

Summative Evaluation facettierter Suche und Exploration auf mobilen Endgeräten

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Informatik

vorgelegt von

Mark Schneider

Erstgutachter: Prof. Dr. Steffen Staab
Institute for Web Science and Technologies, Fachbereich 4: Informatik

Zweitgutachter: Jun.-Prof. Dr. Ansgar Scherp
Institute for Web Science and Technologies, Fachbereich 4: Informatik

Koblenz, im November 2011

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig 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 14. November 2011

Mark Schneider

Abstract. Large and unknown data sets can be easily and systematically discovered by using faceted search. If implementing applications for smartphones, it needs to be considered that unlike desktop applications you can only use smaller screen sizes and there are limited possibilities for interaction between user and smartphone. These limitations can negatively influence the usability of an application. With FaThumb and MobileFacets, two mobile applications exist, which implement and use faceted search, although only MobileFacets is designed for current smartphones with touchscreen. However, FaThumb provides a novel facet navigation, which is newly realized in MFacets for present smartphones within this work. Moreover, this work deals with the performance of a summative evaluation between both applications, MFacets and MobileFacets, with regards to usability and presents the evaluated results.

Kurzzusammenfassung. Mittels facetrierter Suche lassen sich große, unbekannte Datensätze einfach und gezielt erkunden. Bei der Implementation von Anwendungen für Smartphones ist zu beachten, dass im Gegensatz zu Desktop-Anwendungen ein kleinerer Bildschirm und nur beschränkte Möglichkeiten zur Interaktion zwischen Benutzer und Smartphone zur Verfügung stehen. Diese Beschränkungen können die Benutzbarkeit einer Anwendung negativ beeinflussen. Mit FaThumb und MobileFacets existieren zwei mobile Anwendungen, die die facettrierte Suche umsetzen und verwenden, aber nur MobileFacets ist für gegenwärtige Smartphones mit Touchscreenbildschirm ausgelegt. Jedoch bietet FaThumb eine neuartige Facettenavigation, die durch MFacets in dieser Arbeit für aktuelle Smartphones neu realisiert wird. Außerdem befasst sich diese Arbeit mit der Durchführung einer summarischen Evaluation zwischen den beiden Anwendungen, MFacets und MobileFacets, bezüglich der Benutzbarkeit und präsentiert die ausgewerteten Ergebnisse.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	2
2	Verwandte Arbeiten	5
2.1	Taxonomien und Facetten	5
2.1.1	Taxonomie	5
2.1.2	Dynamische Taxonomie	7
2.1.3	Facettierte Taxonomie	8
2.1.4	Realisierung in MobileFacets und MFacets	10
2.2	Facettierte Suche	11
2.2.1	Prinzipien und Funktionalitäten	12
2.2.2	Bestandteile der facettierte Suche	14
3	Anforderungen und Analyse	19
3.1	FaThumb	19
3.2	MobileFacets	22
3.3	Anforderungen für MFacets	26
3.3.1	Übereinstimmungen mit FaThumb und MobileFacets ..	27
3.3.2	Abgrenzung von FaThumb und MobileFacets	27
4	MFacets	29
4.1	Architektur	30
4.2	Client	31
4.2.1	Datenmodell	31
4.2.2	Benutzeroberfläche	32
4.2.3	Programmsteuerung	41
4.3	Server	43
4.3.1	Datenrepositories	44
4.3.2	Webservice zur Facettenstruktur	46
4.3.3	Webservice zur Bereitstellung der Ressourcen	48

4.4	Vorbereitung der Anwendungen auf die Evaluation	49
4.4.1	Änderungen an den Anwendungen	49
4.4.2	Erweiterungen an den Anwendungen	52
5	Summative Evaluation	55
5.1	Konzeption der Evaluation	56
5.1.1	Aufgaben	56
5.1.2	Kommentare	59
5.2	Durchführung	60
5.3	Auswertung und Interpretation der Messergebnisse	61
5.3.1	Beschreibung der Probanden und des Datensatzes	61
5.3.2	Auswertung der Effektivität	63
5.3.3	Auswertung der Effizienz	64
5.3.4	Auswertung der Zufriedenheit	67
5.3.5	Schlussfolgerungen	70
6	Fazit, Ausblick und Danksagung	75
	Literaturverzeichnis	77
	Anhang - Evaluation	81
A.1	Einführung MFacets	81
A.2	Einführung MobileFacets	84
A.3	Aufgaