In diesem Kapitel geht es um die Anforderungen die an das System gestellt werden. Dabei wird in funktionale und nichtfunktionale Anforderungen unterschieden. Weiterhin sind die Anforderungen gruppiert angeordnet. Generalisierte User-Anforderungen werden durch System-Anforderungen erweitert. Die Priorität der Anforderungen wird durch ein (+) und ein (-) Zeichen gekennzeichnet und dienen zur Markierung einer 'Muss' und 'Kann' Anforderung [41]. Die Anforderungsanalyse basiert größtenteils aus den gewonnenen Erkenntnissen aus der Designstudie. Anhand der ersten Designentwürfe und den umfangreichen Auswertungen der Fragebögen haben sich signifikante Anforderungen einerseits an die funktionalen Anforderungen, andererseits auch an die Usability der Applikation ergeben. Die funktionalen Anforderungen lehnen sich zudem stark an bekannte und in der Praxis gängige facettierte Browseranwendungen an [15,33,34]. Die Usability, im Abschnitt Nicht-funktionale Anforderungen 6.2 angesiedelt, zielt darauf ab, die allgemeine Benutzerfeindlichkeit des Systems zu bewahren. Der Anwender sollte die Client-Anwendung verstehen und sich nicht verloren fühlen. Weiterhin soll das facettierte Browsen so bequem wie möglich auf einem mobilen Endgerät zu bedienen sein.

Im Folgenden werden zuerst die funktionalen Anforderungen aufgelistet, im Anschluss die nicht-funktionalen Anforderungen mit Hauptaugenmerk auf die Usability.

### 6.1 Funktionale Anforderungen

Diese Sektion behandelt die funktionalen Anforderungen des Clients und des Servers. Der Client ist eine Android-Applikation, die auf einem mobilen Endgerät installiert ist. Die generelle Aufgabe des Clients ist das facettierte Browsen und Suchen nach semantischen Daten. Der Server ist für die verteilte Datenbeschaffung, die Datenaufbereitung und die Bereitstellung der Daten für den Client verantwortlich.

#### 6.1.1 Client

Der Client ist eine Android-basierte Applikation, die die Funktion des facettierten Browsens bereitstellt. Der Benutzer hat die Möglichkeit mit Hilfe von verfügbaren Facetten seinen Informationsbedarf zu decken. Der Benutzer kann zum Einstieg verfügbare High-Level Facetten auswählen und durch diese die Ergebnismenge nach Belieben einschränken. Für gefundene Ressourcen bietet der Client verschiedene Darstellungsmöglichkeiten, wie beispielsweise

eine Karten- oder Listenansicht. Die visuelle Darstellung der Ergebnisse ist abhängig von dem jeweiligen Facettentyp, wie beispielsweise Ort, Events oder Personen. Die Grundlagen des facettierten Browsens lassen sich in Kapitel 3 einsehen und nachvollziehen.

- R1<sup>+</sup>** Benutzer können für den Einstieg High-Level Facetten selektieren
- R2<sup>+</sup>** Durch Auswählen einer High-Level Facette werden die Informationen soweit gefiltert, so dass nur die jeweiligen relevanten Informationen angezeigt werden
- R3<sup>+</sup>** Benutzer können eine Mehrhauswahl von High-Level Facetten tätigen
- R4<sup>+</sup>** Benutzer können bei hierarchisch angeordneten Facetten Subfacetten im Voraus einsehen
- R5<sup>+</sup>** Es wird die Anzahl der Instanzen pro Facetten angezeigt
- R6<sup>+</sup>** Benutzer haben die Möglichkeit, zuvor ausgewählte Facetten einzeln- oder komplett abzuwählen
- R7<sup>-</sup>** Benutzer können pro Facette ein oder mehrere Attribute auswählen
- R8<sup>-</sup>** Je nach Datentyp der Facette variiert die Auswahlmöglichkeit (Textfeld zur Eingabe bei Jahreszahlen)
- R9<sup>+</sup>** Benutzer haben die Möglichkeit, mithilfe einer Suchfunktion Instanzen und Facetten zu finden
- R10<sup>+</sup>** Beim Anklicken einer Ressource können je nach Verfügbarkeit folgende Detail- und Medieninformationen gezeigt werden:
  - Metainformationen
  - Bilder
  - Videos
  - Webseiten
- R11<sup>+</sup>** Benutzer bekommen zur aktuellen Position umliegende Facetten und Instanzen angezeigt
- R12<sup>+</sup>** Falls die Ressource eine Orts-Information enthält, öffnet sich durch Anklicken dieser die Kartenansicht mit der jeweiligen Position
- R13<sup>-</sup>** Beim Anklicken einer Ressource wird die Entfernung in Kilometer (Meter) zur aktuellen Position angezeigt

## **Kartenansicht**

- R14<sup>+</sup>** Ressourcen mit Ortsinformationen können auf der Basis von Google-Maps eingesehen werden
- R14.1<sup>+</sup>** Die aktuelle Position des Benutzers wird auf der Karte markiert
- R14.2<sup>+</sup>** Die Kartenansicht bietet die Möglichkeit, Events in Abhängigkeit der eingestellten Zeit visuell unterschiedlich darzustellen
- R14.3<sup>+</sup>** Gefundene Ressourcen wie POIs werden visuell auf der Karte dargestellt
- R14.4<sup>+</sup>** POIs lassen sich Anklicken, so dass Detailinformationen eingesehen werden können
- R14.5<sup>-</sup>** Beim Herauszoomen werden überlappende Informationen visuell geclustert
- R14.6<sup>-</sup>** Die Kartenansicht bietet eine Navigationshilfe

## **Listenansicht**

- R15<sup>+</sup>** Ressourcen ohne abhängige Ortsinformationen werden in einer Listenansicht dargestellt
- R15.1<sup>+</sup>** Ein Listeneintrag zeigt den Titel der Ressource und je nach Verfügbarkeit kurze Metainformationen
- R15.2<sup>+</sup>** Gefundene Ressourcen werden absteigend sortiert (Z.B. Entfernungen der Ressourcen)

## **Timeslider**

- R16<sup>+</sup>** Das System bietet eine Zeitleiste um Ressourcen mit entsprechenden Zeitinformationen (z.B. Events) auszuwählen
- R16.1<sup>+</sup>** Ressourcen können über explizite oder intervallabhängige Angaben des Datums und der Zeit gefunden werden
- R16.2<sup>+</sup>** Die Kartenansicht wird je nach Angabe der Zeit und des Datums dynamisch angepasst
- R16.3<sup>-</sup>** Die Listenansicht wird je nach Angabe der Zeit und des Datums dynamisch angepasst

**R17**<sup>+</sup> Beim Verlassen der Applikation werden Anwendungseinstellungen gespeichert

**R18**<sup>-</sup> Beim Verlassen der Applikation wird der aktuelle Suchkontext gespeichert

**R19**<sup>-</sup> Buddies werden auf der Karte angezeigt

**R20**<sup>-</sup> Der Client bietet die Möglichkeit Benutzerprofile zu verwalten

**R21**<sup>-</sup> Der Client bietet die Möglichkeit Favoriten zu verwalten

### 6.1.2 Backend

**R22**<sup>+</sup> Das System besitzt einen Server, der für die Datenabfrage zuständig ist

**R22.1**<sup>+</sup> Der Server kann durch Angaben einer Geokoordinate Instanzen in der Umgebung des Anwenders finden

**R22.2**<sup>+</sup> Der Server kann durch Angabe eines Schlüsselwortes nach Instanzen suchen

**R22.3**<sup>+</sup> Ergebnisse werden in eine valide XML-Antwort geschrieben

**R22.4**<sup>-</sup> High-Level Facetten lassen sich in einem Konfigurationsfile festlegen

**R23**<sup>+</sup> Der Server dient als Kommunikationsschnittstelle zwischen dem Endgerät und den Datenrepositories.

**R23.1**<sup>+</sup> Der Server nimmt Anfragen von den Endgeräten entgegen, verteilt diese an entsprechende Datenrepositories und stellt die Ergebnisse für den Client bereit

**R23.2**<sup>-</sup> Der Server bietet die Möglichkeit der verteilten Anfrage von Datenrepositories

**R24**<sup>-</sup> Der Server kann bei den Repositories Anfragen, welche Facetten diese zu bieten haben

## 6.2 Nicht-funktionale Anforderungen

Bei den nicht-funktionalen Anforderungen liegt der Fokus auf der allgemeinen Usability der Anwendung. Dabei wird hauptsächlich die Android-basierte Client-Anwendung in Betracht gezogen. Für diese Arbeit ist es wichtig, dass neben den allgemeinen funktionalen Anforderungen, die letztendlich die Basis der Anwendungen bilden, auch die Usability der Benutzeroberfläche eine tragende Rolle spielt.

- R25<sup>+</sup>** Der Benutzer versteht die Client-Anwendung
- R26<sup>+</sup>** Der Benutzer fühlt sich bei der Bedienung der Anwendung nicht "verloren"
- R27<sup>+</sup>** Die Client-Anwendung lässt sich für jede Benutzergruppe (Experte oder Novize) einfach bedienen
- R28<sup>+</sup>** Die Inhalte lassen sich einfach, sicher und schnell finden
- R29<sup>+</sup>** Der Benutzer ist im Allgemeinen zufrieden mit der Client-Anwendung
- R30<sup>+</sup>** Lesbarkeit der Client-Anwendung: Der Kontrast zwischen Text und Hintergrund muss optimal gewählt werden
- R31<sup>+</sup>** Navigation der Client-Anwendung: Die Navigation soll erkennbar und nachvollziehbar sein (gering verschachtelter Ablauf)
- R32<sup>+</sup>** Minimierung unnötiger Aktionen: Schritte zur Bewältigung von Aufgaben müssen gering gehalten werden
- R33<sup>+</sup>** Konsistent Bezeichner: Gleiche Begriffe für die gleichen Sachverhalte
- R34<sup>+</sup>** Rückmeldungen: Bestätigungen und Hinweise bei Aktionen durch den Benutzer
- R35<sup>+</sup>** Steuerbarkeit: Der Anwender muss die Kontrolle der Client-Applikation behalten.
- R36<sup>+</sup>** Erwartungskonformität: Aktionen und Auswirkungen durch den Anwender müssen seine Erwartungen entsprechen
- R37<sup>-</sup>** Lernzeit: Der Anwender sollte sich schnell in die Client-Anwendung einfinden können

## 7 Architektur und Entwurf

Das MobileFacets System basiert auf einer Schichtenarchitektur und besteht aus drei Schichten (englisch: 'three tier architecture'). Die erste Schicht ist die Präsentationsschicht ('client tier'), die für die Präsentation der Daten sowie die Interaktion mit diesen zuständig ist. Die zweite Schicht ist die Logikschicht ('application-server tier'), sie stellt Mechanismen für die Verarbeitung der angefragten Daten und die Kommunikation mit den Datenrepositories bereit. Die dritte Schicht (data-server tier, back end) stellt die Datenbank oder Datenhaltungsschicht dar. Diese ist verantwortlich für die Speicherung der Daten, sowie die Bereitstellung von Schnittstellen für die Datenabfrage [42].

Die Clients steuern Anfragen an den Applikations-Server, wie beispielsweise eine Suchanfrage nach verfügbaren Facetten in der Umgebung. Konzeptuell wird im Client das Model-View-Controller<sup>25</sup> Prinzip implementiert, welches eine interaktive Steuerung mit den Daten ermöglicht. Gemäß den Anforderungen aus Kapitel 6 bedarf es für die Darstellung der Daten eine gewisse Diversifikation. Je nach Facettentyp (Orte, Personen, usw.) muss eine unterschiedliche Präsentation der Ergebnisse realisiert werden. Das Model-View-Controller Prinzip ist für diese Problematik somit ein geeignetes Architekturmuster .

Die Kommunikation zwischen Clients und Server, sowie dem Server und den Datenrepositories verläuft über TCP/IP und HTTP. Die Anbindung der Datenrepositories verläuft über eine HTTP RESTful Schnittstelle und SPARQL-Endpunkte, je nachdem welche Schnittstelle angeboten wird. Die Kommunikation zwischen Client und Server verläuft über eine HTTP-RESTful-Schnittstelle, in der für die Arbeit eine eigens definierte XML Datei konstruiert wurde.

Im Folgenden wird zuerst auf die Komponenten aus Abbildung 17 eingegangen, in dem detailliert die Bausteine erklärt werden. Im Anschluss werden die für diese Arbeit angebotenen Datenrepositories behandelt. Im Anschluss wird auf die Suche nach Facetten und Instanzen eingegangen, in der die konzeptuellen Vorgänge zur Suche nach Informationen beschrieben sind. Der letzte Abschnitt Kommunikation behandelt den Datenaustausch zwischen Client und Server.

---

<sup>25</sup>[http://www.phpwact.org/pattern/model\\_view\\_controller](http://www.phpwact.org/pattern/model_view_controller)

## 7.1 Komponenten

Die Komponenten des Systems werden in Abbildung 17 gezeigt. Im Folgenden werden drei Hauptkomponenten betrachtet: die Client(s)-Anwendung, der Server und die Linked Open Data. Dabei spiegelt die Abbildung das Design dieser Komponenten wieder und stellt eine abstrakte Sicht auf die Funktionalitäten sowie den Aufbau des MobileFacets System dar.

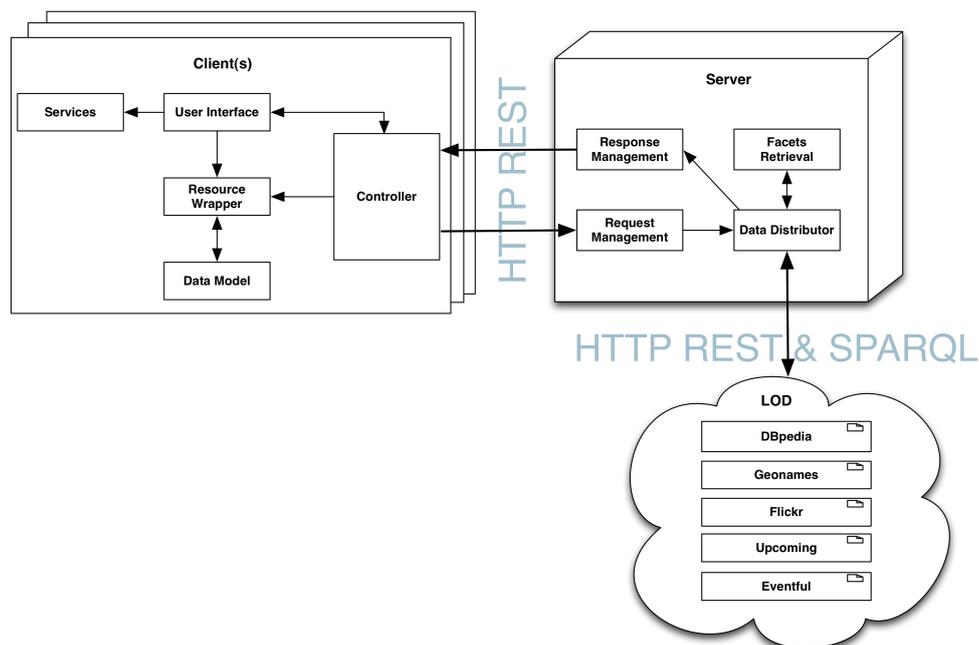


Abbildung 17: Architektur

### 7.1.1 Client

Die Interaktionen mit dem MobileFacets System finden ausschließlich auf einem mobilen Endgerät statt, auf dem die Client-Applikation installiert ist. Das System ist so konzipiert, dass gleichzeitig mehrere Clients Anfragen an den Server stellen können. Die Client-Komponente hat die Hauptaufgabe, Interaktionen mit Facetten, sowie das Suchen nach Informationen bereitzustellen. Die Abfragelogik wird größtenteils von einem lokalen Server erledigt, so dass die Prozesslast der Clients nicht überbeansprucht wird. Lediglich die Facettenklassifikation, die in der Komponente Data Model bereitgestellt wird, verarbeitet Daten. Die restlichen Komponenten sind ausschließlich in

der Funktionalität für die Ein- und Ausgabe beschränkt. Somit handelt es sich um einen abgewandelten Thin-Client.

Die Hauptaufgaben des Clients, Abbildung 17, linke Seite, sind:

- Der *Controller* ist die Steuereinheit des Clients. Hier werden Benutzerinteraktionen verarbeitet, wie etwa die Anfrage von neuen Daten. Anhand der aktuellen Position wird ein entsprechendes HTTP-Request an den Server gestellt. Bei positivem HTTP-Response werden die vom Server gelieferten XML-Daten analysiert und mit Hilfe des Resource Wrappers und dem unterliegenden Daten Models gespeichert. Somit stellt der Controller das Bindeglied zwischen den Clients und dem Server dar. Eine weitere Aufgabe ist die kontinuierliche Aktualisierung des User-Interfaces. Je nachdem welche Facetten der Benutzer auswählt, werden die entsprechenden Views (Kartenansicht, Listenansicht, Photoansicht) durch den Controller in Abhängigkeit der Facetten aktualisiert.
- Das *Data Model* repräsentiert einerseits die Datenstruktur einer Facette, andererseits werden Instanzen, in Abhängigkeit des Facettentyps (Ort, Person, Organisation Event usw.), im Resource Wrapper unterschiedlich gespeichert. Eine Facettenklassifikation sorgt dafür, dass zum Beispiel Instanzen mit einer Orts-Facette als Point of Interests im Resource Wrapper gespeichert werden. Instanzen mit der Facette *Person* oder *Organisation* werden nicht als Point of Interests deklariert und unterscheiden sich dementsprechend in der internen Repräsentation des Resource Wrappers.
- Das *User Interface* ist der für den Nutzer sichtbare Teil des Systems. Es stellt die Interaktion mit dem Facetten bereit. Mit Hilfe von grafischen Steuerelementen kann der Benutzer Facetten auswählen, Instanzen im Detail einsehen oder eine Suche starten. In Abhängigkeit der ausgewählten Facetten werden entsprechende "Views", wie etwa die Kartenansicht, die Listenansicht oder die Fotoansicht von der Controller Komponente aktualisiert. Mehr zur grafischen Benutzeroberfläche in Kapitel 4 und 8.
- *Services* stellen Zusatzdienste für den Client bereit. Diese Zusatzdienste umfassen die Ermittlung der aktuellen Position via GPS und die Live-Vorschau der freien Suche nach Instanzen.
- Die Komponente *Resource Wrapper* repräsentiert die interne Darstellung der Daten. Ermittelte Daten des Controllers werden hier in Abhängigkeit

des Daten Modells abgelegt und für das User-Interface bereitgestellt. Der Resource Wrapper ist somit eine Client-seitige, dynamisch generierte Datenbank.

### 7.1.2 Server

Der Server ist die zentrale und lokale Kommunikationseinheit zwischen Clients und der Linked Open Data. Er nimmt Anfragen von Clients entgegen und verarbeitet diese in entsprechende Anfragen für die eingebundenen Datentöpfe. Er besitzt keine permanente Datenbank, sondern dient nur als Vermittler und Bereitsteller von Informationen. Die komplette Abfrage-logik, wie SPARQL-Anfragen oder HTTP-RESTful Anfragen bei den Webservices, wird von dem Server geregelt. Somit kann der Datenverkehr zwischen Clients und dem Server minimiert werden, da nur die benötigten und ohne großen Overhead belasteten Daten ausgetauscht werden. Die Hauptaufgaben des Servers sind, gezeigt auf Abbildung 17, rechte Seite, wie folgt:

- Die Komponente *Request Management* nimmt Client-seitige HTTP-GET Anfragen entgegen und dient als Kommunikationsschnittstelle zwischen dem lokalen Server und den Clients. Die Hauptaufgabe liegt in der Verarbeitung der Anfrage und der Steuerung des Data Distributors. Die HTTP-GET Anfrage des Clients wird analysiert, und je nachdem welcher Abfragetyp vorliegt, z.B. Kartensuche oder freie Suche, an den Data Distributors weitergeleitet. Nicht gültige Client-Anfragen werden abgewiesen.
- Der *Data Distributor* dient als zentrale Datenbeschaffungskomponente des lokalen Servers. Er nimmt Anfragen des Request Managements entgegen und ist für die verteilte Anfrage der Daten zuständig. Je nachdem welche Datenrepositories eingebunden sind, werden in Abhängigkeit der HTTP-GET Parameter, Anfragen an die Linked Open Data Cloud geschickt. Es können dabei Daten von RESTful Webservices oder SPARQL Endpunkte angefragt werden. Angeforderte Ressourcen werden mittels der Komponenten Facets Retrieval analysiert. Falls eine Ergebnismenge vorhanden ist, werden die Informationen an das Response Management weitergeleitet.
- *Facets Retrieval* dient dazu, aus den angeforderten Daten des Data Distributors, Facetten zu extrahieren. Extrahierte Facetten werden im Anschluss an die entsprechende Ressource abgebildet.
- Das *Response Management* hat die Aufgabe, gefundene Ressourcen des Data Distributors in ein entsprechendes XML Format zu schreiben und

dem anfragenden Client mittels HTTP-REST Response zur Verfügung zu stellen.

### 7.1.3 Datenrepositories

In diesem Abschnitt geht es um die Datenrepositories, welche an das MobileFacets System angebunden sind. In Abbildung 17 illustriert die *|Linked Open Data* Wolke die "Datenbank". Die Kommunikation zwischen Server und den eingebundenen Repositories erfolgt bei DBpedia über einen SPARQL-Endpunkt. Geonames, Eventful, Upcoming und Flickr werden jeweils über ein HTTP RESTful Webservice angesprochen.

#### DBpedia

DBpedia<sup>26</sup> ist ein Gemeinschaftsprojekt, das sich zum Ziel gesetzt hat, strukturierte Informationen aus der Wikipedia zu extrahieren. Diese sollen für Web-Anwendungen, speziell für das semantische Web, zur Verfügung gestellt werden. Die Daten werden als Resource Description Framework Triple (RDF-Triple) gespeichert und können somit für semantische Anfragen genutzt werden. Es ist mithilfe der RDF-Triple möglich, semantische Zusammenhänge zwischen Objekten zu finden. Beispielsweise lassen sich anhand der Ressource *Koblenz* Informationen zu berühmten Persönlichkeiten der Stadt finden. Zudem können aus den Wikipedia-Kategorien Facetten gewonnen werden.

Im Folgenden wird eine kleine Übersicht über die extrahierten Daten aus der Wikipedia gezeigt.

- 3.4 Millionen "Dinge"
- 312.000 Personen, 413.000 Orte, 94.000 Musik-Alben, 49.000 Filme, 15.000 Videospiele, 140.000 Organisationen, 146.000 Spezies
- 841.000 Links zu Bildern, 5.081.000 Homepage-Links, 9.393.000 Links zu anderen RDF-Datensätzen
- 556.000 Wikipedia Kategorien, 75.000 YAGO Kategorien

Für diese Arbeit wird die englische Version der DBpedia Wissensbasis verwendet und umfasst 257 Millionen RDF-Triple. Die Anbindung an das MobileFacets System erfolgt über einen SPARQL-Endpunkt, welcher in Abschnitt 8.3.2 ab Seite 97 diskutiert wird. Weiterhin wird dort auf die Extraktion von Facetten eingegangen.

---

<sup>26</sup><http://dbpedia.org/About>

## Geonames

Geonames<sup>27</sup> ist eine umfangreiche Datenbank, die Informationen zu geographischen Objekten zur Verfügung stellt. Die Datenbank enthält Informationen zu 10 Millionen Entitäten, die 7.5 Millionen einzigartigen topografischen Objekten entsprechen. Die Daten lassen sich über einen Webservice oder über einen Datenexport beziehen [43].

Geonames spielt bei der Beschaffung der Daten des MobileFacets System eine wichtige Rolle, da dieser Service durch eine gezielte Kartensuche alle wichtigen und relevanten Objekte finden kann. Geonames bietet darüber hinaus eine konsequente Integration des Semantic Webs an, die beispielsweise für die Dereferenzierung von Wikipedia Ressourcen genutzt werden kann. Mehr zur Integration von Geonames und der Anbindung des Webservices lässt sich in Abschnitt 8.3.2, Integration der Datenrepositories nachlesen.

## Eventful

Eventful<sup>28</sup> ist ein Webservice, der es Benutzern ermöglicht, Events zu suchen, einzustellen und Informationen zu Events zu finden. Eventful ist ein Service der Firma Eventful, Inc. und wurde bekannt durch den Namen EVDB (Events and Venue Database). Der Service wird mittlerweile von 5 Millionen Personen genutzt und bekommt wöchentlich ungefähr 200.000 neue Nutzer dazu<sup>29</sup>. Neben den Informationen zu Events, umfasst die Datenbank umfangreiche Informationen zu Musikern, Komikern, Schauspielern und mehr.

Für diese Arbeit ist das umfangreiche Repertoire von dauerhaften 4 Millionen lokalen Events interessant. Die Suche nach Events wird realisiert durch einen Webservice, durch den umfangreiche Informationen, wie Austragungsorte, Kategorien und Zeitangaben abgerufen werden können.

## Upcoming

Upcoming<sup>30</sup> ist eine Community zum gemeinsamen Austausch von Events. Dabei können Events von registrierten privaten und kommerziellen Betreibern angelegt werden. Ein Event wird anhand einer Kategorie, eines Austragungsortes und eines definierten Zeitraums spezifiziert. Kategorien können Musik, Sport, Geschäftstreffen und so weiter sein.

Die Upcoming API bietet die Möglichkeit, anhand von angegeben Geoparametern, Events in der Umgebung zu finden. Die Kategorien die-

---

<sup>27</sup><http://www.geonames.org/>

<sup>28</sup><http://eventful.com/>

<sup>29</sup><http://about.eventful.com/>

<sup>30</sup><http://upcoming.yahoo.com/>

nen für diese Arbeit als Subfacetten, die unter der High-Level Facette *Event* angesiedelt sind.

## Flickr

Flickr<sup>31</sup> ist ein kommerzielles Portal, das es Benutzern erlaubt, digitale Bilder und Videos hochzuladen und somit anderen Nutzern zur Verfügung stellen. Bilder können mit Notizen, Tags und Kommentaren versehen werden. In jeder Minute werden circa 3000 Fotos hochgeladen und derzeit befinden sich 5 Milliarden Fotos auf den Servern von Flickr<sup>32</sup>.

Flickr bietet neben dem Austausch von Fotos einen umfangreichen Webservice, der es erlaubt, öffentlich zugängliche Fotos, für eigens entwickelte Anwendungen zu benutzen. Für diese Arbeit spielen die mit Geotags<sup>33</sup> versehenen Fotos eine Rolle: Nutzer können mit Hilfe von GPS-Empfänger die Position der Aufnahme an das Bild hängen. Somit kann eine Kartensuche nach verfügbaren Fotos vorgenommen werden. Mehr zur Integration der Fotos und einem kurzen Beispiel in Abschnitt 8.3.2 auf Seite 97.

## 7.2 Suchen von Facetten und Instanzen

In diesem Abschnitt geht es um das Suchen von Facetten und Instanzen, welches durch das MobileFacets System auf zwei unterschiedliche Ansätze vorgenommen werden kann:

1. Kartensuche mit der Funktion "What's around me?"
2. Freie Suche nach Instanzen

Die Kartensuche realisiert anhand der aktuellen Position und des gezeigten Kartenausschnittes eine typische ortsbezogene Suche. Zu beachten ist, dass das MobileFacets System auch unabhängig der aktuellen Position Informationen suchen kann. Der physische Standort des Anwenders muss nicht zwangsläufig dem Suchradius des Informationsspektrums entsprechen. Das heißt, dass der angezeigte Kartenausschnitt den Suchbereich definiert. Somit ist es durch Verschieben und Zoomen in der Karte möglich, den Suchbereich nach belieben zu verändern. Das ist insofern interessant, wenn man Informationen in entfernten Orten finden möchte. Im Folgenden wird davon ausgegangen, dass die aktuelle mit GPS-ermittelte Position als Ausgangspunkt genommen wird.

---

<sup>31</sup><http://www.flickr.com>

<sup>32</sup><http://blog.flickr.net/de/2010/09/19/5-000-000-000/>

<sup>33</sup><http://www.flickr.com/map/>

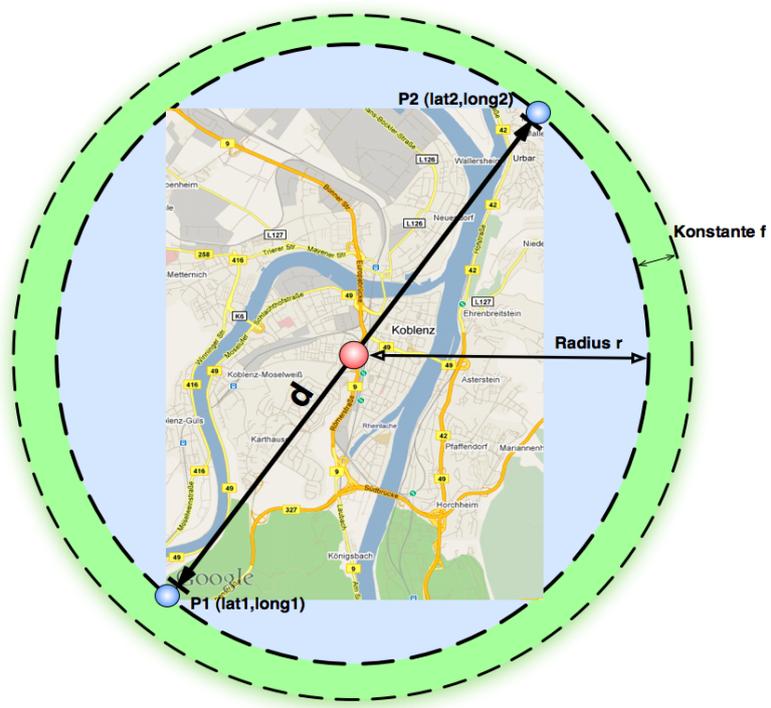


Abbildung 18: Haversine Berechnung

## Abstandsberechnung zwischen zwei Koordinaten

Das MobileFacets System bietet die Suche nach Informationen in Abhängigkeit des jeweiligen Ortskontextes. Dabei müssen folgenden Informationen berechnet werden:

1. Aktuelle Position anhand des GPS-Empfängers
2. Suchradius in Abhängigkeit der ermittelten Position

Die aktuelle Position wird kontinuierlich durch den Thread `LocationThread` berechnet. Voraussetzung ist, dass der Benutzer den Endgerätspezifischen GPS-Empfänger<sup>34</sup> eingeschaltet hat. Abbildung 18 zeigt einen Google-Maps Kartenausschnitt, in dem die aktuelle Position ermittelt wurde. Dabei wird die eigene Position auf der Karte zentriert dargestellt (hier durch den roten Punkt signalisiert). Falls kein GPS-Empfänger am Endgerät vorhanden ist, oder die Signalstärke der Satelliten nicht ausreicht, wird die Position anhand einer Mobilfunk-Triangulation vorgenommen. Jedoch liegt hier die Genauigkeit der Positionsbestimmung deutlich unter der Positionierung mittels GPS.

Bei der Berechnung des Suchradius  $r$  sind die Eckpunkte des angezeigten Kartenausschnittes die Bezugspunkte.  $P1$  repräsentiert die linke untere Ecke,  $P2$  die rechte obere Ecke des Kartenausschnitts. Die Punkte setzen sich aus den Attributen Latitude und Longitude zusammen. Äquivalente Punkte im Code sind die Funktionspaare:

P1: (`getMapBottom`, `getMapLeft`) und  
P2: (`getMapTop`, `getMapRight`).

Anhand dieser zwei Punkte lässt sich mithilfe der Haversine Gleichung  $d$  berechnen.

### Haversine Gleichung

Die Haversine Gleichung ist eine Berechnungsmethode um den Luftliniendistanz zwischen zwei Koordinaten zu ermitteln. Man bezeichnet die kürzeste Verbindung zwischen zwei Punkten auf der Erdoberfläche (Kugeloberfläche) als Orthodrome oder umgangssprachlich als Luftlinie [44]. Dabei setzen sich die Koordinaten aus den Längen- und Breitengraden zusammen.

Die statische Methode

---

<sup>34</sup>Bei modernen Smartphones ist dieser direkt im Gerät integriert

```
public static double calculateDistance(double lat1, double long1,
double lat2, double long2)
```

berechnet folgenden Abstand  $d$ :

Berechnung der geographischen Länge und Breite:

$$R = 6370 \quad (1)$$

$$d2r = \frac{\pi}{180} \quad (2)$$

$$dlon = (lon2 - lon1) * d2r \quad (3)$$

$$dlat = (lat2 - lat1) * d2r \quad (4)$$

Berechnung der Kurswinkel:

$$a = \left(\sin \frac{dlat}{2}\right)^2 + \cos(lat1 * d2r) * \cos(lat2 * d2r) * \left(\sin \frac{dlon}{2}\right)^2 \quad (5)$$

$$c = 2 * \arctan 2(\sqrt{a}, \sqrt{1-a}) \quad (6)$$

Berechnung der Strecke P1P2:

$$d = R * c \quad (7)$$

- (1) Länge des Äquators =  $2 * \pi * R$ , dabei ist  $R \approx 6370$  km der Radius der Erde.
- (2) Die Koordinaten der Anwendung werden intern als Gradmaß behandelt. Für die Harversine-Distanz Berechnung ist eine Umrechnung der Variablen  $dlon$  und  $dlat$  mit dem Faktor  $d2r$  notwendig, der die Parameter in ein äquivalentes Bogenmaß umrechnet.
- (3) Berechnung der Delta-Differenz der Breitengrade beider Orte und Multiplikation mit Faktor  $d2r$
- (4) Berechnung der Delta-Differenz der Längengrade beider Orte und Multiplikation mit Faktor  $d2r$
- (5) Hier wird der Kurswinkel der Strecke berechnet
- (6) Berechnung der Strecke zwischen zwei Punkten im Einheitskreis mit Hilfe des Arkustangens und zwei Parameter  $\sqrt{a}$  und  $\sqrt{1-a}$

- (7) Um den Abstand zwischen zwei Punkten auf der Erdkugel zu bekommen, wird  $c$  noch mit dem Erdradius  $R$  multipliziert

In Abbildung 18 repräsentiert  $d$  die durch die Harvesine Gleichung berechnete Distanz zwischen P1 und P2. Der Suchradius um den Mittelpunkt errechnet sich durch die Halbierung der Distanz  $d$ . Da die Karte rechteckig ist, erstreckt sich der Suchradius  $r$  infolgedessen zusätzlich auf einen nicht-sichtbaren Teil der Karte (blaue Schattierung). So werden auch Informationen gefunden, die nicht gerade in diesem sichtbaren Kartenausschnitt vorhanden sind. Zudem spielt der Zoomfaktor der Karte eine zentrale Rolle: Falls der Anwender die Karte mit einem großen Zoomfaktor bedient, werden Informationen in einem sehr kleinen Suchradius gefunden. Aber vielleicht ist es von Interesse, dass auch entferntere Informationen von Relevanz sind. Stellt man sich das initiale Szenario aus Kapitel 2.3, Point of Interest-Suche vor, so kann es sein, dass gerade weiter entfernte Informationen für den Anwender relevant und somit interessant sind. Um dies zu realisieren, wird der Suchradius  $r$  um eine Konstante  $f$  erweitert. In Abbildung 18 auf Seite 65 ist dieser erweiterte Radius grün schattiert.

### Parameter der Kartensuche

Die ermittelte Distanz  $d$  und GPS-Positionierung mit den Attributen (long,lat) sind zwei Parameter, die für die Anfrage beim Server notwendig sind. Weiterhin lässt sich der Suchkontext modifizieren. Das heißt, der Benutzer kann auswählen, nach welchen Informationen gesucht werden soll. In den Anwendungseinstellungen, Abschnitt 8.2.6, lassen sich bestimmte Datentöpfe, wie beispielsweise Wikipedia, Events oder Flickr an- und abwählen. Im Folgenden wird eine HTTP-REST Anfrage gezeigt, die bei der Funktion "What's around me?" generiert wird:

```
http://csxpoi.isweb.unikoblenz.de:8081/MobileFacetsServer/_  
RestFacetProvider?long=7.59147&lat=50.348618&geonames-  
search-mode=findNearbyWikipedia&m=2&distance_  
=13.818659481892826&search-dbpedia=true&search-events=  
false
```

Listing 2: What's around me? REST Anfrage

Die Parameter `lat` und `long` repräsentieren die aktuelle Position des Nutzers. Parameter `distance` ist die errechnete Distanz, beziehungsweise der Radius in dem gesucht wird. Die Distanz wird hier in Kilometer angegeben. Zusätzlich werden in dieser Beispielanfrage nur nach Wikipedia-Artikeln gesucht, da `search-events` auf `false` gesetzt wurde. Die Verarbeitung dieser Anfrage wird ausführlich in Abschnitt 8.3.1 ab Seite 92 diskutiert.

## 7.3 Kommunikation

Die Kommunikation zwischen einem Client und der Server Komponente wird durch den HTTP RESTful Architekturstil realisiert, welche in Abbildung 17 auf Seite 59 schematisch dargestellt wird. Die Kommunikation zwischen Server-Komponente und der Linked Open Data erfolgt wahlweise über das HTTP RESTful Kommunikationsprotokoll oder über eine HTTP SPARQL Anfrage. In diesem Abschnitt wird auf die Kommunikation zwischen einem MobileFacets Client und der Server Komponente eingegangen. Die Kommunikation zwischen Server und den Datenrepositories wird in Abschnitt 8.3.2 auf Seite 97 ausführlich erläutert.

In einem REST-Architekturstil repräsentieren URLs Ressourcen auf einem Webserver, wobei die HTTP Methoden `GET`, `POST`, `PUT` und `DELETE` mögliche Operationen auf diesen Ressourcen ermöglichen. Das MobileFacets System zielt darauf ab, Informationen aus verschiedensten Datenquellen abzufragen. Die allgemeine Funktionalität beschränkt sich im Hinblick auf die RESTful Architektur nur auf das reine Abfragen von Informationen. Somit werden die HTTP Methoden `POST`, `PUT` und `DELETE` nicht verwendet. Lediglich die Methode `GET` ist für diesen Anwendungskontext von Bedeutung und wird im Folgenden erläutert.

Eine Serveranfrage wird initiiert, indem die Basis-URL `http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Server/RestFacetProvider?` mit den in Tabelle 2 vorhandenen Parameter und Werte gefüllt wird. Die Tabelle zeigt alle möglichen Parameter mit ihren Definitionsbereichen, die vom MobileFacets Server akzeptiert und weiterverarbeitet werden kann. Der Platzhalter `<CSXPOI>` repräsentiert die Basis-URL.

`GET <CSXPOI>/center-long | center-lat` Die Werte der Parameter `center-long`, `center-lat` werden durch die Kartensuche (Abschnitt 7.2, Seite 64) errechneten Werte zugewiesen. Sie repräsentieren hierbei die Kartenmitte und werden in Längen- beziehungsweise Breitengraden angegeben.

`GET <CSXPOI>/north | south | east | west` Diese Parameter repräsentieren die Kartengrenzen. Also der sichtbare Teil der Karte mit den Eckpunkten Süden, Norden, Westen und Osten. Geonames bietet neben der Umkreissuche die Möglichkeit anhand von definierten Eckpunkten, eine Bereichssuche an. Diese Parameter werden in Abhängigkeit des Parameters `geonames-search-mode` verwendet und sind somit optional.

Parameter	HTTP Methode	Definitionsbereich
<CSXPOI>/center-long	GET	-180 ... 180
<CSXPOI>/center-lat	GET	-90 ... 90
<CSXPOI>/north	GET	-90 ... 90
<CSXPOI>/south	GET	-90 ... 90
<CSXPOI>/east	GET	-180 ... 180
<CSXPOI>/west	GET	-180 ... 180
<CSXPOI>/search-mode	GET	1   2
<CSXPOI>/geonames-search-mode	GET	bounding-box   searchNearby
<CSXPOI>/keyword	GET	String
<CSXPOI>/distance	GET	0 ... 10.000
<CSXPOI>/search-dbpedia	GET	TRUE   FALSE
<CSXPOI>/earch-events	GET	TRUE   FALSE

Tabelle 2: REST Parameter und Definitionsbereiche

GET <CSXPOI>/search-mode Mit den Werten 1 und 2 repräsentieren sie den Suchmodi, also ob eine reine Kartensuche oder eine Schlüsselwortsuche vorliegt.

GET <CSXPOI>/geonames-search-mode Wie oben schon kurz erwähnt, kann mit diesen Werten der Geonames-Suchmodus verändert werden. In dieser Arbeit verwendet der MobileFacets Client in der Voreinstellung den Geonames "searchNearby" Suchmodus.

GET <CSXPOI>/keyword Falls der Benutzer eine freie Suche nach Instanzen startet, wird dieser Parameter gebraucht, der das Schlüsselwort repräsentiert. Der Wert kann aus einem beliebigen Text-String bestehen.

GET <CSXPOI>/distance Die Distanz wird immer zusammen mit den Parameter center-long und center-lat angegeben. Der Wert wird anhand in der Haversine-Berechnung aus Abschnitt 7.2 ermittelt und gibt den Suchradius für die Kartensuche an.

GET <CSXPOI>/search-dbpedia Der Benutzer hat die Möglichkeit, durch manuelles Abschalten einzelner Datentöpfe den Suchraum einzuschränken. Mithilfe der Anwendungseinstellungen, auf die in Abschnitt 8.2.6 auf Seite

91 eingegangen wird, kann die Suche nach Wikipedia Artikeln ein- und ausgeschaltet werden.

`GET <CSXPOI>/search-events` Identisch zur vorherigen URI, können auch Events in Abhängigkeit der Anwendungseinstellungen gesucht werden.

Zu beachten ist, dass diese Parameter nicht willkürlich und in beliebigen Kombinationen vom Server als RESTful Abfrage akzeptiert werden. Das System bietet die Suche nach Facetten und Instanzen in zwei Modi an. Die Kartensuche und die freie Suche nach Instanzen und Facetten. Bei der Kartensuche müssen folgende Parameter vorliegen:

- `search-mode`
- `center-long`
- `center-lat`
- `distance`
- `geonames-search-mode`
- `search-dbpedi`
- `search-events`

Bei einer freien Suche nach Instanzen folgende:

- `search-mode`
- `keyword`
- `search-dbpedi`
- `search-events`

Eine dieser beiden Suchmodi eines Clients löst einen Suchvorgang beim Server aus, der bei Erfolg die in Listing 3 oder 4 gezeigte Antwort liefert. Die Antwort wird durch die Server-Komponente `Request Management` geregelt, in dem ein XML-Dokument geschrieben wird. Spezifiziert wird das XML-Dokument durch den MIME Typ `application/xml; charset='utf-8'` [45, 46].

```

<?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE resource ␣
  SYSTEM "http://csxpoi.isweb.uni-koblenz.de/resource.dtd">
<res type="DBpedia" count="92"
  db_name="University␣of␣Koblenz␣and␣Landau"
  db_url="http://dbpedia.org/resource/␣
    University_of_Koblenz_and_Landau"
  db_facets="[Koblenz,␣Universities_in_Germany,␣␣
    Universities_and_colleges_in_Rhineland-Palatinate␣
    ,␣Educational_institutions_established_in_1990,␣␣
    Organisation,␣EducationalInstitution,␣University]␣
  "
  db_longitude="8.108611111111111"
  db_latitude="49.204444444444445"
  db_homepage="http://www.uni-koblenz-landau.de/"
  db_abstract="The␣University␣of␣Koblenz-Landau␣(German␣
    ␣Universität␣Koblenz␣-␣Landau)␣is␣a␣university␣␣
    located␣in␣Koblenz␣and␣Landau,␣Germany.␣It␣was␣
    founded␣in␣1990."
  db_thumbnail="http://upload.wikimedia.org/wikipedia/␣
    commons/thumb/9/91/Universitätsbibliothek_Koblenz␣
    -Landau.jpg/200px-Universitätsbibliothek_Koblenz-␣
    Landau.jpg"
  db_photo="http://www4.wiwiss.fu-berlin.de/␣
    flickrwrappr/photos/␣
    University_of_Koblenz_and_Landau"
  db_isPOI="true"
  db_resourceType="Place"/>

```

Listing 3: MobileFacets Serverantwort (DBpedia XML)

Das erste Beispiel zeigt eine XML-Antwort, in der eine DBpedia-Ressource aufgelistet ist. Bei Listing 4 handelt es sich um ein Event, welches in der Umgebung gefunden wurde. Erwähnenswert sind die Attribute *db\_facets* und *ev\_category*, in der die Facetten aus der Quell-Ressource stehen. Falls es sich um ein POI handelt (signalisiert durch das Attribut *is\_poi*) stehen weiterhin die Längen- und Breitengrade in der XML-Antwort, sodass diese genutzt werden können, diese Ressource auf der Karte anzeigen zu lassen.

```

<?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE resource ␣
  SYSTEM "http://csxpoi.isweb.uni-koblenz.de/resource.dtd">
<res type="Events" count="21"
  ev_id="E0-001-032657346-3"
  ev_name="The␣Big␣Chris␣Barber␣Band"
  ev_description="Chris␣Barber␣und␣seine␣Band␣zählten␣
    bereits␣in␣den␣50er␣Jahren␣zu␣den␣bekanntesten␣
    und␣beliebtesten␣Dixielandgruppen␣Europas␣mit␣New␣
    ␣Orleanscher␣Jazzprägung.␣Und␣noch␣heute␣bieten␣

```

```
    sie_Jazz_und_Swing_vom_Allerfeinsten..._"  
ev_start_date="2010-10-26_20:00:00"  
ev_end_date="2010-10-26_23:59:00"  
ev_latitude="50.430956"  
ev_longitude="7.45858"  
ev_venue_address="Luisenstraße"  
ev_venue_city="Neuwied"  
ev_venue_name="Stadthalle_Heimathaus"  
ev_category="[Music]"  
ev_thumbnail="http://static.eventful.com/images/_  
    medium/I0-001/002/072/543-4.jpeg"  
ev_resourceType="Event"/>
```

Listing 4: MobileFacets Serverantwort (Event XML)

## 8 MobileFacets Prototyp

Die nachfolgenden Abschnitte beschreiben das MobileFacets System in all seinen Komponenten. Zuerst wird die Vorgehensweise der schrittweisen Implementierung des Clients betrachtet. Im Anschluss wird detailliert auf diesen eingegangen. Der nächste Abschnitt behandelt die Server-Komponente, in der es um die Integration der Datenrepositories geht und wie Client-Anfragen vom Server verarbeitet werden.

### 8.1 Iterative Implementierung und Benutzerfeedback

Während der Implementation wurde zum einen drauf wert gelegt, das System in kleine Teilfunktionalitäten zu zerlegen, die nacheinander hinzugefügt wurden. Zum anderen war eine flexible Anpassung hinsichtlich des User-Interfaces wichtig. Die ersten Implementationen des User-Interfaces haben als Grundlage die aus der ersten Designstudie extrahierten Erkenntnisse (nachzulesen in Abschnitt 9.4 aus dem Kapitel Designstudie). Basierend auf diesen Entscheidungen wurden die ersten Entwürfe implementiert, um diese anschließend durch kurze Tests mit den Anwendern zu validieren und gegebenenfalls aussagekräftige Kritiken für die nachfolgenden Implementationsarbeiten zu bekommen.

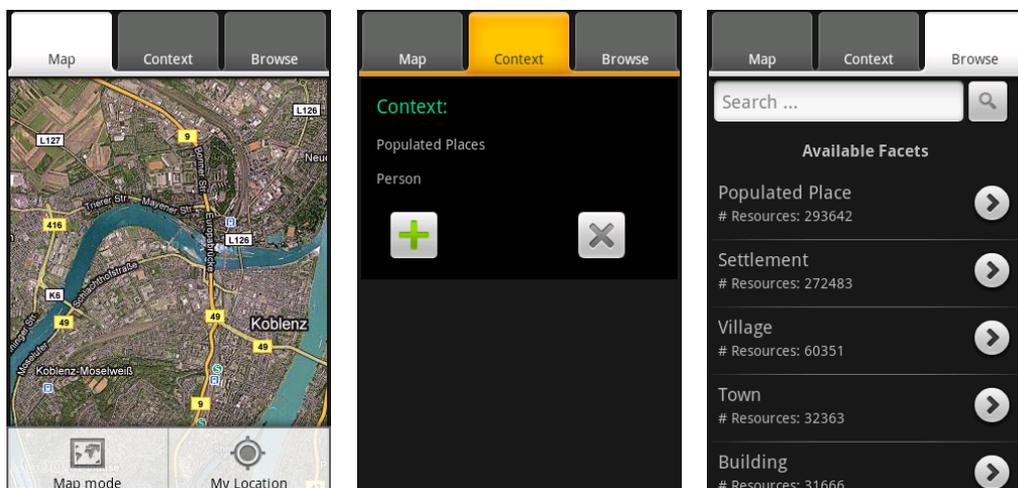


Abbildung 19: Erster lauffähiger Oberflächenprototyp (Client Screenshots)

Diese Vorgehensweise ist an den iterativen Entwicklungsprozess [47] angelehnt und wird unter der agilen Softwareentwicklung eingeordnet. Diese Voraussetzungen ermöglichen flexible Änderungen der Anforderungen an das

System. Im Folgenden wird kurz auf die einzelnen Entwicklungsphasen eingegangen. Es werden beispielhaft Screenshots der Anwendung im Laufe der Implementierung gezeigt.

Bei den folgenden Abbildungen handelt es sich um prototypische Screenshots, die mithilfe des Android-Emulators aufgenommen wurden. Mehr zum Thema Android und dessen Aufbau lässt sich in Abschnitt 8.2 nachlesen. Abbildung 19 zeigt den ersten Entwurf mit einer integrierten Kartenansicht. Am oberen Bildschirmrand befinden sich drei Tabs: **Map**, **Context** und **Browse**. Der Tab **Context** zeigt zwei ausgewählte Facetten in Form eines Labels. Zudem ist hier eine Funktionalität eingebaut, um neue Facetten hinzuzufügen oder zu löschen. Signalisiert wird dies durch die zwei unteren Knöpfe. Der dritte Tab **Browse** ist die Browse-Ansicht, in dem der Anwender die Möglichkeit hat, Facetten auszuwählen. Realisiert wird dies durch eine einfache Liste, in dem ein Listeneintrag aus Facettenname und Anzahl der Ressourcen besteht. Im oberen Bereich befindet sich eine integrierte Suchfunktion, mit der die Suche nach Instanzen implementiert wird. Gemäß den Ergebnissen aus der Designstudie werden Facetten durch das Anklicken eines Listeneintrages ausgewählt. Zudem dient die Liste als Navigation durch Subfacetten und Instanzen.

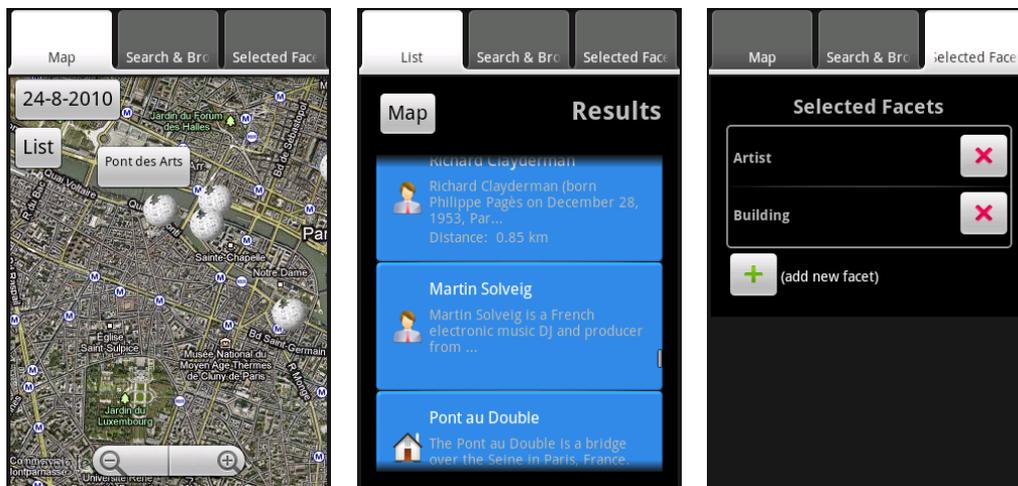


Abbildung 20: Zweiter lauffähiger Oberflächenprototyp (Client Screenshots)

Im zweiten Entwurf (Abbildung 20) wurden einige Funktionalitäten hinzugefügt: Es erfolgte eine Integration einer Zeitleiste<sup>35</sup>, die es ermöglicht, Events in Abhängigkeit des eingestellten Datum visuell unterschiedlich darzustellen. Eine weitere Funktion der Kartenansicht ist der Austausch des

<sup>35</sup>Mehr zur Zeitleiste in Abschnitt 8.2.2 auf Seite 83

Tab **Map** zum Tab **List**. Realisiert wurde dies in dem linken Screenshot, durch den Knopf *List* unterhalb der Datumsanzeige. Somit konnte ohne einen zusätzlichen Tabeintrag die Listenansicht, die zur Anzeige der ausgewählten Facetten ohne spezifischen Koordinatenbezug, integriert werden. Der erste Tab fungiert somit als Ergebnisansicht, der sowohl POIs auf der Karte, als auch alle Facetten in der Liste dargestellt. Dieser Designschritt wurde gewählt, um zum einen keinen zusätzlichen vierten Tab einzubauen. Zum anderen soll der erste Tab (Map und Liste) als spezielle Ergebnisansicht der ausgewählten Facetten dienen.

Der Tab **Selected Facets** entstand aus dem vorherigen Tab **Context** und wurde im Detail angepasst. Es ist nun möglich selektierte Facetten einzeln abzuwählen. Das Anklicken des *add new facet* Knopfes hat zu Folge, dass die Ansicht in den Search & Browse Reiter wechselt, in dem der Anwender neue Facetten mithilfe der Liste (identisch zu Abbildung 19, Screenshot in der Mitte) hinzufügen kann.

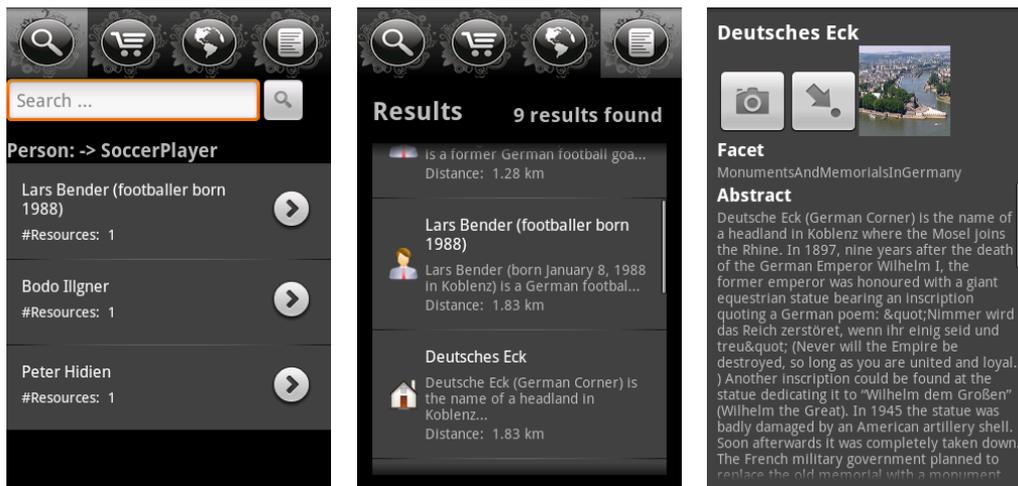


Abbildung 21: Dritter lauffähiger Oberflächenprototyp (Client Screenshots)

Aus den Kommentaren des Benutzerfeedbacks hat sich ergeben, dass die Integration der Tabs **Map** und **List** auf lediglich einen Reiter keine sinnvolle Maßnahme ist. Es verwirrte, dass ein einziger Tab beide Funktionalitäten beinhaltet. Somit wurde ein weiterer Tab eingeführt, welcher in Abbildung 21 begutachtet werden kann. Weiterhin wurde die blaue Farbe der Listenansicht bemängelt, dass zu einer insgesamt Inkonsistenz in der Farbgebung führte. Somit wurde eine einheitlichere Farbgebung gewählt. Zusätzlich wurde das Tab-Design neu gestaltet. Statt eines Namens zur Tab-Identifizierung wurde auf Icons gesetzt, die leichte Assoziation zur jeweiligen Funktionalität bieten.

Weiterhin gab es Modifikationen hinsichtlich der Tab-Reihenfolge, sodass der erste Tab nun als Einstiegspunkt in die Anwendung fungiert. Der Benutzer kann direkt sehen, welche Facetten er auswählen möchte. Dies war in dem vorherigen Oberflächenprototyp nicht der Fall; der Benutzer sah als Erstes die Karte. Tester kritisierten, dass die Kartenansicht keine Interaktion erlaubt, solange keine Facetten ausgewählt sind. Dieser Umstand wurde somit behoben. Der Browse & Search Reiter (Screenshot links, Abbildung 21) hat zusätzlich eine Funktion für die Navigationsunterstützung bekommen. Oberhalb der Liste wird ein Navigationspfad in Abhängigkeit der ausgewählten Facetten angezeigt.

Für die nächste Iteration, welche zum fertigen Prototypen führte, gab es noch folgende Kritikpunkte:

- Ein Einkaufswagen-Icon für den **Selected Facets** Reiter ist nicht intuitiv.
- Das Suchfeld innerhalb des Browse & Search Reiter verwirrte: Man nahm an, dieser sei für die Suche innerhalb der Facetten zuständig. Jedoch wird hier eine neue Suche nach Instanzen initiiert.
- Tester waren verwirrt, als sie Facetten ausgewählt hatten und diese aber erst durch ein manuelles Wechseln in den **Selected Facets** sehen konnten.
- Die Kartenansicht aus Abbildung 20 zeigt POIs aus der DBpedia als kleine Wikipedia-Icons. Man wünschte sich eine klarere Differenzierung hinsichtlich des Facettentyps.

Aus diesen Kritikpunkten wurden einige Modifikationen am Prototypen vorgenommen. Im nächsten Abschnitt wird auf den MobileFacets Prototyp eingegangen, welcher auch Grundlage für die Evaluation aus Kapitel 9 ist.

## 8.2 Client

Die Client Anwendung ist ein für das Android Betriebssystem entwickelter Prototyp, welcher die Interaktionen und Visualisierungen mit dem MobileFacets System bereitstellt. Das Android SDK umfasst eine umfangreiche API<sup>36</sup>, die eine Client-seitige Anwendungsentwicklung ermöglicht. Das Android Betriebssystem basiert auf einem Linux Kernel, der grundlegende Services wie Multitasking, Sicherheitsfunktionen, Hardware-Treiber und Netzwerkfunktionen beinhaltet. Oberhalb des Linux-Kernels

---

<sup>36</sup><http://developer.android.com/reference/packages.html>

befinden sich zahlreiche C++ Bibliotheken, wie eine SQLite Datenbank oder eine Media-Framework für grafische Anwendungen. Anwendungen laufen in der sogenannten Dalvik Laufzeitumgebung, welche eine für mobile Endgeräte entwickelte Java virtuelle Maschine ist. Die oberste Schicht ist das Anwendungsframework<sup>37</sup>, in der umfangreiche Komponenten für die Entwicklung bereitgestellt werden.

Android Applikationen bestehen in erster Linie aus Activities, Intents und Views. Eine Activity repräsentiert ein User-Interface, in dem der Fokus auf die Interaktion des Benutzers mit der Anwendung gelegt wird. Eine Activity ist eine Subklasse der `android.app.Activity` und kann aus mehreren graphischen Komponenten bestehen. Das User-Interface jeder Activity wird von einer View (Subklasse von `android.app.View`) repräsentiert. Ein View wird in XML definiert und kann hierbei aus Knöpfen, Listen, Textlabels usw. bestehen. Dabei wird jede Activity generell in einem eigenen Fenster dargestellt, wobei jede Activity-Instanz einen eigenen Lebenszyklus besitzt. Übergänge zwischen Activities werden über Intents (`android.content.Intent`) realisiert, wobei zusätzliche Übergabeparameter und ähnliches angegeben werden können.

Das MobileFacets System beinhaltet die Main-Activity `MobileFacetsActivity`, die beim Starten der Client-Anwendung aufgerufen wird. Sie stellt das grundlegende Layout der Anwendung bereit. Gemäß den Ergebnissen aus der Design-Studie aus Kapitel 5 besteht die Anwendung aus einem Tab-Layout. Die `MobileFacetsActivity` erweitert eine `TabActivity` und implementiert die Klasse `TabContentFactory`, welche die benötigten Funktionen zur Darstellung von Tabs bereitstellt. Der Bildschirm gliedert sich in vier Tabs, in der folgenden Activities in der Reihenfolge von links nach rechts angeordnet sind: `BrowseFacetsActivity`, `InteractiveMapActivity`, `ResultsListActivity` und `ImageGalleryActivity`. Erwähnenswerte Klasse sind zudem die `DetailViewActivity` und die `SearchActivity`, welche beide unabhängig des jeweiligen Tabs aufgerufen werden können.

Im Folgenden wird detaillierter auf die Anwendung eingegangen, wobei hier das Hauptaugenmerk auf den Klassen des User-Interface-Pakets `de.unikoblenz.west.mobileFacets.ui` gelegt wird. Zuerst werden Kartensuche und freie Suche nach Instanzen geschildert. Danach wird auf die einzelnen Tabs der Anwendung detailliert eingegangen. Im Anschluss geht es um die Tab-unabhängigen Activities, wie die `DetailViewActivity` und die `PreferenceActivity`.

---

<sup>37</sup><http://developer.android.com/guide/basics/what-is-android.html>

## 8.2.1 Suchen, Browsen und Interaktion mit Facetten und Instanzen

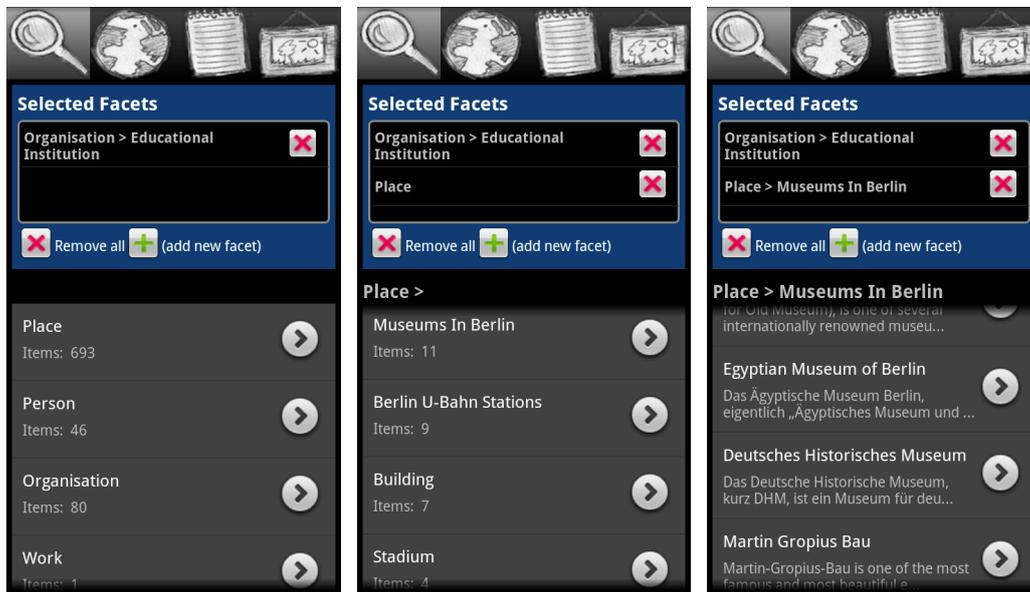


Abbildung 22: High-Level Facetten, Subfacetten und Instanz-Ebene (Client Screenshots)

Die `BrowseFacetsActivity` ist zentraler Bestandteil der `MobileFacets` Anwendung und ermöglicht umfangreiche Interaktionen mit Facetten. Der `Browse & Search` Tab befindet sich an erster Stelle (von links betrachtet). Zu den Hauptfunktionen zählen:

1. Kartensuche
2. Instanz-Suche
3. Selektion und Browsen von Facetten
4. Löschen von Facetten
5. Vorschau von Facetten-Instanzen

Die Kartensuche wird beim Starten der Anwendung automatisch durchgeführt und ermöglicht somit dem Anwender einen direkten Einblick in die verfügbaren, umliegenden Facetten, ohne dass eine zusätzliche Funktion aufgerufen werden muss. Falls der Benutzer seinen physischen Standort gewechselt hat oder je nach angezeigtem Kartenausschnitt (Mehr zur Suche im Kartenausschnitt in Abschnitt 7.2 auf Seite 64) die Umkreissuche neu starten möchte, befindet sich die *What's around me?* Funktionalität im virtuellen

Menü<sup>38</sup>. Nach Anwendungsstart und erfolgreicher Suche wechselt die Ansicht automatisch in die Browse & Search Ansicht. Illustriert in Abbildung 22, in der gefundene Facetten gezeigt werden.

Die Bildschirmaufteilung der Browse & Search Ansicht ist zweigeteilt: Im oberen Teil befindet sich ein Feld, in dem selektierte Facetten aufgelistet werden (*Selected Facets*); der untere Teil zeigt eine Liste, die verfügbare Facetten auflistet. In der linken Abbildung wurde die Facette *Organisation - > Educational Institution* ausgewählt. Das bedeutet, dass die Kartenansicht und die Ergebnisliste, welche in den nachfolgenden Abschnitten detailliert diskutiert werden, Ergebnisse aus der Facette *Educational Institution* beinhalten.

Die Selektion von Facetten wird folgendermaßen durchgeführt: Der Anwender bekommt nach der Suche von Facetten (z.B. durch die Umkreissuche: *What's around me?*) eine Liste mit verfügbaren High-Level Facetten<sup>39</sup>. Durch Anklicken eines Listeneintrages wird die Liste mit den jeweils zugehörigen Subfacetten gefüllt. Abbildung 22, mittleres Bild, zeigt die Liste, in dem die Facette *Place* selektiert und alle zugehörigen Subfacetten aufgelistet sind. Das rechte Bild zeigt die Instanz-Ebene, in dem die Facette *Museums in Berlin* selektiert wurde. Der Indikator *Place -> Museums in Berlin* oberhalb der Liste ist dafür da, um den Navigationspfad innerhalb der Liste zu zeigen. Gemäß den Anforderungen aus Abschnitt 4 für die facettierte Navigation [1, 32], kann somit eine nachvollziehbare Navigationsstruktur ermöglicht werden.

Durch das Anklicken einer Facette wird diese automatisch zur Ergebnismenge hinzugefügt. Das heißt, das System aktualisiert auf Anhieb die Ergebnismenge, die in der Karten- sowie Listenansicht dargestellt werden. Es müssen keine zusätzlichen Mechanismen betätigt werden, um Facetten zu selektieren. Falls der Benutzer beispielsweise nur eine Vorschau der Subfacetten oder den Instanzen wünscht, kann er mit Hilfe des Hardwareknopfes *Back*<sup>40</sup> zurück zur nächsthöheren Ebene gelangen. Das Betätigen des *Back* Knopfes in rechten Screenshot aus Abbildung 22 hätte zur Folge, dass die Liste alle Subfacetten zu *Place* anzeigen würde (mittlerer Screenshot).

Weiterhin zeigt die rechte Abbildung, dass die Facetten *Educational Institution* und *Museums in Berlin* selektiert wurden. Es ist nun möglich, vorher selektierte Facetten wieder zu löschen. Dazu dienen die neben einer selektierten Facette liegenden Knöpfe. Falls der Anwender es wünscht, alle

---

<sup>38</sup>Die *What's around me?* Funktion befindet sich in jedem Tab und kann dort über das virtuelle Menü aufgerufen werden

<sup>39</sup>Definition von High-Level Facetten in Abschnitt 3.2 und deren Datenintegration in 8.3.2

<sup>40</sup>Auf allen gängigen Android-basierten Geräten vorhanden

Facetten zu löschen und somit die Ergebnismenge auf den initialen Zustand zu bringen, dient ein *Remove all* Knopf, welcher unterhalb der selektierten Facetten zu finden ist. Das Löschen einer Facette muss durch den Anwender in einem Dialog bestätigt werden.

Die Funktion *add new facet* ist für das Selektieren von mehr als einer Facette zuständig. Durch Anklicken dieses Knopfes wird die High-Level Facetten Ebene aus dem linken Screenshot aus Abbildung 22 gezeigt und ein Anwender kann eine neue Facette hinzufügen.

Die Instanz-Ebene zeigt alle zugehörigen Ressourcen der aktuell ausgewählten Facette. Dies wird im rechten Screenshot der Abbildung 22 illustriert. Es werden in der Liste alle verfügbaren Museen in der Umgebung von Berlin angezeigt. Durch Anklicken eines Eintrages öffnet sich die Detailansicht, welche in Abschnitt 8.2.5 diskutiert wird. Die Instanz-Ebene in dem Browse & Search Reiter ermöglicht somit, dass der Anwender eine Vorschau der Ergebnisse sehen kann, ohne dass ein zusätzlicher Wechsel in die Ergebnisliste stattfinden muss.

Im folgenden Paragraphen wird auf die Freie Suche nach Instanzen eingegangen.

## Freie Suche nach Instanzen

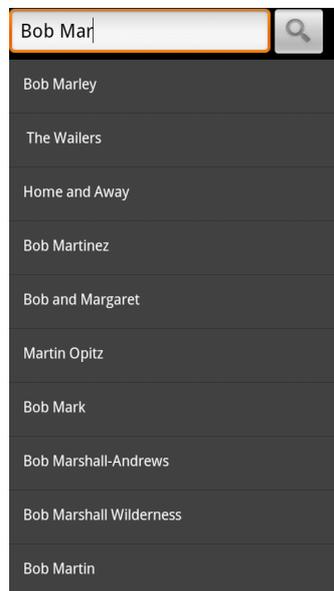


Abbildung 23: Freie Textsuche (Client Screenshots)

Die freie Suche nach Instanzen kann unabhängig des aktuellen Ortskontextes verwendet werden. Das heißt, die Suche beschränkt sich nicht mittelbar auf umliegende Informationen, sondern es können beliebige Anfragen abgeschickt werden. Die Suche lässt sich entweder über das virtuelle Menü aus dem Search & Browse Reiter (siehe: 8.2.1), oder über einen vorhandenen Suchknopf des Endgerätes<sup>41</sup> starten. Abbildung 23 zeigt die `SearchActivity`, in der Suchanfragen an den Server gestellt werden können. Die `SearchActivity` ist in zwei Bildschirmbereiche geteilt. Im oberen Bereich befindet sich ein Suchfeld, in dem der Benutzer mithilfe der virtuellen-, oder falls vorhanden, der Hardwaretastatur Texteingaben tätigen kann. Im unteren Bereich befindet sich eine Liste, in der Vorschläge zum Suchbegriff angezeigt werden. Die sogenannte Live-Vorschau verwendet ein *DBpedia Lookup Webservice*<sup>42</sup>, der anhand eines Schlüsselwortes die am ehesten passenden DBpedia URIs liefert. Der Webservices bekommt als Anfrage das Schlüsselwort, hier im Beispiel den Text-String "Bob Mar", und als Antwort liefert dieser eine absteigend sortierte Liste von relevanten DBpedia URIs. Der Algorithmus des *Lookup Service* sucht nach gezielten Übereinstimmungen im Label einer Ressource. Neben dem Label, wird zudem eine Suche anhand der vorhandenen Synonyme des unterliegenden Lucene Indexes vollbracht. Im Beispiel von "Bob Marley" erkennt man, dass auch seine Band "The Wailers" in der Liste an zweiter Stelle auftaucht. Das ist insofern interessant, da sich mithilfe dieses Webservice zusätzliche semantische Zusammenhänge aufdecken lassen. Für den Anwender hat das den Vorteil, falls er einen bestimmten Informationsaspekt in Gedanken hat, aber diesen möglicherweise nicht korrekt schreiben kann, dieser in der Liste auftauchen könnte. Weiterhin fungiert diese Live-Vorschau als Autovervollständigung und beschleunigt die Suche.

Durch Anklicken eines Listenelementes wird die Suche unmittelbar mit dem jeweiligen Such-String abgeschickt. Der Suchbereich kann, wie auch bei der "What's around me? Funktion, durch Einschränkungen in den Anwendungseinstellungen modifiziert werden. Jedoch sind die Ergebnisse der Live-Vorschau lediglich auf Wikipedia-Artikel beschränkt. Vorschläge zu Events oder ähnlichem werden hier nicht gezeigt.

Die Kartensuche, sowie die freie Suche nach Instanzen kann durch den Benutzer abgebrochen werden. Dies wird realisiert durch den *Back-Hardwareknopf* eines Endgerätes.

---

<sup>41</sup>Dieser ist bei vielen Android-Geräten vorhanden

<sup>42</sup><http://lookup.dbpedia.org/>

## 8.2.2 Kartenansicht, Menü und Interaktion mit POIs und Events

Die Kartenansicht wird repräsentiert durch die View `InteractiveMapView` und ist zentraler Bestandteil der `InteractiveMapActivity`. Realisiert wird die Karte durch die Integration einer Google-Maps API. Durch die frei verfügbare API<sup>43</sup>, lassen sich benötigte Kartenfunktionen einbinden. Die `InteractiveMapView` ist eines der drei Views zur Darstellung von selektierten Facetten und befindet sich in der `MobileFacetsActivity` im zweiten Tab.

Die Karte lässt sich mithilfe der virtuellen Knöpfe (+) und Minus (-) (dargestellt in Abbildung 24) zoomen. Durch das Verschieben der Karte mit dem Finger lässt sich der aktuelle Kartenausschnitt modifizieren. Auf der Karte werden die Informationen angezeigt, die eine Geokoordinate besitzen. Dazu zählen zumeist Facetten aus dem Bereich *Places* und *Organisation*. Der linke Screenshot aus Abbildung 24 zeigt einen Kartenausschnitt mit vorhandenen POIs. Je nach Facettentyp, beispielsweise Museen, Schulen oder U-Bahn-Stationen, werden diese entsprechend ihrer Bedeutung, durch unterschiedliche Icons dargestellt. Die Icons stammen aus einer frei verfügbaren Bibliothek, bestehend aus über 900 Icons, welche eigens für die Google-Maps API entwickelt wurde<sup>44</sup>. Die Zuordnung der Icons zu den verschiedenen Facettentypen wird von der Klasse `FacetClassification` realisiert. Dort wird ein Mapping zwischen gefundenen Facetten und den vordefinierten (Facetten -> Icon) Paaren durchgeführt. Die Zuordnung der Facetten zu den Icons ist ein Bestandteil des unterliegenden Datenmodells, welches in Abschnitt 7.1, Architekturkomponenten, erläutert wird.

### Interaktion mit POIs und Events

Durch Anklicken eines POI öffnet sich eine Detailansicht, in der sich weitere Informationen nachlesen lassen. Somit lassen sich neben der Listenansicht und der Instanz-Ebene des Browse & Search Reiters auch hier Details zu Ressourcen ansehen. Die Detailansicht wird in Abschnitt 8.2.5 weiter erläutert.

Im mittleren und rechten Screenshot der Abbildung 24 werden so genannte Event-basiert POIs [48] gezeigt, die durch kleine bläuliche Sterne markiert sind. Ein Event hebt sich von einem gewöhnlichem POI in der Hinsicht ab, dass zusätzliche Zeitinformationen integriert sind. Ein Event wird definiert aus dem Veranstaltungsort, sowie einem Start- und Endzeitpunkt. In der Abbildung unterscheiden sich Events auf der Karte durch eine dif-

---

<sup>43</sup><http://code.google.com/android/add-ons/google-apis/maps-overview.html>

<sup>44</sup><http://code.google.com/p/google-maps-icons>

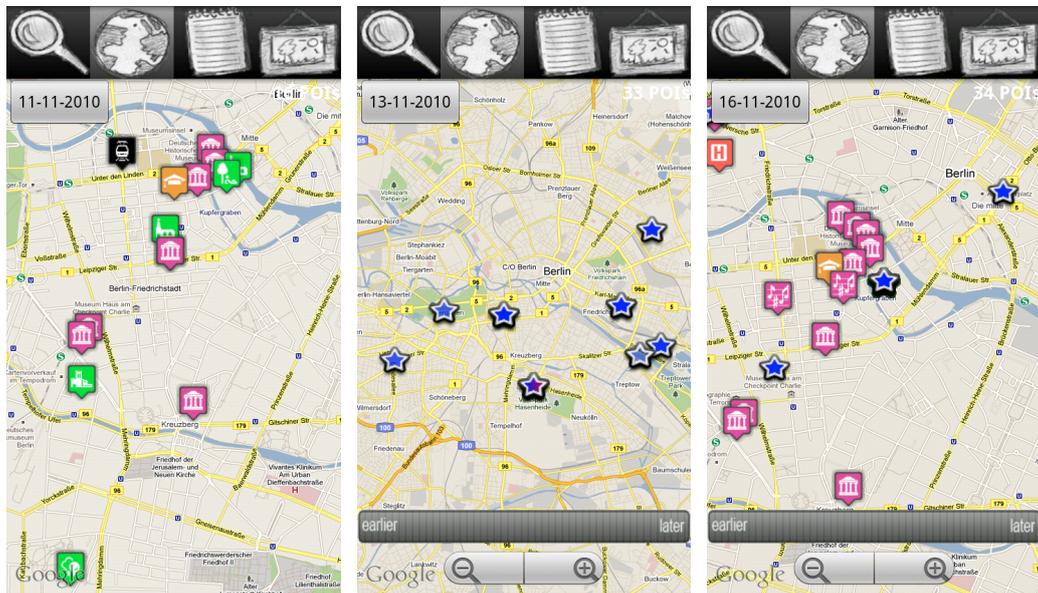


Abbildung 24: Karte mit POIs, Karte mit Events, Karte mit POIs und Events (Client Screenshots)

ferenzierte Darstellung. Sie werden bläulich markiert und je nach zeitlicher Distanz zum Startdatum, werden sie opak bis transparent dargestellt. Das Visualisierungssystem und die Zeitleiste (im unteren Bildschirmteil zu sehen) basiert auf der Arbeit von Daniel Schmeiss, Erweiterung der csxPOI-Anwendung um Event-basierte Points of Interests<sup>45</sup> [48].

Die Zeitleiste wird realisiert durch die Klasse `DateSeekBar` und implementiert das `SeekBar` Widget des Android SDK. Weiterhin ist sie ein zentraler Bestandteil der `InteracviteMapView`. Der Benutzer kann durch Flippen der Zeitleiste das Datum, je nach linker oder rechter Bewegungsrichtung, einstellen. In der oberen linken Ecke der Kartenansicht befindet sich ein `Android:Button`, der in Abhängigkeit der Zeitleiste automatisch das Datum darstellt. Durch Anklicken des Buttons öffnet sich die `DatePickerActivity`, in der sich das Datum manuell einstellen lässt. Das ist insofern hilfreich, falls der Benutzer ein exaktes Datum schneller einstellen möchte. Wie oben schon erwähnt, werden Events in unterschiedlicher Transparenz dargestellt. Je transparenter ein Event erscheint, desto weiter liegt das Startdatum dieses in der Zukunft. Es werden nur Events angezeigt, die in den nächsten 30 Tagen stattfinden. Events, die weiter in der Zukunft liegen, sind somit nicht sichtbar. Weiterhin werden Events, die gerade zum eingestellten Datum stattfinden,

<sup>45</sup>Dort ist die Komponente Zeitleiste und Darstellung von Events entstanden.

durch eine opaken, rötlich markierten Stern gekennzeichnet (in Abbildung 24, mittlerer Screenshot). Die Abstufung der Events-Symbole nach der Zeit wird in zehn unterschiedlichen Skalierungen dargestellt.

Welche Event-Ressourcen im MobileFacets System integriert sind, wird in Abschnitt 8.3.2 diskutiert.

## Menü

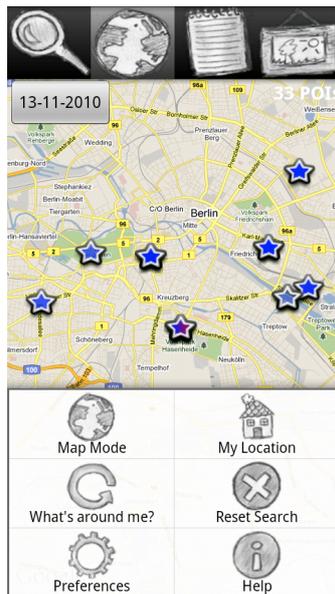


Abbildung 25: Menü der Karte (Client Screenshot)

Das virtuelle Menü der Karte lässt sich über den Hardware-Knopf des Gerätes aktivieren. Abbildung 25 illustriert das Menü der Kartenansicht. Das Menü lässt sich zusätzlich in der `BrowseFacetsActivity`, der `ResultListActivity` und der `ImageGalleryActivity` aktivieren, jedoch sind dort die Kartenspezifischen Funktionen nicht verfügbar. Lediglich die Funktionen *What's around me?*, *Reset Search* und *Preferences* sind identisch. Die Menüfunktionen sind wie folgt:

- *Map Mode*: Hier kann der Benutzer den Kartenmodus auswählen. Eine Satellitenansicht oder einer Kartenansicht. Letzteres wird in der Abbildung 25 gezeigt.
- *My Location*: Diese Menüfunktion zentriert den Kartenausschnitt auf die ermittelte GPS-Position des Benutzers und stellt den Punkt visuell

dar.

- *What's around me?*: Durch Aktivieren dieser Funktion wird die Kartensuche gestartet, die in Abschnitt 7.2 ab Seite 64 diskutiert wird.
- *Reset Search*: Hier kann der Benutzer seine Suchergebnisse löschen und die Anwendung auf den initialen Zustand zurück versetzen.
- *Preferences*: Öffnet die `PreferencesActivity`, die in Abschnitt 8.2.6 erklärt wird. Hier lassen sich Anwendungseinstellungen vornehmen.
- *Help*: Öffnet einen Hinweisdialog, in dem eine kurze Erklärung zur Symbolik von POIs und Events gezeigt wird.

### 8.2.3 Ergebnisliste



Abbildung 26: Liste mit Personen, Liste mit Orten (Client Screenshots)

Die Ergebnisliste ist der dritte Tabeintrag der `MobileFacetsActivity` und wird von der `ResultListActivity` implementiert. Sie zeigt eine Liste, in der aktuell ausgewählte Facetten angezeigt werden. Alle selektierten Facetten aus dem Search & Browse Reiter (Abschnitt 8.2.1) definieren die anzuzeigenden Facetten. Linker Screenshot in Abbildung 26 zeigt die Ergebnisliste, in der die Facetten *Artist* und *Athlete* selektiert sind. Der Benutzer sieht zudem, wie hoch die Anzahl der Ergebnisse ist (Hier dreizehn Listeneinträge).

Die Ergebnisliste ist nach aufsteigender Entfernung zu umliegenden POIs sortiert - vorausgesetzt der Benutzer hat Facetten selektiert, die aus dem Bereich *Places* und *Events* sind. Anschaulich wird dies durch den rechten Screenshot, in dem die Facetten *Archaeology Museums* und *Urban Public Parks* selektiert sind. Anderweitige Facetten, wie *Person*, *Organisation* und *Work* haben keine Auswirkung auf die Sortierung der Liste. Hier werden die Ergebnisse willkürlich aufgelistet.

Die Interaktionsmöglichkeiten der `ResultListActivity` beschränken sich auf das Scrollen durch die Liste und das Anklicken eines Listenelementes. Durch das Anklicken einer Ressource öffnet sich die Detailansicht, die in Abschnitt 8.2.5 diskutiert wird. Im Gegensatz zur Kartensicht, in der nur Facetten mit einer Koordinate (POIs und Events) angezeigt werden können, wird die Ergebnisliste definiert durch alle ausgewählten Facetten. Somit kann die Ergebnisliste Orte, Personen, Events und andere High-Level Facetten mit ihren Instanzen beinhalten. Die Ergebnisliste wird wie folgt errechnet:

High-Level Facetten:

$$hl_1, \dots, hl_n \quad (8)$$

Subfacetten:

$$hl_1(sub_{11}, \dots, sub_{1k}), \dots, hl_n(sub_{n1}, \dots, sub_{nk}) \quad (9)$$

definieren die Ergebnismenge:

$$hl_1(sub_{11} \cup \dots \cup sub_{1k}) \cap \dots \cap hl_n(sub_{n1} \cup \dots \cup sub_{nk}) \quad (10)$$

(8) Selektierte High-Level Facetten 1 ... n (siehe Abschnitt 3.2 auf Seite 11)

(9) Selektierte Subfacetten 1 ... k

(10) Schnittmenge der High-Level Facetten und Vereinigungsmenge der zugehörigen Subfacetten

Durch Schnittmengen der High-Level Facetten ergeben sich Konstellationen, in der die Ergebnisliste leer sein kann. Man stelle sich das Szenario vor, in dem der Benutzer die Facetten

- *Person*  $\Rightarrow$  Soccer Manager
- *Place*  $\Rightarrow$  Towers in Germany

selektiert hat. Falls die Facette *Person*  $\Rightarrow$  *Soccer Manager* vereinigt mit der Facette *Place*  $\Rightarrow$  *Towers in Germany* keine Schnittmenge bildet, bleibt die Ergebnisliste leer. Nur wenn ein semantischer Zusammenhang zwischen *Towers in Germany* und *Soccer Manager* existiert, ergeben sich Einträge in der Liste. Mehr zu den Taxonomien von Facetten lassen sich in Abschnitt 3.2 nachlesen.

### 8.2.4 Fotoansicht



Abbildung 27: Fotoansicht (Client Screenshots)

Die Fotoansicht ist der vierte und letzte Tab der `MobileFacetsActivity` und hat die Aufgabe, umliegende oder gesuchte Fotos anzeigen zu lassen. Der linke Screenshot aus Abbildung 27 zeigt zugehörige Bilder, die bei einer Kartensuche gefunden wurden. Im Gegensatz dazu werden bei dem rechten Screenshot Bilder die aus der freien Textsuche angezeigt.

Der obere Bildschirmbereich zeigt eine Vorschau-Gallery und wird realisiert durch das Widget "Gallery"<sup>46</sup>. Der Benutzer kann durch manuelles Flippen die einzelnen Bilder ansehen und das aktuell-angeklickte Foto wird im unteren Bildschirmbereich vergrößert dargestellt. Unterhalb des vergrößerten Bildes wird der jeweilige Titel angezeigt. Die Bilder stammen von Flickr<sup>47</sup> und werden entweder mithilfe der Kartensuche *What's around me?*

<sup>46</sup><http://developer.android.com/reference/android/widget/Gallery.html>

<sup>47</sup><http://www.flickr.com/>

oder der freien Suche nach Instanzen gefunden. Die Suche nach Fotos erfolgt nicht über den lokalen Server, sondern wird von der `ImageGalleryActivity` erledigt. Diese bekommt als Parameter, die aus Abschnitt 7.2 errechneten Werte (Radius und Koordinaten des Benutzers), übermittelt.

Die Anbindung an Flickr verläuft über die Flickr API<sup>48</sup> und ermöglicht anhand einer Suchfunktion Fotos im Umkreis zu finden. Weiterhin lässt sich, im Gegensatz zur Kartensuche, nach Schlüsselwörtern suchen. Die Integration von Flickr wird ausführlich in Abschnitt 8.3.2 auf Seite 97 diskutiert.

### 8.2.5 Detailansicht

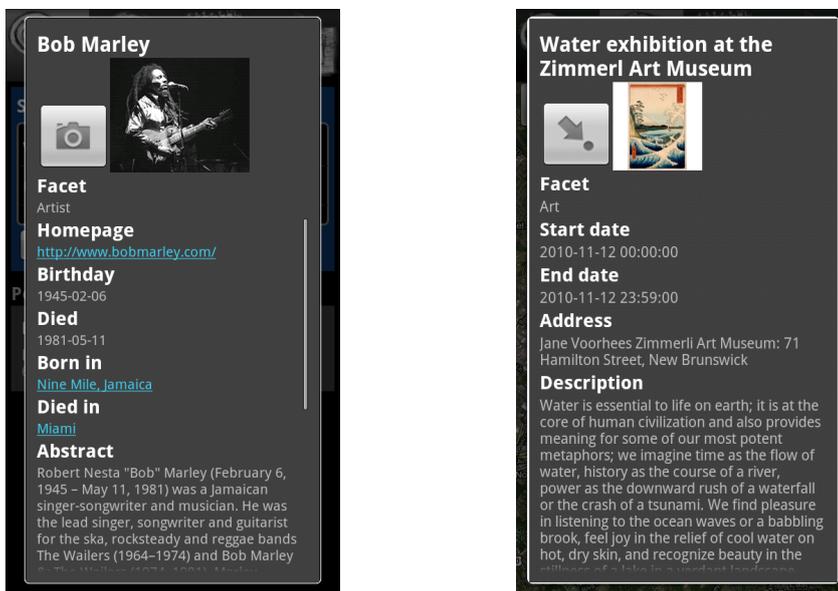


Abbildung 28: Personen Details, Event Details (Client Screenshots)

Die Detailansicht einer Ressource kann im Browse & Search Reiter, der Kartenansicht und in der Listenansicht aufgerufen werden und ist somit unabhängig des jeweiligen Kontextes. Durch Anklicken einer Ressource öffnet sich ein neues Fenster, in dem ausführliche Informationen eingesehen werden können. Je nach High-Level Facettentyp unterscheiden sich die angezeigten Elemente. Im Folgenden wird kurz auf die einzelnen Informationsaspekte eingegangen:

#### DBpedia Ressourcen:

<sup>48</sup><http://www.flickr.com/services/api>

- *Title*: Hier wird das Label der Ressource angezeigt.
- *Thumbnail*. Falls die Ressource das Attribut `owl:thumbnail` enthält, wird hier ein kleines Vorschaubild angezeigt.
- *Facet*: Hier wird der zugehörige Facettentyp angezeigt. Der Facettentyp wird durch die Facettenanalyse der Komponente *Data Model* (Siehe Kapitel 7.1 auf Seite 59) definiert.
- *Homepage*: Enthält die DBpedia-Ressource das Attribut `foaf:homepage`, erscheint ein entsprechender Link. Durch Anklicken des Links öffnet sich die `WebViewActivity`, die einen einfachen Browser implementiert.
- Die Felder *Birthday* und *Died* erscheinen typischerweise bei Personen, vorausgesetzt die Attribute `owl:birthDate` und `owl:deathDate` sind vorhanden.
- *Born in* und *Died in* zeigen bei Personen den Geburts- beziehungsweise Sterbeort. Die vorhandene Referenzierung von `owl:birthPlace` und `owl:deathPlace` kann verwendet werden, um eine neue Suche nach Informationen in den jeweiligen Orten zu starten. Das heißt, der Nutzer kann durch Anklicken eines Ortes, diesen als neues Suchsubjekt definieren. In Abbildung 8.2.6 würde ein Klick auf Miami bedeuten, dass eine Suche nach Miami (Personen, Orte, Events usw. ) gestartet wird. Der Nutzer wird mittels eines Dialogs darauf hingewiesen, ob er die Suche starten möchte.
- *Abstract*: Bei DBpedia-Einträgen wird hier der englische Abstract eines Artikels gezeigt.

#### **Event Ressourcen:**

- *Facet*: wie auch bei DBpedia-Ressourcen wird hier die jeweilige Facette angezeigt. Der Facettentyp wird durch die Facettenanalyse der Komponente *Data Model* (Siehe Kapitel 7.1 auf Seite 59) definiert.
- *Start Date*: Signalisiert das Startdatum eines Events. Dabei setzt sich dieses aus dem Datum und der Uhrzeit zusammen.
- *End Date*: Signalisiert das Enddatum eines Events. Auch hier setzt sich dieses aus dem Datum und der Uhrzeit zusammen.
- *Address*: Die Adresse setzt sich aus dem Austragungsort, der Straße und des Ortes zusammen.

- *Description*: Hier lassen sich Informationen zu einem Event einsehen. Typischerweise wird hier eine ausführliche Beschreibung geboten.

Neben den aufgelisteten Informationsaspekten, bietet die `DetailView-Activity` die Möglichkeit, nach Fotos zu suchen. Abbildung 8.2.6 enthält unter dem Label einer Ressource einen Foto-Knopf. Hier lassen sich mit Hilfe des `FlickrWrappers` relevante Flickr-Bilder einsehen. Dabei wird die `Image-GalleryActivity` mit der Ressourcen-URI als Übergabeparameter aufgerufen. Weitere Informationen zum Thema Integration von Daten lassen sich in Kapitel 8.3.2 ab Seite 97 nachlesen.

Neben der Foto-Suche können bestimmte Ressourcen mit Geokoordinaten verknüpft sein. Bei DBpedia-Ressourcen sind dafür die Attribute `geo:lat` und `geo:lat` von Relevanz. In dem rechten Screenshot der Abbildung 28 lässt sich mithilfe des `LocationKnopfes` unterhalb des Labels, der zugehörige Ort dieses Events anzeigen. Dabei öffnet sich die Kartenansicht und der Austragungsort wird automatisch in der Kartenmitte zentriert.

## 8.2.6 Anwendungs-Einstellungen

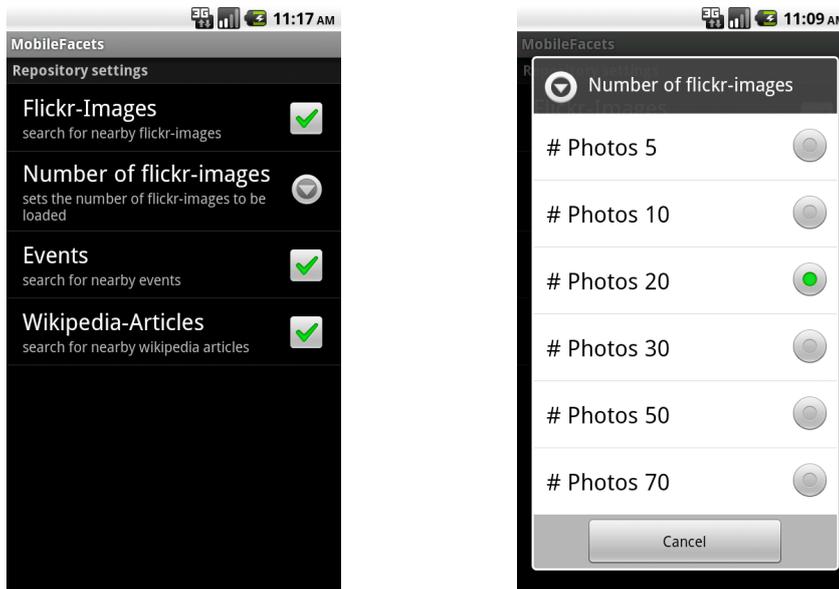


Abbildung 29: Repository Einstellungen (Client Screenshots)

Über das virtuelle Menü lässt sich die `PreferenceActivity` starten. Hier werden vorhandene Datentöpfe an- und abwählen. Falls der Benutzer wünscht, nur nach Events zu suchen, lässt sich hier die Suche nach Wikipedia-Artikeln, sowie eine Suche nach umliegenden Flickr-Fotos abwählen. Weiterhin

kann die Suche nach Flickr-Fotos auf eine bestimmte Anzahl eingeschränkt werden. Die Anzahl kann in einem Intervall von 10 bis maximal 70 Fotos eingestellt werden. Die Kartensuche und freie Suche nach Instanzen (Abschnitt 7.2 auf Seite 64) werden in Abhängigkeit der eingestellten Datentöpfe ausgeführt. Die zugehörigen Parameter sind *search-dbpedia* und *search-events*, die mit den Argumenten *true* oder *false* gesetzt werden. Der rechte Screenshot aus Abbildung 29 illustriert die Anwendungseinstellungen, in der aktuell alle Datentöpfe aktiviert sind. Der Rechte Screenshot zeigt die Auswahlmöglichkeit von Flickr-Fotos.

### 8.3 Server

Die Server Komponente des MobileFacets Systems bildet die zentrale Kommunikationseinheit zwischen den Clients und den Datenrepositories. Der Server läuft auf einer Apache Tomcat<sup>49</sup> Instanz und ist unter der URL <http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Server/> erreichbar. Die Funktionen des Servers werden über ein Java Servlet unter der Adresse *MobileFacet-Server/* aufgerufen. Das Management dieser Funktionen wird durch die Response- und Request Komponente aus Abbildung 17 im Kapitel Architektur realisiert.

Die Anbindung erfolgt bei Eventful, Upcoming, Flickr und Geonames über einen HTTP RESTful Webservice. Die Anfragen bei DBpedia werden an einen öffentlich zugänglichen SPARQL-Endpunkt<sup>50</sup> gesendet. Die Integration der Daten wird in Abschnitt 8.3.2 diskutiert, während im folgenden Abschnitt 8.3.1 auf die Verarbeitung von Client-Anfragen eingegangen wird. Die Kommunikation zwischen Client und dem Server erfolgt über das HTTP REST Protokoll und lässt sich in Kapitel 7, Abschnitt Kommunikation, nachlesen.

#### 8.3.1 Prozesskette

In diesem Abschnitt wird der funktionale Ablauf einer Client-Anfrage ausführlich diskutiert. Initiiert wird eine Suchanfrage durch ein HTTP-Request, welches Informationen zu der aktuellen Position, des Suchmodus und des Suchradius enthält. Abbildung 30 zeigt schematisch den Prozessablauf von einer Anfrage bis hin zur Antwort, welche in XML verfasst wird. Es handelt sich hierbei um ein UML-Sequenzdiagramm, in der die beteiligten Klassen mit ihren Funktionen und Funktionsaufrufen beschrieben werden. Die gelb-

---

<sup>49</sup><http://tomcat.apache.org/>

<sup>50</sup><http://dbpedia.org/sparql>

lich markierten Zahlen dienen zur Übersicht und werden als Anhaltspunkte für die folgende Auflistung der einzelnen Schritte definiert.

1. Bei einem Client-Request stellt der Server in Schritt **1** fest, welcher Suchmodus gewählt wurde. Dazu dient der Parameter `m`, der die Werte *1* und *2* annehmen kann. In dieser Abbildung wurde aufgrund eines besseren Verständnisses äquivalente Werte *vicinity* und *keyword* genommen. Falls es sich um eine Kartensuche handelt, verläuft der Prozess in der oberen Hälfte des Sequenzdiagrammes ab.
2. Hier wird der Request auf die Parameter `searchDBpedia` und `searchEvents` überprüft. Beide Parameter können jeweils die Werte *true* oder *false* annehmen. Die Funktionalität liegt darin, dass der Nutzer die Möglichkeit hat, Repositories an- und abzuwählen.
3. Ab diesem Schritt verläuft der eigentliche Suchprozess nach Informationen. Die Klasse `ServletHelper` ermittelt aus dem gegebenen Request die Parameter `location` und `distance`<sup>51</sup>. Die Funktion `findNearby` der Klasse `Geonames` ermittelt anhand einer Anfrage an den Geonames Webservice geografische Objekte. Dazu zählen Städte, Sehenswürdigkeiten, berühmte Gebäude, Kirchen, Museen und so weiter. In der Klasse `Geonames` wird anhand der gefundenen Objekte eine Liste erstellt. Der Geonames Webservice liefert pro Objekt eine referenzierbare DBpedia-Ressource, die für die nachfolgenden Schritte von Relevanz sind. Die Ressourcen sehen folgendermaßen aus: `http://dbpedia.org/resource/?`, wobei `?` durch das entsprechende Objekt ersetzt ist. Da es sich bei der Geonames-Umkreissuche um eine Suche nach Point of Interests handelt, werden diese Ressourcen im Vorfeld in die High-Level Facette *Place* zugeordnet.
4. Die Liste mit gefundenen Ressourcen wird in der Klasse `DBpedia` weiter verarbeitet. Diese Klasse hat die Aufgabe, aus vorhandenen Point of Interests weitere Ressourcen aus unterschiedlichen Facettenbereichen zu finden. Dafür kommt folgende SPARQL-Anfrage zum Einsatz:

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
SELECT ?personBirth ?personDeath ?personHome ?campus
WHERE{
    {?personBirth dbpedia:birthPlace <http://dbpedia.
        .org/resource/Koblenz> }
    UNION{?personDeath dbpedia:deathPlace <http://
        dbpedia.org/resource/Koblenz> }
```

---

<sup>51</sup>Die Berechnung dieser Werte wird in Abschnitt 7.2 diskutiert.



Abbildung 30: Server Sequenzdiagramm

```

UNION{?personHome dbpedia:hometown <http://
dbpedia.org/resource/Koblenz> }
UNION{?campus dbpedia:campus <http://dbpedia.
org/resource/Koblenz> } .
}

```

Listing 5: DBpedia Personen/Organisation-Suche (SPARQL-Abfrage)

Diese SPARQL- Abfrage zeigt beispielhaft die Suche nach Personen und Organisationen aus Koblenz. Der Aufbau und die Dereferenzierungsmöglichkeiten einer DBpedia Ressource wird in Abschnitt 8.3.2 ab Seite 97 weiter diskutiert.

5. In diesem Arbeitsschritt werden zu den gefundenen DBpedia Ressourcen weitere Informationen gesucht. Dazu zählen Detailinformationen, wie beispielsweise der Abstract, ein verknüpftes Bild oder etwa ein Homepage-Link. Weiterhin werden in diesem Schritt die verfügbaren Facetten abgefragt und an die jeweiligen Ressourcen geknüpft. Die Facettenextraktion wird in Abschnitt 8.3.2 beschrieben.
6. Falls aus den vorherigen Arbeitsschritten Ressourcen gefunden wurden, wird die Facettenliste in ein entsprechendes XML-Antwort Format geschrieben. Dieser Vorgang wird von der Klasse `XMLWriter` geregelt. Die Syntax und der generelle Aufbau des für die Arbeit eigens entwickelte Format, wird in Abschnitt 7.3 aus dem Kapitel Architektur beschrieben.
7. Neben DBpedia Ressourcen bietet das MobileFacets System die Integration von Event-Seiten. In dieser Sequenz wird die Facette *Event* bedient, in dem die Webservices der Seiten Upcoming.org und Eventful.com abgefragt werden. Die Event-suche wird initiiert, falls der Parameter `searchEvents` den Wert *true* besitzt. Für die Suche nach umliegenden Events, wird analog zur Geonames-Suche, die Parameter `location` und `distance` verwendet, die durch die Klasse `ServletHelper` ermittelt wurden. Die Funktion `findNearbyEvents` aus den Klassen `Eventful` und `Upcoming` sorgen dafür, dass umliegende Events gesucht werden. Bei einem positiven Ergebnis werden diese mithilfe des `XMLWriters` geschrieben und für den Client bereitgestellt.
8. Die Schritte **8** bis **11** behandeln im Gegensatz zu den vorherigen Schritten, die Suche anhand eines Schlüsselwortes. Die Suche nach einem Schlüsselwort wird durch den Benutzer initiiert, in dem er die freie Suche nach Instanzen verwendet (siehe Abschnitt 8.2.1, Paragraf:

Freie Suche nach Instanzen). Wie auch in der Kartensuche kann es eine Differenzierung in den verwendeten Suchrepositories geben.

9. Falls der Parameter `searchEvents` den Wert `true` besitzt, wird eine DBpedia-Suche eingeleitet. In diesem Schritt wird mittels des Parameters `keyword` die Funktion `checkKeywordLocation` aufgerufen, die die Aufgabe hat, anhand des Schlüsselwortes einen Ort zu finden. Dazu wird folgenden Abfrage generiert:

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
SELECT DISTINCT ?lat ?long
WHERE {
    <http://dbpedia.org/resource/Koblenz> geo:lat ?lat .
    <http://dbpedia.org/resource/Koblenz> geo:long ?long
}
```

Listing 6: DBpedia Ortssuche (SPARQL-Abfrage)

Falls eine Ressource (in diesem Beispiel Koblenz) eine Geokoordinate besitzt (durch die Variablen `?lat` und `?long` repräsentiert), handelt es sich um einen Ort, der für weitere Anfragen genutzt werden kann. Schematisch wird eine erfolgreiche Ortssuche durch den Punkt **A** in der Abbildung dargestellt. Anhand der Ressource und seinen Koordinaten wird die ab in Schritt **3** beschriebene Prozesskette gestartet. Da die freie Suche keine Distanz als Parameter beinhaltet, wird ein konstanter Faktor für die Umkreissuche verwendet.

10. Die Funktion `keywordSearch` wird aufgerufen, falls es sich bei dem Suchbegriff um keinen Ort handelt. Mit Hilfe von SPARQL lässt sich eine Freitextsuche durchführen. Die folgende Abfrage sucht in der DBpedia nach dem Suchbegriff "Bob Marley":

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?x FROM <http://dbpedia.org>
WHERE {
    ?x rdfs:label ?y . ?y bif:contains "'Bob Marley'" .
    ?x skos:subject ?z .
}
LIMIT 80
```

Listing 7: DBpedia Schlüsselwort-Suche (SPARQL-Abfrage)

Die Suche wird realisiert durch die Komponente `bif:contains`, die aus allen DBpedia-Labels passende Treffer liefert. Bei der Funktion `bif:contains` handelt es sich um keine standardmäßige SPARQL-Funktion. Es ist eine sogenannte *built-in-function* (steht für `bif`) der Openlink Virtuoso Datenbank<sup>52</sup>, die den SPARQL-Endpunkt von DBpedia realisiert. Die Suche wird auf einer volltext-indizierten Datenbank durchgeführt und ist laut Angaben von Virtuoso signifikant schneller, als wenn die von SPARQL bereitgestellte Funktion `regex` verwendet wird<sup>53</sup>. Relevant für weitere Suchvorgänge sind die gefundenen Subjekte, hier durch `skos:subject` definiert. Die maximale Anzahl der Suchtreffer ist auf 80 limitiert, da es sein kann, dass der Benutzer unpräzise Suchanfragen stellt. Zum Beispiel würde das Suchwort "the" eine zu große Anzahl an Ressourcen liefern, da viele Artikel aus der englischen Wikipedia das Wort "the" als Artikelname haben.

11. Dieser Aufruf ist verhält sich äquivalent zu Schritt 5. Hier werden die aus der freien Textsuche gefundenen Ressourcen mit Details versehen. der `XMLWriter` sorgt im Anschluss dafür, dass die Ergebnisse in ein entsprechendes XML-Format geschrieben werden.
12. Das MobileFacets System bietet neben der freien Suche nach DBpedia Ressourcen auch eine Schlüsselwort-basierte Suche nach Events an. Die Funktion `keywordSearch` sucht in den angebundenen Webservices `Eventful` und `Upcoming` nach Events, die dem Schlüsselwort entsprechen. Die Anfragen an jeweilige Event-Seiten werden im folgenden Abschnitt ausführlich erläutert.

### 8.3.2 Integration der Datenrepositories und Extraktion von Facetten

In diesem Abschnitt wird die Integration der einzelnen Datenrepositories und die Extraktion von Facetten behandelt. Es werden Beispiele zur Datenabfrage gegeben und es wird explizit auf einzelne Facettentypen eingegangen. Die Anbindung erfolgt über die Server Komponente "Data Provider", welche in der Abbildung 17 auf Seite 59 schematisch dargestellt wurde. Lediglich die Fotosuche mittels Flickr wird über den Client selbst geregelt.

---

<sup>52</sup><http://virtuoso.openlinksw.com/>

<sup>53</sup><http://docs.openlinksw.com/virtuoso/rdfsparql.html>

## Geonames

Geonames bietet die Suche nach nah-gelegenen Wikipedia-Ressourcen an. Das wird realisiert durch folgende Anfrage an den Webservice welche unter <http://ws.geonames.org/> erreichbar ist. Die folgende Anfrage wird genutzt, um gezielte Informationen aus der Umgebung zu finden.

```
http://ws.geonames.org/findNearbyWikipedia?lat=50.4302&lng=7.612371&radius=10&maxRows=1000&lang=en
```

### Listing 8: Geonames-Abfrage (URL)

Die Parameter `lat`, `lng` und `radius` werden durch den Client errechnet. Die Berechnungsvorschrift wurde in Abschnitt 7.2 erläutert. Anhand dieser Werte werden von Geonames umliegende Informationen zurückgeliefert, wobei die maximale Anzahl auf 1000 Informationseinheiten beschränkt ist. Es mag verwundern, dass es sich hierbei um eine *findNearbyWikipedia* Suche handelt, also keine explizite Suche nach DBpedia Ressourcen. Jedoch wird eine gelieferte Wikipedia-URL, beispielsweise [http://en.wikipedia.org/wiki/Deutsches\\_Eck](http://en.wikipedia.org/wiki/Deutsches_Eck), genutzt, um daraus eine äquivalente DBpedia-Ressource zu formen. Lediglich das Subjekt einer URL, in diesem Beispiel *Deutsches\_Eck* ist relevant für weitere Arbeitsschritte. Das Mapping eines Identifiers auf eine DBpedia Ressource ist einfach durch ein Austausch des URL-String <http://en.wikipedia.org/wiki/> durch <http://dbpedia.org/> gelöst.

**Geonames-Facetten** Anhand des Sequenzdiagramms aus Abschnitt 8.3.1 sieht man, dass Geonames die erste Station ist, um gezielt Informationen aus der Umgebung zu finden und diese dann für weitere DBpedia-Abfragen zu nutzen. Da Geonames ortsbasierende Informationen anbietet, werden diese Informationen von der Server-Komponente direkt als *Place*-Facettentyp deklariert. Eine DBpedia-äquivalente Abfrage zu umliegenden Orten ist auch möglich, jedoch hat sich bei einem kurzen Vergleichstest herausgestellt, dass Geonames die Ergebnisse schneller liefert.

## Dbpedia

Die Integration von DBpedia wird realisiert durch gezielte SPARQL-Anfragen an einen Webservice, welcher unter der URL <http://dbpedia.org/sparql> zu erreichen ist. Im vorherigen Kapitel wurde schon kurz auf die verschiedenen Abfragen bei der Verarbeitung eines Client-Requests eingegangen. In diesem Abschnitt geht es vorwiegend darum, welche Daten aus der DBpedia relevant

sind und inwiefern sich Facetten aus DBpedia-Ressourcen extrahieren lassen. Zuerst wird eine Übersicht gegeben, in der alle relevanten Eigenschaften einer Ressource aufgelistet sind.

## Dbpedia-Properties

- `rdfs:label`: Das Label einer Ressource definiert den anzuzeigenden Titel, beispielsweise aus der Instanz-Ebene des Browse & Search Reiter. In der DBpedia wird das Label einer Ressource in verschiedenen Sprachen angegeben. Für diese Arbeit wird der englische Titel abgefragt.
- `dbpedia-owl:abstract`: Der Abstract beschreibt die Ressource in wenigen Sätzen und stellt eine kurze Übersicht der wichtigsten Informationen dar. Ein Benutzer des MobileFacets Clients kann den Abstract einer Ressource in der Detailansicht einsehen.
- `geo:lat`, `geo:long`: Diese beiden Eigenschaften werden genutzt, um bei einer freien Textsuche herauszufinden, ob es sich um einen realen Ort handelt. Erläutert wird diese Vorgehensweise anhand einer SPARQL-Abfrage in Abschnitt 8.3.1, Listing 6.
- `is dbpedia-owl:birthPlace of`: Bei der Umkreissuche werden die von Geonames gefundenen Ressourcen mithilfe dieser Eigenschaft genauer untersucht: Handelt es sich um einen Ort, lässt sich eine Referenzierung einer Personen zu diesem Ort herausfinden. Listing 5 zeigt beispielhaft eine Suche nach Personen, die mit der Ressource <http://dbpedia.org/resource/Koblenz> referenziert sind.
- `is dbpedia-owl:campus of`: Wie die vorherige Eigenschaft, nur dass hier statt des Geburtsortes einer Personen, örtliche Hochschulen gefunden werden.
- `is dbpedia-owl:deathPlace of`: Äquivalente Eigenschaft wie oben. Statt des Geburtsortes, der Todesort.
- `is dbpedia-owl:hometown of`: Durch diese Eigenschaften lassen sich viele Ressourcen aus der Facette *Organisation* finden, solange eine entsprechende Referenzierung vorhanden ist.
- `foaf:homepage` : Bei der Detailansicht ist es möglich, die Homepage der jeweiligen Ressource aufzurufen. Die Eigenschaft `foaf:homepage` kann dazu benutzt werden.

- **rdf:type**: Durch die Eigenschaft **rdf:type** werden Ressourcen aus der DBpedia typisiert. Die Eigenschaft **rdf:type** ist eine Instanz von **rdf:Property**, die verwendet wird, um eine Ressource zu einer Klasse zuzuordnen. Die Klassen sind in der DBpedia OWL-Ontology<sup>54</sup> definiert, die im Moment 259 Klassen beinhaltet. Im Paragrafen Klassifikation wird weiterführend auf die Klassifikation und die Extraktion von Facetten anhand dieser Eigenschaft eingegangen.
- **skos:subject**: SKOS<sup>55</sup> steht für "Simple Knowledge Organization System" (frei übersetzt "einfaches System zur Organisation von Wissen") und ist eine auf RDF basierende formale Sprache zur Kodierung von Dokumentations-sprachen. Dazu zählen Thesauri, Klassifikationen oder andere kontrollierbare Vokabellisten<sup>56</sup>. Wie auch die OWL-Klassifikationen werden diese für die Facetten-Extraktion verwendet.
- **dbpedia-owl:thumbnail**: Durch diese Eigenschaft können in der Detailansicht referenzierte Thumbnails aus der Wikipedia angezeigt werden. Der Wert besteht aus einer URL.
- **dbpprop:hasPhotoCollection**: Die Detailansicht bietet die Möglichkeit, referenzierte Bilder einer Ressource anzeigen zu lassen. Ein Flickr wrappr<sup>57</sup> erweitert DBpedia Ressourcen anhand von RDF Links zu Fotos, die auf Flickr hochgeladen sind. Für jede DBpedia Ressource generiert der wrappr einen Link, der die Ressource mittels referenzierten Fotos abbildet.
- **dbpedia-owl:birthDate**: Falls es sich bei einer Ressource um eine Person handelt, kann das Geburtsdatum abgerufen und in der Detailansicht des Clients angezeigt werden.
- **dbpedia-owl:deathDate**: Wie auch das Geburtsdatum lässt sich, wenn vorhanden, das Todesdatum anzeigen.
- **dbpedia-owl:birthPlace**: Der Geburtsort einer Ressource kann verwendet werden, um ein sogenannten "blended browsing" zu ermöglichen. Abbildung 26 aus dem Abschnitt Listenansicht verdeutlicht dies. Durch Anklicken des Geburtsortes (in diesem Beispiel "Nine Mile, Jamaica") startet das System eine neue Umkreissuche in diesem Ort.

<sup>54</sup><http://wiki.dbpedia.org/Ontology>

<sup>55</sup><http://www.w3.org/2004/02/skos/>

<sup>56</sup>weitere Informationen unter: [http://dbpedia.org/page/Simple\\_Knowledge\\_Organization\\_System](http://dbpedia.org/page/Simple_Knowledge_Organization_System)

<sup>57</sup><http://www4.wiwiss.fu-berlin.de/flickrwrappr/>

- `dbpedia-owl:deathPlace`: Identisch zur vorherigen Beschreibung, nur dass statt des Geburtsortes der Todesort verwendet wird (im Beispiel Bob Marley der Ort Miami).

Beispielhaft eine SPARQL-Detailabfrage mit der Ressource [http://dbpedia.org/resource/Bob\\_Marley](http://dbpedia.org/resource/Bob_Marley):

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX dbprop: <http://dbpedia.org/property/>
SELECT ?abstract ?homepage ?birthdate ?deathdate ?facet ?
       thumbnail ?photo ?birthplace ?deathplace ?skosfacet ?
       owlfacet
WHERE {
    {<http://dbpedia.org/resource/Bob_Marley>
      dbowl:abstract ?abstract
      FILTER ( lang(?abstract) = "en" )}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      foaf:page ?homepage}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbowl:birthDate ?birthdate}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbowl:deathDate ?deathdate}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbowl:thumbnail ?thumbnail}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbprop:hasPhotoCollection ?photo}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbowl:birthPlace ?birthPlace}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      dbowl:deathPlace ?deathPlace}}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      skos:subject ?skosfacet}
    UNION
    {
      <http://dbpedia.org/resource/Bob_Marley>
      rdf:type ?owlfacet}
}
```

Listing 9: DBpedia-Detailabfrage (SPARQL)

**DBpedia-Facetten** Abbildung 31 zeigt ein Auszug der Ressource Koblenz, welche vollständig unter der URL <http://dbpedia.org/resource/Koblenz> eingesehen werden kann. Viele DBpedia Ressourcen sind mithilfe der Eigenschaft `rdf:type` klassifiziert. Die wichtigen Werte sind hier die Einträge unter `dbpedia-owl`, anhand derer das System die Zuordnung zu den High-Level Facetten vornehmen kann. Dies wird realisiert durch ein statisches Mapping der OWL-Klassifikationen zu den vordefinierten High-Level Facetten (zur Erinnerung: *Place*, *Person*, *Organisation*, *Work*, *Species*, *Event*). In dem Beispiel von Koblenz würde ein Mapping der Ressource in die Facette *Place* stattfinden, da eine Zuordnung der Subfacetten *Town*, *Settlement*, *PopulatedPlace* definiert ist. Die High-Level Facette als solche (`dbpedia-owl:Place`), wird bei der Zuordnung nicht beachtet.

Bei der Implementierung der Facettenkomponente stellte sich die Frage, inwiefern die Subfacetten-Tiefe realisiert werden sollte. In diesem Beispiel sieht die Hierarchie-Ebene wie folgt aus:

- `dbpedia-owl:Town` ist `rdfs:subClassOf` von `dbpedia-owl:Settlement`
- `dbpedia-owl:Settlement` ist `rdfs:subClassOf` von `dbpedia-owl:PopulatedPlace`
- `dbpedia-owl:PopulatedPlace` ist `rdfs:subClassOf` von `dbpedia-owl:Place`
- `dbpedia-owl:Place` ist `rdfs:subClassOf` von `owl:thing`

<code>rdf:type</code>	<ul style="list-style-type: none"> <li>▪ <code>yago:Location100027167</code></li> <li>▪ <code>owl:Thing</code></li> <li>▪ <code>yago:CitiesInRhineland-Palatinate</code></li> <li>▪ <code>yago:CitiesOnTheRhine</code></li> <li>▪ <code>gml:_Feature</code></li> <li>▪ <code>dbpedia-owl:PopulatedPlace</code></li> <li>▪ <code>dbpedia-owl:Settlement</code></li> <li>▪ <code>dbpedia-owl:Place</code></li> <li>▪ <code>dbpedia-owl:Town</code></li> </ul>
-----------------------	---

Abbildung 31: OWL Klassifikation einer DBpedia Ressource

Anhand dieser Hierarchie könnte man eine Tiefe von maximal vier Ebenen erreichen. Das würde bedeuten, ein Nutzer könnte die Suchmenge präziser definieren, da eine weitaus größere Anzahl von Subfacetten-Ebenen vorhanden ist. Andererseits gibt es hier ein Trade-Off zwischen Präzision des Suchvorgangs und der Benutzerfreundlichkeit hinsichtlich der Usability. Bei einer Hierarchie-Tiefe von vier oder größer könnte das zulasten der Übersicht

während der Navigation durch die Facetten gehen. Bei dieser Arbeit wurde die Hierarchie-Tiefe auf maximal drei beschränkt, wobei die dritte Ebene die Instanz-Ebene ist. Das heißt, in diesem Beispiel würde keine Unterscheidung zwischen *PopulatedPlace* und *Town* gemacht; beide tauchen unter der High-Level Facette *Place* als Subfacetten auf.

Neben der OWL-Klassifizierung besitzen viele DBpedia-Ressourcen YAGO-Klassen. YAGO ist eine semantische Wissensbasis, in der über zwei Millionen Entitäten und darüber hinaus etwa 20 Millionen Fakten dieser Entitäten enthalten sind [49]. Diese Fakten werden automatisch mithilfe von WordNet<sup>58</sup> aus der Wikipedia extrahiert. Die Ontology beinhaltet eine Is-A Hierarchie und kann äquivalent zur vorherigen Methodik genutzt werden, um Facetten aus den DBpedia-Ressourcen zu extrahieren.

## Eventful

Die Integration von Eventful ist für die Facette *Event* von Bedeutung: Mithilfe des Webservices, welcher unter <http://api.eventful.com/rest> erreichbar ist, lassen sich zum einen Events in der Umgebung finden, zum anderen kann eine Schlüsselwort-basierte Suche vorgenommen werden. Die folgende URL zeigt beispielhaft eine Suche von Events in der Umgebung von Koblenz.

```
http://api.eventful.com/rest/events/search?location_
=50.3302,7.492371&app_key=tDNDLZ4Q9ZhrG7cS&within=10&
units=km&include=categories
```

Listing 10: Eventful RESTful-Umkreissuche (URL)

In Anlehnung an die URL von Geonames, werden auch hier die Parameter `location` und `within` durch den Client bestimmt. Der Parameter `app_key` dient für den Authentifizierungsvorgang des Servers. Die nächste URL zeigt eine Schlüsselwort-basierte Suche nach "Koblenz".

```
http://api.eventful.com/rest/events/search?keywords=Koblenz&
app_key=tDNDLZ4Q9ZhrG7cS&include=categories
```

Listing 11: Eventful RESTful-Schlüsselwortsuche (URL)

Es werden Events gefunden, die im Titel, der Beschreibung oder beispielsweise in Namen des Austragungsortes Übereinstimmungen mit dem Schlüsselwort besitzen.

**Eventful-Facetten** Der Parameter `include` mit seinem Wert `categories` dient dazu, um aus den gefundenen Events Facetten zu finden. Events werden

---

<sup>58</sup><http://wordnet.princeton.edu/>

anhand ihrer Kategorie in unterschiedliche Bereiche zugeordnet. Geläufige Kategorien sind:

- Music
- Performing/ Visual Arts
- Media
- Social
- Education
- Commercial

Anhand dieser Kategorien können alle Subfacetten der High-Level Facette *Event* spezifiziert und für den Nutzer sichtbar gemacht werden.

## Upcomig

Upcoming ist neben Eventful, die zweite Eventseite, aus denen Informationen für die Facette *Event* gefunden werden. Die Anbindung erfolgt ähnlich wie bei Eventful. Anhand der Parameter *location* und *mi* lässt sich eine Umkreissuche von Events realisieren. Im Folgenden eine Beispielanfrage:

```
http://upcoming.yahooapis.com/services/rest/?method=event.┘  
getBestInPlace&api_key=369ac5e375&location┘  
=50.3302,7.492371&mi=7&per_page=100
```

Listing 12: Upcoming RESTful-Schlüsselwortsuche (URL)

**Upcoming-Facetten:** Auch hier liefert Upcoming Events mit kategorischen Informationen. Es gibt insgesamt dreizehn Kategorien, die als Subfacetten dienen können. Im vorherigen Paragraphen befindet sich eine Auflistung dieser, wobei sie einen Auszug darstellt.

## Flickr

Die Integration von Flickr erfolgt über einen Webservice, der umfangreiche Funktionen zur Verfügung stellt. Das RESTful Interface ist unter der Adresse <http://api.flickr.com/services/rest/> erreichbar. Die Suche nach Fotos wird vom MobileFacets System nicht wie bei den vorherigen Datenrepositoires von der Server Komponenten geregelt, sondern die installierte Anwendung auf dem Client kümmert sich um die Beschaffung der Daten. Dieser Designschritt

wurde gewählt, da es bei der Flickr-Integration keine vorverarbeiteten Prozesse wie beispielsweise eine Facettenextraktion geben muss. Die Server-Komponente als Intermediär der Datenbeschaffung von Fotos hätte zur Folge, dass die Suche unnötig länger dauern würde.

Die Datenbeschaffung wird vom Client durch den `MobileFacetsController` realisiert und kann auf zwei verschiedene Vorgehensweisen durchgeführt werden:

- Foto-Umkreissuche
- Schlüsselwort-basierte Fotosuche

Die Foto-Umkreissuche wird durch die Methode

```
public ArrayList<String> searchNearbyFlickrImages(int numImages)
```

initiiert. Der Parameter `numImages` wird durch die gewählte Anzahl der Flickr-Fotos in den Anwendungs-Einstellung gesetzt. Die Suchanzahl kann zwischen 10 und 70 Fotos variieren. Der Rückgabewert besteht aus eine Liste von gefundenen Flickr-URLs.

```
http://api.flickr.com/services/rest/?method=flickr.photos.
search&api_key=37dfe1fb9666a18fbba548d300d7196e&lat
=50.361&lon=7.581&radius=13&extras=url_s,url_m,geo&
per_page=20&min_upload_date=2008-10-10
```

Listing 13: Flickr-RESTful Umkreissuche (URL)

Listing 14 veranschaulicht eine generierte URL für die Umkreissuche. Die `flickr.photos.search` Methode bekommt als Übergabeparameter einen persönlichen API-Key, der für den Authentifizierungsvorgang gebraucht wird. Die Parameter `lat`, `long` und `radius` werden durch den Controller ermittelt. Auf die Berechnung dieser Werte wird in Abschnitt 7.2 eingegangen. Erwähnenswert ist zudem der Parameter `min_upload_date`, bei dem gefundene Fotos jünger als das angegebene Datum sein müssen. Diese Restriktion ist von Nöten, da die Flickr-API bei einer Umkreissuche ohne diesen Parameter nur die Fotos der letzten zwölf Stunden liefern würde.

Bei einer Schlüsselwortsuche handelt es sich um eine freie Textsuche die Fotos zurückliefert, die Übereinstimmungen im Titel, der Beschreibung oder anhand der Tags aufweisen. Das folgende Listing zeigt beispielhaft eine generierte URL, die das Schlüsselwort "Bob Marley" beinhaltet.

```
http://api.flickr.com/services/rest/?method=flickr.photos.
search&api_key=37dfe1fb9666a18fbba548d300d7196e&text=Bob+
```

```
Marley&extras=url_s,url_m,geo&per_page=20&min_upload_date_
=2008-10-10
```

Listing 14: Flickr-RESTful freie Suche (URL)

Die URL ähnelt sich stark der vorherigen Umkreissuche, jedoch sind die Parameter `lat`, `long` und `radius` durch den Parameter `text` ersetzt, der den Suchbegriff signalisiert.

## 9 Evaluation von MobileFacets

### 9.1 Vorgehen

Die Evaluation des MobileFacets System soll zeigen, wie intuitiv sich das System bedienen lässt und inwiefern die Funktionalitäten ausreichend sind. Gemäß den nicht-funktionalen Anforderungen auf Seite 57 soll validiert werden, inwiefern die die Usability des Systems gewährleistet wird. Weiterhin soll diese Evaluation zeigen, ob die Qualität der zur Verfügung stehenden Facetten in Ordnung ist.

Die Evaluation wird wie folgt durchgeführt: Die Teilnehmer werden anhand eines definierten Szenarios insgesamt sechs Teilaufgaben lösen, wobei fünf der Aufgaben sich an das Szenario orientieren. Das Szenario ist so aufgebaut, dass alle Funktionalitäten des System einzeln abgearbeitet werden. Dabei erhöht sich je Teilabschnitt die Komplexität der Aufgaben. Das hat den Hintergrund, dass die Teilnehmer sich im Laufe der Evaluation in das System einfinden müssen. Nach jeder Teilaufgabe des Szenarios werden explizite Fragen zu dieser Thematik gestellt. Die Teilnehmer haben zudem die Möglichkeit, Kommentare, Anregungen und Kritiken in Form eines Freitextes nach jeder Aufgabe zu äußern. Die letzte Aufgabe ist unabhängig des Szenarios und zielt darauf ab, das Gesamtsystem in Bezug auf die Funktionalität und Usability zu bewerten. Währenddessen werden nur kleine Tipps gegeben, falls irgendwelche Sachverhalte unklar sind.

### 9.2 Testpersonen

Bei den Testpersonen handelt es sich vorwiegend um wissenschaftliche Mitarbeiter der Universität Koblenz aus dem WeST Institut<sup>59</sup>, sowie um Studenten aus dem Bereich der Informatik und Computervisualisik. Außerdem nahm eine Person aus dem pädagogischen Bereich teil. Insgesamt nahmen 15 Teilnehmer, davon 14 männliche Teilnehmer und eine weibliche Person an der Evaluation teil. Dies kommt zustande, da im Informatikbereich vorwiegend männliche Personen aktiv sind. Das Alter der Teilnehmer variiert zwischen 22 und 32 Jahren und ergibt im Mittel einen Alterskursschnitt von 28 Jahren. Da es sich um eine Evaluation im mobilen Bereich handelt, war es interessant zu wissen, inwiefern die Teilnehmer Erfahrungen im Bereich von mobilen Endgeräten mitgebracht haben. Alle Teilnehmer besitzen mindestens ein mobiles Endgerät in Form eines Handys. Mehr als zehn Personen besitzen zudem ein modernes Smartphone oder einen Tablet-PC. Elf Personen haben weiterhin schon einmal Erfahrungen mit einem Touchscreen-basierten Smartphone

---

<sup>59</sup><http://isweb.uni-koblenz.de/>

mitgebracht. Keiner der Teilnehmer konnte sich selbst als Experte im Gebiet von mobilen Endgeräten einschätzen, jedoch sieben gaben an, dass sie sich zumindest gut mit diesen Geräten auskennen. Diese gewisse Mischung an Erfahrungswerten war für die Evaluation von Vorteil, da ein Hauptaugenmerk auf die Usability des MobileFacets System gelegt wurde. Auch nicht versierte Touchscreen-basierte Anwender sollten sich schnell in das System einfinden und die Aufgaben ohne Probleme lösen können.

### 9.3 Aufgaben

Die Aufgaben der Evaluation lehnen sich an die Szenarien aus Kapitel 2 an, wobei einige Modifikationen hinsichtlich des Ablaufes vorgenommen sind. Das Szenario handelt von einem Mitarbeiter, der geschäftlich in Koblenz unterwegs ist und mit seinem mobilen Endgerät Informationen zur Umgebung akquirieren möchte. Das Szenario ist in fünf Einzelteile zerlegt, wobei jeder Arbeitsschritt verschiedene Funktionalitäten des Systems abdeckt. Dabei werden im jedem Arbeitsschritt die zur Verfügung stehenden Facetten genutzt, um für die Auswertung eine Diversifikation hinsichtlich der Facettenqualität und der Navigationsführung zu ermöglichen. Aufgabe (a) handelt von der Suche nach umliegenden POIs. Hier findet eine schrittweise Einführung in die Verwendung der Applikation anhand der Facette *Place* statt. Die nachfolgende Aufgabe hat das Ziel, berühmte Personen in der Umgebung zu finden und detaillierte Informationen zu diesen zu erfahren. Im nächsten Schritt werden gezielte POIs gesucht. In Anlehnung an die Tabelle 2.1 auf Seite 4 spiegeln die ersten drei Aufgaben einen gewissen Anteil von *Trivia* und *Point of Interests* wieder. Die nächsten beiden Aufgaben haben das Ziel, dass die Teilnehmer sich mit der Event-Funktionalität auseinandersetzen müssen. Das heißt, es müssen Events gesucht werden und Informationen zu diesen abgerufen werden. Die letzte Aufgabe handelt von der willkürlichen Suche nach Informationen in entfernten Gebieten, welches durch das System durchgeführt werden kann.

Nach jedem Arbeitsschritt wurden explizit Fragen gestellt, die in Abschnitt 9.4 eingesehen werden können. Im Folgenden das Szenario mit den Arbeitsschritten (a)- (e), welche auf einem mobilen Endgerät ausgeführt wurden.

#### (a) POI-Browsen

Stell dir vor, du bist Mitarbeiter der Berliner Software AG und an diesem Wochenende geschäftlich in Koblenz unterwegs. Nach einem Messebesuch am Samstagmorgen hast du noch jede Menge Zeit Koblenz etwas näher kennen zu lernen. Da du ein kultivierter Mensch bist, hast du Lust die

Sehenswürdigkeiten der Stadt zu erforschen. Nur mit einem Handy und der Anwendung MobileFacets machst du dich auf den Weg...

- Nimm dir dazu das Handy in die Hand und starte die Anwendung. Unmittelbar nach dem Start der Anwendung werden umliegende Informationen gesucht.
- Wenn du keine Erfahrung im Umgang mit einem Touchscreen-basierten Handy hast, mach dich erst einmal damit vertraut.
- Versuche nun alle "Places" der Umgebung auf der Karte anzeigen zu lassen.
- Du möchtest nun herausfinden wo der "Koblenzer Hauptbahnhof" in Koblenz ist. Verfeinere dazu die Suchmenge mit Hilfe der Facetten und lass dir nur den Koblenzer Hauptbahnhof auf der Karte anzeigen.
- Sehe dir im Anschluss Bilder auf der Karte an.

#### **(b) Personen-Suche**

Dein Interesse wurde geweckt und du machst dich auf den Weg zur "Herz-Jesu-Kirche" um sie in voller Größe zu bewundern.

- Suche dafür in allen "Places" und finde die exakte Position der Herz-Jesu-Kirche auf der Karte.

Nachdem dein Durst nach kulturellen Informationen gestillt ist, begibst du dich auf den Weg zum nächstgelegenen Café. Bei einem Latte Macchiato interessierst du dich für berühmte Persönlichkeiten der Stadt..

- Die Anwendung bietet neben dem Suchen nach POIs die Möglichkeit berühmte oder bekannte Persönlichkeiten der Stadt und der Umgebung zu finden.
- Wähle dazu die Facette Personäus und lasse dir die Sportler ("Athlete") der Stadt anzeigen.
- Lasse dir zusätzlich die Schauspieler ("Actor") der Stadt anzeigen.
- Suche in der Ergebnisliste nach "Bodo Illgner" und lies dir Details von ihm durch.
- Schau dir Bilder von "Bodo Illgner" an.

### (c) POI-Suche

Gut gestärkt und mit interessanten Informationen bereichert machst du dich auf den Weg zum Hauptbahnhof und trittst deine Heimreise nach Berlin an. Aus dem Zugabteil schaust du aus dem Fenster und entdeckst auf der Anhöhe den Koblenzer Fernmeldeturm. Neugierig fragst du dich wie hoch doch dieser im Vergleich zum Berliner Fernsehturm ist...

- Nimm dir dazu das Handy in die Hand und suche nach umliegenden Informationen. Dafür kannst du die Menüfunktion *What's around me* benutzen.
- Verfeinere die Suchmenge und lass dir nur die Türme in der Umgebung anzeigen.
- Schau im Abstract der Detailansicht nach wie hoch der Koblenzer Fernsehturm ist.
- Um die Höhe des Berliner Fernsehturmes nachzulesen starte die Textsuche im *Browse & Search* Reiter. Die Suche lässt sich über das *virtuelle Menü* starten.
- Tippe dazu einen geeigneten Suchbegriff in das Textfeld ein. Darunter werden zum Suchbegriff entsprechende Vorschläge angezeigt. Schicke die Suche ab und schau nach wie hoch der Berliner Turm ist.
- Gehe dazu auf die Webseite suche nach: Bauwerk & Geschichte ⇒ Technik.
- Schau dir Photos vom Berliner Fernsehturm in der Photo-Ansicht an.

### (d) Event-Browsen

Du fährst gerade mit dem Zug durch Hannover und überlegst dir, ob du nicht Abends noch was in Berlin "erleben" möchtest. Dazu nimmst du dein Handy in die Hand und lässt dir lokale Events von Berlin anzeigen...

- Die Anwendung bietet die Möglichkeit die Suche auf bestimmte Daten einzuschränken. Öffne dafür die *Preferences* und stelle die Suche so ein, dass nur nach Events gesucht wird.
- Da wir uns nun "gedanklich" in Hannover befinden aber das Gerät mithilfe der GPS- Koordinaten unseren Koblenzer Aufenthaltsort

bestimmt, muss der Suchkontext nach Berlin gewechselt werden. 1. Möglichkeit: Flippe und Zoome mithilfe der Karte nach Berlin und benutze die *What's around me* Funktion. 2. Möglichkeit: Suche nach Berlin mit Hilfe der *Suchfunktion*.

- Schau auf der Karte (oder Liste) welche Events heute stattfinden. Informationen zu den Event-POIs lassen sich in der Hilfe über das *virtuelle Menü* der Karte nachlesen.
- Lass dir nur Musik-Events anzeigen und schau nach welche Events am 9. Dezember statt finden (Benutze hierfür die *Zeitleiste*)

### (e) Event-Suche und weiteres Browsen

Du siehst dass am 9. Dezember ein Shakira Konzert in Berlin statt findet. Aus purer Neugier möchtest du wissen wann und wo weitere Konzerte von Shakira sind...

- Suche nach Shakira und lasse dir alle Events von ihr anzeigen.
- Suche das Konzert in Dublin und lass dir die genaue Position des Konzertes auf der Karte anzeigen.

Dir fällt ein, dass du in drei Wochen auch geschäftlich zu einem Kunden in Dublin reisen wirst. Jetzt wäre es doch schön zu wissen, was es alles dort zu entdecken gibt...

- Lass dir alle Wikipedia-Einträge, Events sowie Photos zu Dublin anzeigen und browse ein wenig...

## 9.4 Auswertung und Analyse

Für jede Testperson wurde ein Fragenkatalog ausgeteilt, der zum Ziel hatte, gezielt die einzelnen Arbeitsschritte des Szenarios zu hinterfragen. Neben ordinal skalierten Fragen hatte jeder Teilnehme die Möglichkeit, Kommentare, Anregungen oder Kritiken in Form eines Freitextes zu verfassen. Die ordinal skalierten Fragen haben eine Rangfolge von 1-5, wobei sie synonym für die Begriffe "trifft nicht zu", "trifft kaum zu", "trifft teilweise zu", "trifft zu und trifft voll" stehen. Diese Skala stammt aus der "IsoMetrics usability inventory", in der Designprinzipien für die Usability von Software-Systemen definiert sind [50].

Die folgenden Abschnitte behandeln die Analyse der Fragen, wobei zuerst in den Abschnitten 9.4.1 bis 9.4.5 separat jede Aufgabe diskutiert wird. In

den anschließenden vier Teilabschnitten wird das Gesamtsystem betrachtet. Bei diesen Fragen geht es um die Angemessenheit der Arbeitsschritte, die Intuition der einzelnen Arbeitsschritte, die Anwendungsunterstützung bei der Durchführung einer Aufgabe und die Erwartungskonformität des Systems. Im Anschluss werden die allgemeinen Fragen zum System diskutiert.

#### **9.4.1 Aufgabe A: POI-Browsen**

In diesem und den folgenden Abschnitten wird detailliert auf die einzelnen Arbeitspakete und deren Auswertungen eingegangen. Die letzten vier Fragen sind dabei immer die gleichen. Die nachfolgenden Fragen beziehen sich auf die spezifischen Aufgaben und sollen die Funktionalitäten abdecken. Es werden hierbei nur einzelne Fragen diskutiert.

Tabelle 3 zeigt den ersten Fragenblock des Szenarios. Hier geht es um das allgemeine Browsen nach Point of Interests. Da es sich um die erste Aufgabe handelt, sind noch keine komplizierten Arbeitsschritte enthalten. Lediglich ein einfaches Auswählen der Facette *Place* und einer Subfacette bildet die Aufgabe. Die Aufgabe lässt sich in Abschnitt 9.3 nachlesen.

#### **Angemessenheit und Intuition**

Die Angemessenheit der Arbeitsschritte um die Facette *Place* und *Railway Stations* auszuwählen wurde mit 4.24 Punkten und einer Standardabweichung von 0.24 Punkten bewertet. Es handelt sich hierbei um sehr gute Werte. Kaum ein Proband hatte Probleme diese Aufgabe zu lösen.

#### **Anwendungsunterstützung und Erwartungskonformität**

Auch die Anwendungsunterstützung und Erwartungskonformität wurden mit 4.24 beziehungsweise 3.76 überdurchschnittlich bewertet. Für viele Teilnehmer war einerseits der Umgang mit den Facetten neu, andererseits haben viele zum ersten Mal mit einem Touchscreen-basierten Handy gearbeitet. Somit konnten einzelne Navigationsschritte nicht zu 100% nachvollzogen werden.

#### **Detailfragen**

Die sofortige Suche nach umliegenden Facetten bei Programmstart haben alle Teilnehmer mit 4.73 Punkten bewertet und zeigt somit, dass dies eine sehr gute Funktion ist. Hier muss jedoch beachtet werden, dass je nach Ort und Anzahl der verfügbaren Facetten der Suchvorgang länger dauern kann. Bei diesem Szenario war eine Suche in Koblenz und Umgebung notwendig.

	Mittelwert	Standard- abweichung
Das sofortige Suchen nach umliegenden Informationen bei Anwendungsstart erachte ich als sinnvoll.	4.73	0.2
Der Browse & Search Reiter enthält alle notwendigen Informationen die für das Browsen und Auswählen von Facetten relevant sind.	3.34	0.32
Das Verfeinern der Suchmenge über die Facetten (wie im Beispiel: Anzeige der Bahnhöfe) ist für mich nachvollziehbar.	4.19	0.44
Auf der Kartenansicht finde ich alle Informationen die ich für diesen Suchkontext benötige.	4.03	0.44
Photos werden auf der Karte angezeigt. Ich finde diese Funktion sinnvoll für diesen Suchkontext.	3.33	0.48
Die von der Anwendung gelieferten Suchergebnisse entsprachen meinen Erwartungen.	4.33	0.26
Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.	4.24	0.24
Die Durchführung der einzelnen Arbeitsschritte war intuitiv.	4.03	0.44
Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.	4.24	0.24
Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.	3.76	0.3

Tabelle 3: Kennzahlen: Aufgabe A: POI-Browsen

Die Suchzeit empfanden die Teilnehmer als angemessen. Die nächste Frage zielt darauf ab, inwieweit der erste Reiter "Browse & Search" alle notwendigen Informationen bereitstellt, um ein einfaches Suchen und Browsen von Facetten zu ermöglichen. Mit 3.34 Punkten und einer Standardabweichung von 0.32 Punkten traf dies bei den meisten Teilnehmern nur teilweise zu. Die Hauptkritikpunkte zum Browse & Search Reiter sind:

- Keine alphabetische Sortierung der Facetten und Instanzen
- Mehrdeutigkeiten von Facetten, beispielsweise Populated Places vs. Settlement vs. Town
- Keine Suchfunktion innerhalb von Facetten

Beim ersten Punkt vermissten viele eine alphabetische Sortierung der Facetten und Instanzen. Gerade bei einer großen Anzahl von Instanzen mussten die Teilnehmer lange durch die Liste scrollen, um das gewünschte Ergebnis zu finden. Hier sollte man für etwaige Implementierungen in der Zukunft eine Sortierung auf Instanz-Ebene anbieten können. Einige bemängelten die Mehrdeutigkeiten der Facetten. In der DBpedia-Klassifikation sind viele Ressourcen in mehreren Facetten untergeordnet, welches dazu führen kann, dass ein und dieselbe Ressource unter *Populated Place* und *Town* auftauchen kann. Weiterhin wünschten sich einige Teilnehmer eine Suchfunktion innerhalb der Facetten, um somit die angezeigte Facettenanzahl einschränken zu können.

Bei diesem Szenario mussten die Teilnehmer vorhandene Flickr-Fotos auf der Karte ansehen. Mit 3.33 Punkten im Durchschnitt wurde diese Funktion im Rahmen dieser Aufgabe bewertet. Kritikpunkte sind unter anderem, dass die Fotos immer auf der Karte angezeigt werden und somit teilweise die Sicht auf nahegelegene POIs überdeckte. Die Informationen, die auf der Kartensicht angezeigt werden, empfanden alle Teilnehmer mit 4.03 Punkten als gut und angemessen. Weiterhin kamen die von der Anwendung gelieferten Ergebnisse überdurchschnittlich gut an (4.33 Punkte).

#### **9.4.2 Aufgabe B: Personen Suche**

##### **Angemessenheit und Intuition**

Die Angemessenheit und Intuition dieser Aufgabe wurde tendenziell eher schlecht bewertet. Das lag daran, dass die Suche nach der Herz-Jesu Kirche in Koblenz zu mühselig empfunden wurde. Die Kirche ist unter der Facette *Landmark* untergeordnet, die eine sehr hohe Anzahl von Objekten beinhaltet. Bei ungefähr 80 Einträgen war die Suche innerhalb der Liste nicht intuitiv

	Mittelwert	Standard- abweichung
Die Suche nach Personen, die etwas mit der Stadt oder der Umgebung zu tun haben, erachte ich als eine nützliches Feature.	4.14	0.46
Die Detailansicht (z.B. von Bodo Illgner) enthält für die Aufgabe alle notwendigen Informationen.	4.2	0.33
Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.	3.83	0.37
Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.	3.73	0.39
Die Durchführung der einzelnen Arbeitsschritte war intuitiv.	3.27	0.45
Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.	3.55	0.46
Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.	3.41	0.36

Tabelle 4: Kennzahlen: Aufgabe B: Personen Suche

genug. Auch hier wäre eine alphabetische Sortierung innerhalb des Browse & Search Reiters von Vorteil gewesen. Wenige haben die Facette *Landmark* ausgewählt und sind danach auf die Listenansicht gewechselt. Diese Ansicht bietet bei ausgewählten Facetten aus dem Bereich *Place* eine aufsteigende Sortierung der Ressourcen nach ihrer Entfernung zum jeweiligen Standort an. Somit wurde die Durchführung der einzelnen Arbeitsschritte mit lediglich 3.27 Punkten bewertet.

### **Anwendungsunterstützung und Erwartungskonformität**

Wie eben schon erwähnt, leidet die Suche nach der Herz-Jesu Kirche unter der großen Anzahl der Instanzen, die unter der Facette *Landmark* untergeordnet sind. Nachdem viele die Kirche gefunden haben, sollten sie die exakte Position der Kirche auf der Karte ausfindig machen. Die Anwendung bietet in der Hinsicht keinen visuellen Hinweis über die ausgewählte Instanz. Durch Anklicken des "Show Locations" Knopfes in der Detailansicht wechselt die Ansicht auf die Karte und die jeweilige Instanz wird in der Kartenmitte zentriert. Hierbei könnte man sich eine Hervorhebung des POI-Symbols vorstellen. Beispielweise durch eine vergrößerte Darstellung oder ein farbliche Abhebung zu den anderen POIs.

### **Detailfragen**

Die Suche nach Personen, die semantisch mit der Facette *Place* verknüpft sind, erachten viele als ein nützliches Feature. Bewertet wurde diese Frage mit 4.14 Punkten und einer geringen Standardabweichung von 0.46. Dieses Feature hebt sich von vielen Anwendungen, die ein reines Suchen und Finden von Point of Interests anbieten, ab. Die Detailansicht zu Personen wurde mit 4.2 Punkten bewertet, und gilt als umfassend genug um den Informationsbedarf zu decken.

### **9.4.3 Aufgabe C: POI-Suche**

#### **Angemessenheit und Intuition**

Die Angemessenheit der Arbeitsschritte und die Intuition bei der Ausführung der Aufgabe wurde von den Teilnehmern mit 3.97 beziehungsweise 4.04 Punkten bewertet. Ein Teilnehmer fand es nicht intuitiv, die Ergebnismenge mithilfe der Facetten *Place* -> *Towers in Germany* einzuschränken. Das verwundert etwas, da viele der Umgang mit Facetten von Seiten wie Amazon oder Ebay kennen. Kontrovers betrachtet, war es für diesen Teilnehmer sichtlich neu, ein Umgang mit Facetten zu bewerkstelligen. Vielleicht hat er

	Mittelwert	Standard- abweichung
Die Suchfunktion lässt sich über das virtuelle Menü im „Browse & Search“ Reiter erreichen. Ich empfinde diese Vorgehensweise als sinnvoll.	4.53	0.22
Die Live-Vorschau der Suchergebnisse empfinde ich als eine hilfreiche Funktion.	4.66	0.26
Die Vorschläge der Live-Vorschau entsprechen meinen Erwartungen.	4.06	0.35
Der Wikipedia-Abstract ist umfangreich genug um die Aufgabe zu bearbeiten.	3.46	0.49
Die von der Anwendung gelieferten Suchergebnisse entsprachen meinen Erwartungen.	4.07	0.26
Die angezeigten Photos in der Photoansicht entsprechen meinen Erwartungen.	4.2	0.41
Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.	3.97	0.39
Die Durchführung der einzelnen Arbeitsschritte war intuitiv.	4.03	0.33
Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.	3.83	0.5
Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.	3.7	0.57

Tabelle 5: Kennzahlen: Aufgabe C: POI-Suche

die nötige Assoziation nicht gesehen, dass die Auswirkungen von Facetten in dieser mobilen Anwendung identisch funktionieren.

### **Anwendungsunterstützung und Erwartungskonformität**

Hier wurde bemängelt, dass dem Nutzer kein ausreichender Hinweis gegeben wurde, dass nur im Kartenausschnitt gesucht wurde. Im Vorfeld der Evaluation wurde diese Einschränkung den Teilnehmern mitgeteilt, jedoch erwarteten viele bei der Suche nach dem Koblenzer Fernsehturm, dass dieser unabhängig des ausgewählten Kartenausschnittes zu finden ist. In den Tests zeigte sich, dass manche den Kartenzoom zu klein gewählt haben, so dass die "What's around me?" Funktion den Turm nicht gefunden hat. Für spätere Implementationsarbeiten sollte ein entsprechender Hinweis geboten werden, der zeigt, dass nur in dem ausgewählten Kartenausschnitt gesucht wird. Weiterhin könnte man sich vorstellen, dass der Radius der Umkreissuche manuell durch den Benutzer definierbar ist.

### **Detailfragen**

Bei der ersten Frage geht es um die Integration der Suchfunktion im virtuellen Menü des Browse & Search Reiters. Das erachten fast alle Teilnehmer als eine gute Funktion und wurde mit insgesamt 4.53 Punkten bewertet. Um die Suche zu starten, benutzten dennoch viele den Hardware-Suchknopf, über dem die Suche unabhängig des gewählten Reiters aufgerufen werden kann. In diesem Arbeitsschritt wurde zum ersten mal die Suchfunktion zum Finden des Fernsehturmes Berlin genutzt, bei dem die Live-Vorschau der Ergebnisse zum Vorschein kam. Die Live-Vorschau erachten die Teilnehmer in diesem Suchkontext als eine sehr hilfreiche Funktion (4.66 Punkte). Ein wesentlicher Kritikpunkt hierbei war nur, dass beim Drehen des Gerätes die virtuelle Tastatur die Liste der Vorschläge überdeckte.

Die Teilnehmer mussten die Höhe des Fernsehturmes von Berlin herausfinden. Leider war der Abstract der DBpedia Ressource nicht umfangreich genug, um diese Aufgabe ohne den Umweg über die Homepage zu lösen. Das spiegelt sich in der Analyse wieder, wobei die Teilnehmer den Umfang des Wikipedia-Artikels nur mittelmäßig mit 3.46 Punkten bewertet haben. Das Problem liegt aber nicht an der Qualität und des Umfanges einer DBpedia-Ressource Es ist vielmehr eine "Problematik" bei den Quelldaten. Solange nicht im Abstrakt eines Wikipedia-Artikels alle signifikant, wichtigen Daten stehen, werden diese auch nicht bei der semantischen Analyse eines Artikels in eine entsprechende DBpedia-Ressource übernommen. Abhilfe könnte eine Ausweitung der Detailansicht sein, in dem je nach Facettentyp,

detailliertere Informationen bereitgestellt werden. In diesem Beispiel müsste beim Facettentyp *Towers in Germany*, zusätzlich die Eigenschaft *dbpedia-owl:height* integriert werden.

#### 9.4.4 Aufgabe D: Event-Browsen

##### Angemessenheit und Intuition

Bei diesem Arbeitspaket musste der Suchkontext von Koblenz nach Berlin gewechselt werden, um dort lokale Events zu finden. Eine Hauptaufgabe war zudem, den Suchbereich nur auf Events einzuschränken, welches über die Anwendungs-Einstellungen getätigt werden musste. Beide Schritte wurden von den Teilnehmern als ausreichend intuitiv und gemäß den Arbeitsschritten als angemessen gesehen. Mit 4.1 und 4.06 Punkten wurden diese beiden Schritte bewertet. Man sieht, dass die Streuung um den Mittelwert mit knapp 0.35 und 0.39 Punkten sehr gering ausfällt.

Intuitiv wurde auch das Verwenden der Zeitleiste bewertet. Die Teilnehmer haben das Funktionsprinzip schnell verstanden und konnte indes ohne Probleme "an der Zeit drehen". Weiterhin war die Darstellung der einzelnen Events auf der Karte angemessen. Insgesamt kann man also sagen, dass die Integration der Zeitleiste und die visuelle Darstellung aus der Arbeit von Daniel Schmeiss [48] eine sinnvolle Maßnahme im Kontext dieser Arbeit war.

##### Anwendungsunterstützung und Erwartungskonformität

Auch die allgemeine Anwendungsunterstützung für das Browsen von Events sahen mit 4.13 Punkten alle Teilnehmer als angemessen. Die Erwartungskonformität liegt mit 3.99 Punkten höher als bei den vorherigen Aufgaben. Insgesamt kann man sagen, dass die Teilnehmer mit der Aufgabe "Event-Browsen" besser zurecht kamen, als das Browsen nach nahegelegenen POIs. Das liegt zum einen daran, dass in dieser vierten Aufgabe die Teilnehmer den Navigationsfluss innerhalb der Anwendung besser kennen gelernt haben. Man kann hier von einer gewissen Eingewöhnungszeit sprechen, in der die Anwender Schritt für Schritt Vertrauter mit der Anwendung wurden. Zum anderen bekam der Nutzer nur die Facette *Event* gezeigt, da vorher die Suche nach Wikipedia-Artikel in den Anwendungs-Einstellungen deaktiviert wurde. Das führte dazu, dass es keinen Informationsoverhead wie beispielsweise bei der Suche nach der Herz-Jesu Kirche in der Facette *Landmark* gegeben hat.

	Mittelwert	Standard- abweichung
Den Suchkontext von Koblenz nach Berlin zu wechseln empfinde ich als einen mühseligen Vorgang. (über Karte oder Suchfunktion).	2.37	0.42
Ich vermisse eine Funktion, die mir bei Eingabe eines Ortes direkt diesen auf der Karte anzeigt.	3.54	0.63
Die Suche auf bestimmte Daten (Wikipedia/Flickr/Events) einzuschränken empfinde ich als sinnvolle Funktion.	4.26	0.41
Die Einschränkung der zu suchenden Daten lässt sich in den Preferences festlegen. Ich empfinde diese Vorgehensweise als sinnvoll.	3.6	0.48
Events werden in Abhängigkeit des Startdatums durch unterschiedlich farbige Sterne markiert. Die Darstellungsweise empfinde ich als sinnvoll.	3.83	0.37
Events werden in in Subfacetten unterteilt (Musik, Social, Sport,...). Diese Einteilung in Subfacetten empfinde ich als sinnvoll.	3.74	0.41
Die Detailansicht von Events enthält alle nötigen Informationen die ich benötige.	4.53	0.22
Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.	4.14	0.28
Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.	4.1	0.35
Die Durchführung der einzelnen Arbeitsschritte war intuitiv.	4.06	0.39
Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.	4.13	0.32
Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.	3.99	0.4

Tabelle 6: Kennzahlen: Aufgabe D: Event-Browsen

## Detailfragen

Die erste Detailfrage handelt von dem Wechseln des Suchkontextes Berlin nach Koblenz. 14 der 15 Teilnehmer haben sich für die Suchfunktion statt des manuellen Wechsels des Kartenausschnittes entschieden. Mit nur 2.37 Punkten wurde dieser Vorgang als mühselig angesehen. Bei einer Invertierung der Frage (hier repräsentiert ein Punkt den besten Wert) käme man auf nur lediglich 2.36 Punkten, was ein schwacher Wert ist. Hier muss man sich die Frage stellen, warum die Eingabe des Suchwortes "Berlin" in die Suchmaske für die Teilnehmer als mühselig angesehen wurde. Möglicherweise war es auch nicht intuitiv genug, nach Facetten durch Eingabe eines Ortes zu suchen. Hier besteht noch Verbesserungsbedarf.

Den Suchbereich auf bestimmte Daten einzuschränken (Wikipedia, Events, Flickr) empfanden mit 4.26 und einer Streuung von 0.41 die meisten Teilnehmer als eine hilfreiche Funktion. Jedoch zeigt die nächste Frage (Einschränkung der Daten unter den Preferences), dass dieser Schritt nicht ideal ausgefallen ist. Kritikpunkt ist hier, dass es eventuell besser wäre, die Einschränkung der Suche unter dem Search & Browse Reiter vorzunehmen.

Events lassen sich in Abhängigkeit des eingestellten Datums visuell unterschiedlich darstellen. Diese Vorgehensweise wurde insgesamt gut aufgenommen, wobei dies für die meisten Teilnehmer neu war. Dennoch zeigt die Kennzahl (3.83 Punkte), dass sich die Zeitleiste intuitiv bedienen lässt. Der Informationsgehalt der Detailansicht wurde mit 4.53 Punkten aller Teilnehmer als umfangreich genug gesehen. Manche kritisieren, dass es schön wäre, wenn weitere Informationen zu Events, wie Preise, Verfügbarkeiten oder einen direkten Link auf die Veranstalterseite angezeigt würden. Weiterhin vermisst ein Teil der Anwender, dass die Listenansicht sich nicht dynamisch in Abhängigkeit der eingestellten Zeit verändert hat. Hier werden alle Events, ohne Einfluss des Datum, angezeigt. Ein Benutzer stellte die Zeitleiste um, bevor er die Suche nach Events in Berlin startete. Deswegen wurden bei ihm teilweise keine Ergebnisse angezeigt, da keine Event in Abhängigkeit der eingestellten Zeit, in den nächsten 30 Tagen statt gefunden haben. Diese Umstand kann man lösen, in dem die Zeiteinstellung auf das heutige Datum zurückgestellt wird, wenn Nutzer eine neue Suche starten.

### 9.4.5 Aufgabe E: Event-Suche und weiteres Browsen

#### Angemessenheit und Intuition

Im letzten Aufgabenblock wurde das freie Browsen in entfernten Orten betrachtet und beurteilt. Mit einem Mittelwert von 4.4 und einer Standardab-

	Mittelwert	Standard- abweichung
Die Suche nach Events erachte ich als eine sinnvolle Funktion.	4.53	0.28
Das Browsen nach Informationen in entfernten Orten (Dublin) erachte ich als eine sinnvolle Funktion.	4.64	0.26
Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.	4.0	0.36
Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.	4.4	0.22
Die Durchführung der einzelnen Arbeitsschritte war intuitiv.	4.19	0.33
Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.	4.13	0.36
Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.	4.07	0.45

Tabelle 7: Kennzahlen: Aufgabe E: Event-Suche und weiteres Browsen

weichung von lediglich 0.22 Punkten wurden die erforderlichen Arbeitsschritte als sehr angemessen gesehen. Auch die Durchführung der Aufgabe wurde als intuitiv wahrgenommen. Insgesamt haben viele Teilnehmer angegeben, dass sich die Anwendung im Laufe der Aufgabenpakete wesentlich besser und intuitiver bedienen lässt. Das kann daran liegen, dass die notwendigen Funktionen schneller gefunden werden und sich eine gewisse Gewöhnung mit dem Umgang von Facetten eingestellt hat.

Hier zeigte sich, dass die Kartensuche ein hilfreiches Instrument ist, um die Suche in entfernten Orten zu ermöglichen.

### **Anwendungsunterstützung und Erwartungskonformität**

Auch die Anwendungsunterstützung und Erwartungskonformität ist bei diesem Arbeitspaket mit über 4 Punkten bewertet worden. Einzige Kritikpunkte waren, dass in dem Suchkontext Dublin teilweise POIs angezeigt wurden, die nicht unmittelbar interessant genug erschienen. Dazu zählten die Stadtteile von Dublin. Hier muss man sagen, dass Geonames als auch DBpedia regional unterschiedliche Qualitäten hinsichtlich der Datenmenge und Vielfalt bietet. Unabhängig des Beispiels Dublin, werden in entlegenen Gegenden kaum oder nur unzureichende Informationen durch die Kartensuche gefunden.

Für weitere Arbeiten sollte die Anwendung einen entsprechenden Hinweis liefern, falls in dem angezeigten Kartenausschnitt wenige oder keine Ergebnisse vorhanden sind, dass der Nutzer den Kartenzoom manuell erhöhen sollte. Weiterhin kann man sich vorstellen, dass das System eigens den Suchradius erhöht.

### **Detailfragen**

Mit 4.53 Punkten wurde die erste Detailfrage bewertet, in der es um die Suche nach Events geht. Hier sieht man, dass es interessant sein kann, gezielt nach einem Event (in diesem Beispiel Shakira) zu suchen. Die folgende Frage nach Browsen in entfernte Orte wurde mit respektablen 4.64 Punkten und einer geringen Standardabweichung von 0.28 Punkten bewertet. Die Frage wurde deutlich höher als alle anderen Fragen bewertet und bildet somit im Rahmen dieser Evaluation das beste Ergebnis. Das zeigt, dass es durchaus für den Anwender interessant ist, mit seinem mobilen Endgerät auch unabhängig des eigenen Standortes in entfernten Orten Informationen zu konsumieren.

#### **9.4.6 Angemessenheit der Arbeitsschritte**

Bei dieser und den nächsten drei Fragen wurde ein statistisches Mittel aus allen Antworten genommen, bei dem die absoluten Häufigkeiten jeder Antwort

<b>Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.</b>	<b>Häufigkeit</b>	<b>Prozent</b>	<b>Kumulierte Prozenze</b>
(1) Trifft nicht zu	0	0	0
(2) Trifft kaum zu	3	0.04	0.04
(3) Trifft teilweise zu	10	0.13	0.17
(4) Trifft zu	41	0.55	0.72
(5) Trifft voll zu	21	0.28	1
Gesamt	75	1	
<b>Statistische Maßzahlen</b>			
Mittelwert	4.07		
Standardabweichung	0.75		

Tabelle 8: Kennzahlen: Angemessenheit der Arbeitsschritte

addiert und in Relation zu allen möglichen Antworten gesetzt sind. Tabelle 8 zeigt die Skala 1 bis 5 mit den absoluten Häufigkeiten der abgegebenen Antworten. Insgesamt konnten auf die fünf Arbeitspakete 75 Punkte vergeben werden. Somit konnte eine Gesamttendenz in Bezug auf die Usability errechnet werden.

Die erste Frage "Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen" soll die allgemeine Angemessenheit der Durchführung der Arbeitsschritte ausdrücken. Also wie zweckmäßig sind einzelne Arbeitsschritte und wie verhält sich der Arbeitsaufwand in Relation zu den Ergebnissen. Für die Analyse der Ergebnisse werden zwei statistisch signifikante Maßzahlen verwendet: den Mittelwert und die Standardabweichung. Die Durchführung der einzelnen Arbeitsschritte empfanden die Teilnehmer größtenteils als angemessen. Im Mittelwert sind dies 4.07 Punkte, wobei das bei 55 % der Teilnehmer zutraf. Hier sieht man, dass die Anwendung bei der Durchführung der Aufgaben keine allzu großen unnötigen Arbeitsschritte bedarf. Lediglich 17 % sahen das nur teilweise oder kaum. Keine Person konnte in den 5 Arbeitspaketen feststellen, dass diese völlig unangemessen waren.

#### 9.4.7 Intuition

Bei der Frage "Die Durchführung der einzelnen Arbeitsschritte war intuitiv" geht es in erster Linie darum, ob die Interaktionsabläufe so gestaltet sind, dass sich ein Anwender ohne Probleme darin zurecht findet. Auch hier existiert mit 55 % ein starkes Mittelfeld, in dem die Teilnehmer die einzelnen Arbeitsschritte als intuitiv empfunden haben. Lediglich zwei Punkte entfal-

<b>Die Durchführung der einzelnen Arbeitsschritte war intuitiv.</b>	Häufigkeit	Prozent	Kumulierte Prozen-te
(1) Trifft nicht zu	2	0.03	0.03
(2) Trifft kaum zu	5	0.07	0.10
(3) Trifft teilweise zu	9	0.12	0.22
(4) Trifft zu	41	0.55	0.77
(5) Trifft voll zu	18	0.24	1
Gesamt	75	1	
<b>Statistische Maßzahlen</b>			
Mittelwert	3.93		
Standardabweichung	0.93		

Tabelle 9: Kennzahlen: Intuition

len auf den schlechtesten Wert. Im Gegensatz zur ersten Frage, ist Streuung um den Mittelwert mit 0.93 Punkten etwas höher.

Die Evaluation der Designstudie hat einen großen Anteil daran, dass viele die Anwendung als intuitiv bedienbar sehen. Basierend auf den Erkenntnissen der ersten Studie konnte somit vor der Implementation sicher gestellt werden, dass zumindest im Vorfeld keine unnötigen und schlechten Designentscheidungen getroffen werden. Weiterhin tragen der iterative Entwurf und das ständige Benutzerfeedback, der in Abschnitt 8.1 ab Seite 74 diskutiert wird, einen gewissen Anteil daran.

#### 9.4.8 Anwendungsunterstützung

Die signifikanten Eigenschaften bei der Anwendungsunterstützung zur Durchführung der Aufgabe sind: hinreichende Hinweise, etwa in Form von Dialogen, visuelle Unterstützungen beim Auswählen von Facetten oder etwa die Vorschau von Ergebnissen bei der Suche. Im Gegensatz zu den ersten beiden Fragen empfanden nur 48 % der Teilnehmer, dass dies zu trifft. Bei 25 % traf dies teilweise, kaum oder nicht zu. Entscheidend ist aber, dass das Gesamtsystem mit 3.98 Punkten ordentlich bewertet wurde. Die Streuung um den Mittelwert mit 0.9 Punkten befindet sich zudem in einem vergleichbaren Rahmen. 28 % der Teilnehmer sahen weiterhin, dass die Anwendungsunterstützung in einem vollen Umfang zugetroffen hat.

<b>Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.</b>	<b>Häufigkeit</b>	<b>Prozent</b>	<b>Kumulierte Prozente</b>
(1) Trifft nicht zu	2	0.03	0.03
(2) Trifft kaum zu	2	0.03	0.06
(3) Trifft teilweise zu	14	0.19	0.25
(4) Trifft zu	36	0.48	0.73
(5) Trifft voll zu	21	0.28	1
Gesamt	75	1	
<b>Statistische Maßzahlen</b>			
Mittelwert	3.98		
Standardabweichung	0.9		

Tabelle 10: Kennzahlen: Anwendungsunterstützung

<b>Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.</b>	<b>Häufigkeit</b>	<b>Prozent</b>	<b>Kumulierte Prozente</b>
(1) Trifft nicht zu	2	0.03	0.03
(2) Trifft kaum zu	5	0.07	0.10
(3) Trifft teilweise zu	18	0.24	0.34
(4) Trifft zu	31	0.41	0.75
(5) Trifft voll zu	19	0.25	1
Gesamt	75	1	
<b>Statistische Maßzahlen</b>			
Mittelwert	3.78		
Standardabweichung	0.98		

Tabelle 11: Kennzahlen: Erwartungskonformität

### 9.4.9 Erwartungskonformität

Die Erwartungskonformität mit der Frage "Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen" zielt darauf ab, inwiefern die Teilnehmer über die Interaktions-Eingaben überrascht oder genau das Verhalten des Systems erwartet haben. Tendenziell ist diese Frage insgesamt etwas schlechter bewertet worden. 24% der Teilnehmer waren über etwaige Interaktionen überrascht oder haben diese nicht erwartet. 41% empfanden die Reaktionen des Systems als gut und immerhin ein Viertel der Teilnehmer wurden in ihrer Erwartungshaltung nicht negativ überrascht. Der Mittelwert liegt etwas unter den vorherigen Fragen und ist mit 3.78 Punkten und der Standardabweichung dennoch im guten Mittelfeld.

### 9.4.10 Weitere Anmerkungen

Tabelle 12 zeigt den letzten Block des Fragenkatalogs, in dem allgemeine Fragen zu der Anwendung gestellt wurden. Unabhängig einer Aufgabe mussten die Teilnehmer die Anwendung hinsichtlich der Usability bewerten. Im Folgenden eine Übersicht über diskussionswürdige Punkte:

- Die Meldungen des MobileFacets Clients waren für viele Anwender sofort verständlich und wurde mit 4.13 Punkten bewertet. Es wurde bei der Entwicklung darauf Wert gelegt, die Anzahl der Dialoge auf ein für den Anwender verständliches Minimum zu legen. Eine eindeutige, identifizierbare Symbolik soll unnötige Dialoge und Hinweise ersetzen um somit den Arbeitsfluss während der Anwendung erleichtern. Das wurde erreicht, in dem beispielsweise ein Mapping der verfügbaren Facetten an entsprechende Icons stattgefunden hat. Weiterhin trägt die iterative Implementierung des Prototypen einen maßgeblichen Anteil daran, dass unnötige Dialoge und Namensgebungen<sup>60</sup> entfallen sind.
- Ein intuitives Arbeiten mit der Anwendung war eines der Hauptaspekte bei der Gestaltung und Implementierung dieser. Insgesamt wurde die Anwendung mit 3.54 Punkten und einer geringen Standardabweichung von 0.28 Punkten bewertet. In Anbetracht dessen, dass es sich um einen Prototyp und kein fertiges Produkt handelt, stellen diese Werte durchaus eine gute Ausgangslage dar. Man muss hier erwähnen, dass für alle der Umgang mit Facetten auf einem mobilen Endgerät neu ist, da es zu dieser Zeit wenig vergleichbare Anwendungen gibt.

---

<sup>60</sup>Beispielscreenshot 20 auf Seite 75: Tabeinträge mit Namen, statt Icons

	Mittelwert	Standardabweichung
Die Meldungen der Anwendung waren für mich immer sofort verständlich.	4.13	0.32
Die Anwendung erfordert keine überflüssigen Eingaben.	4.13	0.36
Die Anwendung lässt sich intuitiv bedienen.	3.54	0.28
Die Gestaltung der Anwendung ist einheitlich gehalten.	3.93	0.3
Die Anwendung enthält alle für die Aufgaben benötigten Funktionen.	4.14	0.28
Ich finde dass der erforderliche Aufwand für die Aufgabenergebnisse jeweils angemessen ist.	3.93	0.26
Ich musste mich mit Umwegen oder Tricks behelfen um die geforderten Ergebnissen so zu erzielen wie ich sie haben möchte.	2.54	0.49
Die Informationen, die zur Erledigung der Aufgaben notwendig sind, sind auf dem Bildschirm übersichtlich verfügbar.	3.26	0.26
Werden Sie vor Aktionen, die nicht rückgängig gemacht werden können, von der Anwendung hinreichend gewarnt?	3.45	0.51
Macht das Programm manchmal etwas, ohne dass ich es zu diesem Zeitpunkt wollte?	2.0	0.44
Lassen sich die Funktionen die zur Bewältigung der Aufgaben dienen, einfach finden?	3.53	0.22
Waren die Wartezeiten bei den Suchvorgängen zu lange?	3.43	0.48
Ich war mir bei den Wartezeiten immer sicher, dass die Anwendung weiterarbeitet.	3.53	0.51
Ich war manchmal überrascht, wie die Anwendung auf meine Eingaben reagiert.	2.17	0.41
Arbeitet die Anwendung während der Bedienung immer stabil und zuverlässig?	3.79	0.55

Tabelle 12: Kennzahlen: Weitere Anmerkungen

- Der Umgang mit Facetten empfanden die meisten Teilnehmer anfangs etwas ungewöhnlich, jedoch nach einer gewissen Einarbeitszeit wurden die Interaktionen zunehmend intuitiver. Eine Suchfunktion innerhalb der angezeigten Facetten, sowie eine sinnvolle Sortierung der Instanzen kann hier eine Verbesserung in Bezug auf notwendige Arbeitsschritte liefern. Gerade die Suche innerhalb der *Landmark*-Facette war für viele Teilnehmer mühselig, da dort sehr viele Instanzen untergeordnet sind.
- Die Facettenqualität ist ein Hauptkritikpunkt der Teilnehmer. Die Herz-Jesu Kirche beispielsweise, ist unter der Facette *Landmark* unterordnet, welches nicht intuitiv genug ist. Eine Subfacette *Churches in Germany* wäre in diesem Suchkontext sicherlich sinnvoller. Das MobileFacets System bedient sich an externe Datenquellen und nimmt keine Vorfilterung bezüglich der vorhandenen Facetten vor. DBpedia liefert durch die OWL-Klassifizierung der Ressourcen schon eine vergleichsweise gute Qualität der Daten. Jedoch sieht man an diesem Beispiel, dass es Verbesserungsbedarf hinsichtlich der Facettenextraktion geben muss. Hier müsste man ein System zur Facettenerkennung<sup>61</sup> auf beliebigen Datenquellen hernehmen, um dem Anwender im Anschluss das Browsen auf diesen zu erleichtern.
- Während der Evaluation kam es vor, dass die angebundenen Webservices nicht hundertprozentig verfügbar waren. Das hatte zur Folge, dass die Server-Abfragen nicht funktioniert haben und somit keine Ergebnisse gezeigt wurden. Die MobileFacets Architektur baut auf eine massive Anbindung an externen Datenquellen und ist somit auf die Verfügbarkeit dieser angewiesen. Gerade die API von Geonames machte teilweise starke Probleme. Hier müsste man sich in Zukunft überlegen, ob die Anbindung an Geonames vollständig durch Abfragen aus der DBpedia ersetzt werden kann, da auch DBpedia Ressourcen semantisch mit einer Geokoordinate verknüpft sind. Die Designentscheidung beim Architektur-Entwurf baute auf die Anbindung an Geonames, da äquivalente SPARQL-Abfragen des Dbpedia-Endpunktes tendenziell langsamer von statten liefen.
- Die Frage "Waren die Wartezeiten bei den Suchvorgängen zu lange?", welche mit 3.43 Punkten bewertet wurde, zeigt, dass dies noch verbesserungswürdig ist. Teilweise waren die Wartezeiten bei der "What's around me?" Suche für die Teilnehmer zu lange. Als Nutzer ist man es gewohnt, dass Ergebnisse innerhalb von zehn bis spätestens 15

---

<sup>61</sup>Siehe TripleRank: Ranking Semantic Web Data By Tensor Decomposition [24]

Sekunden auf den Bildschirm erscheinen. Alles was länger dauert, wird als störend empfunden und hat einen negativen Einfluss auf die zukünftige Verwendung einer Anwendung [18]. Somit besteht in der Server-Komponente Verbesserungsbedarf hinsichtlich der Verarbeitung der Abfragen.

- Die What's around me? Funktion wurde sehr gut aufgenommen. Jedoch war für viele nicht klar, dass nur im angezeigten Kartenausschnitt gesucht wird. Hier müsste man dem Anwender einen zusätzlichen Hinweis, beispielsweise durch eine Hilfefunktion, bereitstellen.
- Die einheitliche Gestaltung der Anwendung wurde mit guten 3.93 Punkten bewertet.

## 10 Schlussfolgerung

Die Ergebnisse dieser Evaluation zeigen, dass der gesamte Entwicklungsprozess von den initialen Designentwürfen bis hin zu Implementierung eines lauffähigen Prototypen dazu geführt hat, dass es durchaus Möglich und Bequem sein kann, auf einem mobilen Endgerät mittels des facettierten Browsens nach semantischen Informationen zu suchen. Einzelne kritische Punkte lassen sich zumindest soweit verbessern, so dass das MobileFacets System durchaus hinreichende Funktionalitäten für den Alltag mitbringt. In folgenden werden noch einmal die wichtigsten Kritikpunkte erläutert und es wird darauf eingegangen, inwiefern sich der gesamte Entwicklungsprozess auf die Qualität des Systems niedergeschlagen hat.

Das Suchen von Informationen mithilfe von Facetten ist im Desktopbereich üblich. Man nehme sich nur die Beispiele Amazon und Ebay vor, in dem der Nutzer gezielt durch Auswählen von Facetten die Ergebnismenge nach seinen Bedürfnissen gestalten kann. Im Zuge dieser Evaluation zeigte sich, dass es für manche eher ungewöhnlich war, dieses Paradigma der Facettennavigation auf einem mobilen Endgerät zu sehen. Das kann daran liegen, dass diese Art der Navigation (mittels hierarchisch angeordneten Listen) im Desktopbereich eher untypisch ist. Hier stellt man sich die Frage, inwiefern sich eine komplexe Navigation mithilfe von Facetten auf einem mobilen Endgerät realisieren lässt. Diese Arbeit zeigt, dass es schwierig ist, identische Navigationspfade (wie z.B. expandierende Listen oder umfangreiche Filterfunktionen bei Ebay) auf einem mobilen Endgerät abzubilden.

Anhand der Evaluations-Ergebnisse zeigt sich dennoch, dass es nach einer kurzen Eingewöhnungszeit für viele Teilnehmer kein Problem war, mit dieser Art von Navigation und Selektion der Facetten umzugehen. Kritikpunkte, wie beispielsweise dass die Instanzen nicht alphabetisch sortiert waren, oder dass der Suchvorgang zu lange gedauert hat, müssen in weiteren Implementationsarbeiten bereinigt werden. Weiterhin sollte eine Suchfunktion vorhanden sein, die es ermöglicht, innerhalb von Facetten- oder Instanzlisten nach bestimmten Begriffen zu suchen.

Zusammenfassend kann man zudem sagen, dass insbesondere die Facettenqualität für die Usability des MobileFacets System eine tragende Rolle spielt. Dieses System bedient sich ausschließlich an externen Datenquellen und ist darauf angewiesen, dass diese Informationsräume eine hinreichende Klassifikation der Daten besitzt. Denn eine effiziente Navigation durch einen Informationsraum kann nur realisiert werden, falls die Facetten Teile des Informationsraumes eindeutig beschreiben. Das ist bei den angebundenen Datenquellen nur teilweise der Fall. Die Dbpedia Datensätzen ist in dieser Hinsicht noch verbesserungswürdig, wenn man sich das Beispiel

der Herz-Jesu Kirche in der Facette *Landmark* vor Augen hält. Ohne eine Verbesserung des Hintergrundwissens der Daten kann dieser Umstand schwer oder gar nicht gelöst werden.

In Zuge der Facetten-Extraktion aus solchen semantischen Datenquellen muss man sich die Frage stellen, inwiefern die Facettenqualität verbessert werden kann. Das MobileFacets System bietet eine eher statische Ansicht auf die verfügbaren Facetten. Der Anspruch ist aber, eine generisches und dynamisches Retrieval von solchen Facetten auf semantischen Datenquellen zu realisieren. Im besten Falle sollte die Anwendung willkürliche Datensätze anbinden können, ohne dass sie ein spezielles Wissen über die Struktur der Daten mit sich bringt. Hier bietet sich beispielsweise TripleRank an, mit dem sich RDF-Daten anhand eines RDF Graphen nach ihrer Relevanz anordnen lassen. Es wird dabei eine gewichtete Liste von relevanten Facetten errechnet, wobei jede Facette einen speziellen Informationsaspekt repräsentiert. Zu jeder Facette ist eine Liste von Ressourcen gelistet, die semantisch relevant mit dieser in Verbindung steht [23].

Was hat die erste Designstudie im Rahmen dieser Arbeit gebracht? Die erste Designstudie trägt einen signifikanten Anteil an der Implementierung und Realisierung dieses Projektes. Die Thematik des facettierten Browsens ist nicht neu. Aber gerade im mobilen Bereich sind die Anforderungen hinsichtlich des Entwurfs einer Facetten-basierten Benutzeroberfläche um ein Vielfaches komplexer. Es mussten neue Ansätze bei der Gestaltung der Oberfläche gefunden werden, da Bildschirmplatz und Interaktion mit eben dieser Oberfläche sich von den Desktopumgebungen abheben. Es mussten nachvollziehbare Navigationsstrukturen und flexible Möglichkeiten entworfen werden, um ein effizientes und benutzerfreundliches Browsen über Facetten zu ermöglichen. Durch die Designstudie konnten viele unnütze Vorgänge im Voraus vermieden - und sinnvolle Designentscheidungen getroffen werden.

## 11 Zusammenfassung und Ausblick

Diese Diplomarbeit präsentiert eine Designstudie, die daraus resultieren Anforderung, die Implementierung und die abschließende Evaluation des MobileFacets Systems. Das System bietet aus Sicht eines Anwenders unter anderem folgenden Möglichkeiten:

- Die Suche nach Facetten mithilfe eines mobilen Endgeräts. Dabei werden Facetten und ihre zugehörigen Instanzen in Abhängigkeit des Ortes gesucht und gefunden.
- Es lassen sich Informationen in entfernten Orten Suchen und Browsen.

Realisiert wird dies durch eine Suchfunktion oder über die Suche mit Hilfe der eingebauten Karte.

- Facetten lassen sich flexibel und ohne großen Interaktionsaufwand selektieren, löschen und einsehen.
- Es werden für den Benutzer interessante Orte und Gegebenheiten gefunden. Darüber hinaus bietet das System die Möglichkeit, Personen, Organisationen wie Firmen, Bands oder sonstige zu finden, die semantisch in Relation mit dem jeweiligen Ort stehen.
- Es lassen sich Informationen zu Events in der Umgebung finden.
- Die eingebaute Fotoansicht zeigt nah gelegene oder gesuchte Fotos an.
- Die Detailansicht zu Instanzen bieten ausführliche Informationen für Anwender.

Der Client-Prototyp als Android Packet (APK)<sup>62</sup> sowie der Prototyp des Servers als Tomcat Web Archive (WAR)<sup>63</sup> lassen sich herunterladen. Zu den Binär-Paketen ist zusätzlich der Quellcode des Clients<sup>64</sup> und des Servers<sup>65</sup> als Download verfügbar.

Das MobileFacets System kann für zukünftige Arbeiten mit weiteren semantischen Repositories erweitert werden. Zu den möglichen Datenquellen zählen unter anderem FreeBase<sup>66</sup> und DBtune<sup>67</sup>. Letzteres stellt umfangreiche und strukturierte Musik-Relevante Informationen bereit, die dazu genutzt werden können, um zum Beispiel semantische Zusammenhänge zwischen Musiker aus DBpedia-Ressourcen aufzubauen. Über den semantischen Datenquellen hinaus, lassen sich weitere Webservices einbinden, solange sie genügend strukturierte Metainformationen bereitstellen, die für eine sinnvolle Facetten-Navigation nützlich sind. Weiterhin muss untersucht werden, inwiefern das System bei vielen Benutzern skaliert. Dazu kann man sich Tests in realen Bedingungen vorstellen, in dem Nutzer die Anwendung im alltäglichen Gebrauch verwenden.

Ein generisches Konzept für das facettierte Browsen auf semantischen oder anderen strukturierten Daten kann realisiert werden, wenn es seitens der Datenanbieter eine Schnittstelle gibt, die umfangreiche Informationen

---

<sup>62</sup><http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Client.apk>

<sup>63</sup><http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Server.war>

<sup>64</sup><http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Client.tar.gz>

<sup>65</sup><http://csxpoi.isweb.uni-koblenz.de:8081/MobileFacets-Server.tar.gz>

<sup>66</sup><http://www.freebase.com/>

<sup>67</sup><http://dbtune.org/>

über die unterliegende Datenstruktur liefert. Idealerweise können hier die verfügbaren Facetten aufgelistet sein, so dass eine aufwändige Analyse zur Gewinnung der Facetten wegfällt.

## Literatur

- [1] Hearst, M.A.: UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In: in the Workshop on Computer Interaction and Information Retrieval, HCIR 2008. (2008)
- [2] Hearst, M.A.: Design recommendations for hierarchical faceted search interfaces. In: Proc. SIGIR 2006, Workshop on Faceted Search. (August 2006) 26–30
- [3] Stoica, E., Hearst, M.A., Richardson, M.: Automating Creation of Hierarchical Faceted Metadata Structures. In Sidner, C.L., Schultz, T., Stone, M., Zhai, C., eds.: HLT-NAACL, The Association for Computational Linguistics (2007) 244–251
- [4] Franz, T., Koch, J., Dividino, R., Staab, S.: LENA-TR : Browsing Linked Open Data Along Knowledge-Aspects. In: Linked AI: AAAI Spring Symposium Linked Data Meets Artificial Intelligence". (2010)
- [5] Schenk, S., Saathoff, C., Staab, S., Scherp, A.: SemaPlover - Interactive semantic exploration of data and media based on a federated cloud infrastructure. *J. Web Sem.* **7**(4) (2009) 298–304
- [6] Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. In: The Semantic Web - ISWC 2006. (November 2006) 272–285
- [7] Gartner, I.: Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond
- [8] Steiniger, S., Neun, M., Edwardes, A.: Foundations of Location Based Services
- [9] Sohn, T., Li, K.A., Griswold, W.G., Hollan, J.D.: A diary study of mobile information needs. In Czerwinski, M., Lund, A.M., Tan, D.S., eds.: CHI, ACM (2008) 433–442
- [10] : Faceted DBLP (03 2010) <http://dblp.l3s.de>.
- [11] Sacco, G.: Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience. Springer, Berlin (2009)
- [12] Vickery, B.C.: Knowledge Representation: A Brief Review. In: Journal of Documentation. Volume 42(3). (1986) 145–159

- [13] Lindsay, P.H., Norman, D.A.: Human Information Processing. Academic press, New York (1977)
- [14] Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006. (2006)
- [15] Karlson, A.K., Robertson, G.G., Robbins, D.C.C., Czerwinski, M.P., Smith, G.R.: FaThumb: a facet-based interface for mobile search. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 711–720
- [16] Tunkelang, D.: Faceted Search. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers (2009)
- [17] Aristotle: Historia Animalium. (343 BC)
- [18] Arndt, H.: Integrierte Informationsarchitektur: Die erfolgreiche Konzeption professioneller Websites. Springer, Berlin (2006)
- [19] Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5**(2) (1993) 199–220
- [20] Ranganathan., S.R.: Elements of library classification., Bombay: Asia Publishing House. (1962)
- [21] Yee, K.P., Swearingen, K., Li, K., Hearst, M.: Faceted metadata for image search and browsing. In: Human factors in computing systems, New York, NY, USA, ACM (2003) 401–408
- [22] Pietriga, E., Bizer, C., Karger, D., Lee, R.: Fresnel: A Browser-Independent Presentation Vocabulary for RDF. In: The Semantic Web - ISWC 2006. Volume 4273 of Lecture Notes in Computer Science., Springer-Verlag (2006) 158–171
- [23] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen. Springer, Berlin (2008)
- [24] Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: Ranking Semantic Web Data By Tensor Decomposition. In: International Semantic Web Conference (ISWC). (10 2009)
- [25] Hearst, M.A., Elliott, A., English, J., Sinha, R.R., Swearingen, K., Yee, K.P.: Finding the flow in web site search. Commun. ACM **45**(9) (2002) 42–49

- [26] Becker, C., Bizer, C.: DBpedia Mobile: A Location-Enabled Linked Data Browser, 1st Workshop about Linked Data on the Web (2008)
- [27] Wilson, M.L., Russell, A., Smith, D.A., Owens, A., schraefel, m.c.: mSpace Mobile: A Mobile Application for the Semantic Web, End User Semantic Web Workshop, ISWC2005 (2005)
- [28] Azuma, R.: A Survey of Augmented Reality. *Presence* **6**(4) (1997) 355–385
- [29] van Aart, C.J., Wielinga, B.J., van Hage, W.R.: Mobile Cultural Heritage Guide: Location-Aware Semantic Search. In Cimiano, P., Pinto, H.S., eds.: EKAW. Volume 6317 of Lecture Notes in Computer Science., Springer (2010) 257–271
- [30] Gibbins, N., Harris, S., schraefel, m.: Applying mSpace Interfaces to the Semantic Web. *World Wide Web Conference* (2004)
- [31] Wilson, M.L., Smith, D.A., Russell, A., schraefel, m.c.: mSpace Mobile: a UI Gestalt to Support On-the-Go Info-Interaction, the SIGCHI Conference on Human Factors in Computing Systems (CHI06) (2006)
- [32] usability.de: Faceted Search - Die neue Suche im Usability-Test (06 2010)
- [33] Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search, 13th International Conference on Business Information Systems (BIS 2010) (05 2010)
- [34] English, J., Hearst, M., Sinha, R., Swearingen, K., Yee, K.P.: Flexible Search and Navigation Using Faceted Metadata. Technical report, University of Berkeley (2002)
- [35] Neumann, F.: Formative vs. summative Evaluation (06 2002)
- [36] Gediga, G., Hamborg, K.C.: Evaluation of Software Systems. *Encyclopedia of Computer Science and Technology* **45** (2001)
- [37] Jeffries, R., Miller, J.R., Wharton, C., Uyeda, K.: User interface evaluation in the real world: a comparison of four techniques. In: CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (1991) 119–124
- [38] Schneidermeier, T.: Usability of Mobile Devices. *Untersuchungen zur Gebrauchstauglichkeit mobiler Endgeräte am Beispiel Apple iPhone* (September 2009)

- [39] Lewis, C., Rieman, J., Blustein, A.J.: Task-Centered User Interface Design: A Practical Introduction (1993)
- [40] Nielsen, J.: Usability Heuristics. In: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
- [41] Sommerville, I.: Software engineering (5th ed.). Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA (1995)
- [42] Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley (2003) With contributions from Rice, D., Foemmel, M., Hieatt, E., Mee, R. and Stafford, R.
- [43] Geonames.org: About Geonames <http://www.geonames.org/about.html>.
- [44] : Great Circle (11 2010) <http://mathworld.wolfram.com/GreatCircle.html>.
- [45] Biron, P.V., Malhotra, A.: XML Schema Part 2: Datatypes Second Edition (2004) <http://www.w3.org/TR/xmlschema-2/>.
- [46] Murata, M., Laurent, S.S., Kohn, D.: XML Media Types (2001) <http://www.ietf.org/rfc/rfc3023.txt>.
- [47] Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. 2 edn. Object Tecnology. Addison Wesley Longman, Massachusetts (April 1999)
- [48] Schmeiß, D.: Erweiterung der csxPOI-Anwendung um Event-basierte Points of Interests. Master's thesis, Universität Koblenz-Landau (01 2010)
- [49] Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: WWW '07: Proceedings of the 16th International World Wide Web Conference, Banff, Canada. (2007) 697–706
- [50] Gediga, G., Hamborg, K.C.: IsoMetrics: An usability inventory supporting summative and formative evaluation of software systems. In Bullinger, H.J., Ziegler, J., eds.: HCI (1), Lawrence Erlbaum (1999) 1018–1022

## Abbildungsverzeichnis

1	Hierarchische Taxonomie . . . . .	11
2	FaThumb (Screenshot) . . . . .	17
3	Links: iPhone Applikation <i>Mobile cultural heritage guide: location aware semantic search</i> [29], Rechts: <i>mSpace Mobile: A Mobile Application for the Semantic Web</i> [27] . . . . .	19
4	Start der Applikation (Mockup) . . . . .	22
5	Browsen über Facetten (Mockup) . . . . .	23
6	Szenario: POI-Suche, Anwendungsstart (Mockup) . . . . .	25
7	Variante 1: Browsen über Facetten (Mockup) . . . . .	26
8	Darstellung der ausgewählte Facetten (Mockup) . . . . .	27
9	Kartensicht mit POIs (Mockup) . . . . .	27
10	Darstellung der Suchfunktion (Mockup) . . . . .	29
11	Szenario Trivia: Darstellung der ausgewählten Facetten und das Browsen über Facetten (Mockup) . . . . .	29
12	Szenario Trivia: Ergebnisliste und Detailansicht (Mockup) . . . . .	30
13	Variante 2: Browsen über Facetten (Mockup) . . . . .	32
14	Variante 3: Browsen über Facetten (Mockup) . . . . .	33
15	Darstellung der ausgewählten Facetten (Mockup) . . . . .	33
16	Varianten um ausgewählte Facetten darzustellen . . . . .	50
17	Architektur . . . . .	59
18	Haversine Berechnung . . . . .	65
19	Erster lauffähiger Oberflächenprototyp (Client Screenshots) . . . . .	74
20	Zweiter lauffähiger Oberflächenprototyp (Client Screenshots) . . . . .	75
21	Dritter lauffähiger Oberflächenprototyp (Client Screenshots) . . . . .	76
22	High-Level Facetten, Subfacetten und Instanz-Ebene (Client Screenshots) . . . . .	79
23	Freie Textsuche (Client Screenshots) . . . . .	81
24	Karte mit POIs, Karte mit Events, Karte mit POIs und Events (Client Screenshots) . . . . .	84
25	Menü der Karte (Client Screenshot) . . . . .	85
26	Liste mit Personen, Liste mit Orten (Client Screenshots) . . . . .	86
27	Fotoansicht Kartensuche, Fotoansicht freie Textsuche (Client Screenshots) . . . . .	88
28	Personen Details, Event Details (Client Screenshots) . . . . .	89
29	Repository Einstellungen (Client Screenshots) . . . . .	91
30	Server Sequenzdiagramm . . . . .	94
31	OWL Klassifikation einer DBpedia Ressource . . . . .	102

## Quellcodeverzeichnis

1	Beispielontologie (OWL) . . . . .	12
2	What's around me? REST Anfrage . . . . .	68
3	MobileFacets Serverantwort (DBpedia XML) . . . . .	72
4	MobileFacets Serverantwort (Event XML) . . . . .	72
5	DBpedia Personen/Organisation-Suche (SPARQL-Abfrage) . . . . .	93
6	DBpedia Ortssuche (SPARQL-Abfrage) . . . . .	96
7	DBpedia Schlüsselwort-Suche (SPARQL-Abfrage) . . . . .	96
8	Geonames-Abfrage (URL) . . . . .	98
9	DBpedia-Detailabfrage (SPARQL) . . . . .	101
10	Eventful RESTful-Umkreissuche (URL) . . . . .	103
11	Eventful RESTful-Schlüsselwortsuche (URL) . . . . .	103
12	Upcoming RESTful-Schlüsselwortsuche (URL) . . . . .	104
13	Flickr-RESTful Umkreissuche (URL) . . . . .	105
14	Flickr-RESTful freie Suche (URL) . . . . .	105

## Tabellenverzeichnis

1	Kategorien der Informationsbedürfnisse . . . . .	4
2	REST Parameter und Definitionsbereiche . . . . .	70
3	Kennzahlen: Aufgabe A: POI-Browsen . . . . .	113
4	Kennzahlen: Aufgabe B: Personen Suche . . . . .	115
5	Kennzahlen: Aufgabe C: POI-Suche . . . . .	117
6	Kennzahlen: Aufgabe D: Event-Browsen . . . . .	120
7	Kennzahlen: Aufgabe E: Event-Suche und weiteres Browsen . . . . .	122
8	Kennzahlen: Angemessenheit der Arbeitsschritte . . . . .	124
9	Kennzahlen: Intuition . . . . .	125
10	Kennzahlen: Anwendungsunterstützung . . . . .	126
11	Kennzahlen: Erwartungskonformität . . . . .	126
12	Kennzahlen: Weitere Anmerkungen . . . . .	128

# Anhang

## Fragenkatalog der Designstudie

Nr.	Allgemeine Fragen
1.1	Alter
1.2	Geschlecht
1.3	Wie gut kennen Sie sich allgemein mit mobilen Endgeräten aus?
1.4	Besitzen Sie ein mobiles Endgerät, wenn ja wie viele?
1.5	Welches Endgerät besitzen Sie?
1.6	Wie oft am Tag benutzen Sie Ihr Gerät? (in Minuten)
1.7	Ich weiss über die neuesten Handyentwicklungen im Bezug auf Technik bescheid
1.8	Mit welchen Interaktionsformen haben Sie bereits Erfahrung?

Nr.	Fragen zur POI-Suche
2.1	Könnten Sie die einzelnen Schritte in Bezug auf das Suchen nach Points-of-Interests nachvollziehen?
2.2	Beurteilen Sie: Ist die Navigation über das virtuelle Menü nachvollziehbar?
2.3	Finden Sie im allgemeinen die Navigationsstruktur der Applikation nachvollziehbar?
2.4	Das Browsen über Facetten wird mit einzelnen Listen realisiert. Können Sie diese Navigationsform nachvollziehen?
2.5	Die Kartenansicht zeigt gefundene Ressourcen als kleine Hinweissymbole mit jeweiligem Titel. Könnte die Übersicht verloren gehen, falls sehr viele Ressourcen gefunden werden?
2.6	Beim Anklicken einer Ressource auf der Kartenansicht öffnet sich ein Dialog um Details anzuzeigen oder nicht. Würden Sie eine direkt Detailansicht der Ressource präferieren? (ohne Umweg über einen Dialog)
2.7	Ist Ihnen ersichtlich, dass Facetten und Ressourcen nur in der umliegenden Umgebung gesucht werden?
2.8	Bei dem virtuellen Menü befinden sich kleine Symbole neben dem Text. Finden Sie Verwendung von Symbolen im allgemeinen hilfreich?

<b>Nr.</b>	<b>Fragen zum Szenario 1: Sightseeing</b>
2.9	Ist das gezeigte Szenario „Sightseeing“ realistisch und authentisch?
2.10	Haben Sie selbst schon einmal eine ähnliche Situation gehabt?

<b>Nr.</b>	<b>Fragen zur Suche nach ortsunabhängigen Informationen</b>
3.1	Konnten Sie die einzelnen Schritte zur Suche nach Informationen zu Fernsehtürmen nachvollziehen?
3.2	Finden Sie im allgemeinen die Navigationsstruktur zur Suche nach ortsunabhängigen Informationen angemessen?
3.3	Im Gegensatz zum Szenario 1 wurde hier mit einer Tableiste anstatt eines Menüs gearbeitet. Beurteilen Sie: Vereinfacht die Verwendung von Tableisten die Navigation?
3.4	Erachten Sie eine Live-Vorschau bei der Suche im allgemeinen für eine hilfreiche Funktion?
3.5	In Schritt 8 werden gefundene Facetten angezeigt. Haben Sie hier eher eine Ergebnisliste erwartet?
3.6	Der Location Kontext wird in den Schritten 11-12 über eine Suche von Berlin nach Koblenz geändert. Finden Sie diese Vorgehensweise kompliziert?
3.7	Im Tab „Facets“ wird neben den Facetten noch die zuvor ausgewählten Facetten am oberen Bildschirmrand angezeigt. Halten Sie das für sinnvoll?
3.8	Wünschen Sie sich eine Historie der zuletzt gesuchten Begriffe?

<b>Nr.</b>	<b>Fragen zum Szenario 1: Social Lunch</b>
3.9	Ist das gezeigte Szenario „Social Lunch“ realistisch und authentisch?
3.10	Haben Sie selbst schon einmal eine ähnliche Situation gehabt?

<b>Nr.</b>	<b>Verwendung von Tabs</b>
4.1	Bevorzugen Sie allgemein eine Applikation mit Tableisten? (1.1,1.3)
4.2	Bewahren Tableisten ihre Meinung nach die Übersicht einer Applikation?
4.3	Tableisten verkleinern zum Beispiel die Kartenansicht. Empfinden Sie das als einen Nachteil?
4.4	In Beispiel 1.3 könnte man noch das virtuelle Menü ausklappen. Die Tableiste wäre somit nicht mehr sichtbar. Würden Sie das als Störend empfinden?
4.5	Bevorzugen Sie im allgemeinen eine Tableiste am oberen oder unterem Bildschirmrand?
4.6	Welche der folgenden Funktionen sollten per Tabs erreichbar sein? Kreuzen Sie jeweils ja oder nein an!

<b>Nr.</b>	<b>Darstellung und Verständnis von Facetten</b>
5.1	Ist es hilfreich, wenn die zuvor ausgewählten Facetten als fester Bestandteil auf dem Bildschirm einzusehen sind? (2.1 , 2.3)
5.2	Bei den Beispielen 2.2, 2.4 und 2.5 lassen sich die Facetten Places, Populated Places und Building jeweils unabhängig voneinander abwählen(bzw. löschen). Ist es ersichtlich, dass Populated Places und Building Subfacetten von Places sind?
5.3	Was denken Sie wie die Ergebnismenge von Beispiel 2.5 aussieht? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.4	Was denken Sie was passiert, wenn man bei Beispiel 2.5 die Facette Populated Places deaktiviert? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.5	Beispiel 2.1 und 2.3 zeigen die Facetten in einer hierarchischen Ansicht. Was denken Sie passiert, wenn man den Knopf neben der Facette Building anklickt? Kreuzen Sie das nach Ihrem Verständnis Richtige an!
5.6	Anhand der gezeigten Beispiele: Für welche Darstellung von Facetten würde Sie sich entscheiden?

<b>Nr.</b>	<b>Browsen über Facetten</b>
6.1	In Beispiel 3.1 werden Facetten und ihre zugehörigen Subfacetten farblich getrennt. Beurteilen Sie: Ist eine solche farbliche Trennung hilfreich für die Navigation?
6.2	Finden Sie die räumliche Aufteilung des Bildschirms ansprechend? (3.2)
6.3	Applikation 3.2 wird über einen Stift bedient. Könnten Sie sich vorstellen eine solche Applikation mittels eines Touchscreens zu bedienen?
6.4	Finden Sie die Listendarstellung der verfügbaren Facetten als eine angemessene Darstellungsform? (3.3)
6.5	In Beispiel 3.3 werden Facetten und ihre zugehörigen Subfacetten über Listen realisiert die den kompletten Bildschirm einnehmen. Könnte man hier schnell die Übersicht verlieren?
6.6	Finden Sie ein Dropdown-Menüs als eine angemessene Darstellungsform? (3.4)
6.7	Könnte man bei Beispiel 3.5 schnell den Überblick verlieren?
6.8	Beurteilen Sie: Ist die Schriftgröße in der Listenansicht bei Beispiel 3.5 in Ordnung?
6.9	Anhand der gezeigten Beispiele: Für welche Form des „Browsen über Facetten“ würden Sie sich entscheiden? Kreuzen Sie an!

<b>Nr.</b>	<b>Funktionsumfang der Applikation</b>
7.1	Wünschen Sie sich eine Kartenansicht um Informationen anzeigen zu lassen?
7.2	Wünschen Sie sich eine Ergebnisliste der gefundenen Informationen?
7.3	Wünschen Sie sich eine Navigationsfunktion?
7.4	Wünschen Sie sich eine Augmented Reality View? (Siehe Mobile Cultural Heritage)
7.5	Wünschen Sie sich eine Historie der zuletzt gesuchten Begriffe?
7.6	Wünschen Sie sich eine Favoritenverwaltung?

## Fragenkatalog zur Evaluation von MobileFacets

Nr.	POI-Browsen
1.1	Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.
1.2	Die Durchführung der einzelnen Arbeitsschritte war intuitiv.
1.3	Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
1.4	Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
1.5	Das sofortige Suchen nach umliegenden Informationen bei Anwendungsstart erachte ich als sinnvoll.
1.6	Der Browse & Search Reiter enthält alle notwendigen Informationen die für das Browsen und Auswählen von Facetten relevant sind.
1.7	Das Verfeinern der Suchmenge über die Facetten (wie im Beispiel: Anzeige der Bahnhöfe) ist für mich nachvollziehbar.
1.8	Auf der Kartenansicht finde ich alle Informationen die ich für diesen Suchkontext benötige.
1.9	Photos werden auf der Karte angezeigt. Ich finde diese Funktion sinnvoll für diesen Suchkontext.
1.10	Die von der Anwendung gelieferten Suchergebnisse entsprachen meinen Erwartungen.

Nr.	Personen-Suche
2.1	Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.
2.2	Die Durchführung der einzelnen Arbeitsschritte war intuitiv.
2.3	Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
2.4	Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
2.5	Die Suche nach Personen, die etwas mit der Stadt oder der Umgebung zu tun haben, erachte ich als eine nützliches Feature.
2.6	Die Detailansicht (z.B. von Bodo Illgner) enthält für die Aufgabe alle notwendigen Informationen.
2.7	Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.

Nr.	POI-Suche
3.1	Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.
3.2	Die Durchführung der einzelnen Arbeitsschritte war intuitiv.
3.3	Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
3.4	Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
3.5	Die Suchfunktion lässt sich über das virtuelle Menü im "Browse & SearchReiter erreichen. Ich empfinde diese Vorgehensweise als sinnvoll.
3.6	Die Live-Vorschau der Suchergebnisse empfinde ich als eine hilfreiche Funktion.
3.7	Die Vorschläge der Live-Vorschau entsprechen meinen Erwartungen.
3.8	Der Wikipedia-Abstract ist umfangreich genug um die Aufgabe zu bearbeiten.
3.9	Die von der Anwendung gelieferten Suchergebnisse entsprachen meinen Erwartungen.
3.10	Die angezeigten Photos in der Photoansicht entsprechen meinen Erwartungen.

Nr.	Event-Browsen
4.1	Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.
4.2	Die Durchführung der einzelnen Arbeitsschritte war intuitiv.
4.3	Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
4.4	Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
4.5	Den Suchkontext von Koblenz nach Berlin zu wechseln empfinde ich als einen mühseligen Vorgang. (über Karte oder Suchfunktion)
4.6	Ich vermisse eine Funktion, die mir bei Eingabe eines Ortes direkt diesen auf der Karte anzeigt.
4.7	Die Suche auf bestimmte Daten (Wikipedia/Flickr/Events) einzuschränken empfinde ich als sinnvolle Funktion.
4.8	Die Einschränkung der zu suchenden Daten lässt sich in den Preferences festlegen. Ich empfinde diese Vorgehensweise als sinnvoll.
4.9	Events werden in Abhängigkeit des Start-Datums durch unterschiedlich farbige Sterne markiert. Die Darstellungsweise empfinde ich als sinnvoll.
4.10	Die Detailansicht von Events enthält alle nötigen Informationen die ich benötige.
4.11	Events werden in in Subfacetten unterteilt (Musik, Social, Sport,...). Diese Einteilung in Subfacetten empfinde ich als sinnvoll.
4.12	Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.

Nr.	Event-Suche und weiteres Browsen
5.1	Ich empfinde die Anzahl der erforderlichen Arbeitsschritte als angemessen.
5.2	Die Durchführung der einzelnen Arbeitsschritte war intuitiv.
5.3	Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
5.4	Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
5.5	Die Suche nach Events erachte ich als eine sinnvolle Funktion.
5.6	Das Browsen nach Informationen in entfernten Orten (Dublin) erachte ich als eine sinnvolle Funktion.
5.7	Die von der Anwendung gelieferten Ergebnisse entsprachen meinen Erwartungen.

Nr.	Weitere Fragen
6.1	Die Meldungen der Anwendung waren für mich immer sofort verständlich.
6.2	Die Anwendung erfordert keine überflüssigen Eingaben.
6.3	Die Anwendung lässt sich intuitiv bedienen.
6.4	Die Gestaltung der Anwendung ist einheitlich gehalten.
6.5	Die Anwendung enthält alle für die Aufgaben benötigten Funktionen.
6.6	Ich finde dass der erforderliche Aufwand für die Aufgabenergebnisse jeweils angemessen ist.
6.7	Ich musste mich mit Umwegen oder Tricks behelfen um die geforderten Ergebnissen so zu erzielen wie ich sie haben möchte.
6.8	Die Informationen, die zur Erledigung der Aufgaben notwendig sind, sind auf dem Bildschirm übersichtlich verfügbar.
6.9	Werden Sie vor Aktionen, die nicht rückgängig gemacht werden können, von der Anwendung hinreichend gewarnt?
6.10	Macht das Programm manchmal etwas, ohne dass ich es zu diesem Zeitpunkt wollte?
6.11	Lassen sich die Funktionen die zur Bewältigung der Aufgaben dienen, einfach finden?
6.12	Waren die Wartezeiten bei den Suchvorgängen zu lange?
6.13	Ich war mir bei den Wartezeiten immer sicher, dass die Anwendung weiterarbeitet.
6.14	Ich war manchmal überrascht, wie die Anwendung auf meine Eingaben reagiert.
6.15	Arbeitet die Anwendung während der Bedienung immer stabil und zuverlässig?

## CD-Inhalt

- Designstudie
  - Mockup-Designentwürfe
  - Evaluation der Designstudie
  - Auswertung der Designstudie
- Architekturdiagramme
- Iterativer Implementierung
  - Fragebögen
- Präsentation des Zwischenstands
- Evaluation des MobileFacets Prototyps
  - Folien des Vortrags
  - Auswertungen
- Android Client Quellcode
- Tomcat Server Quellcode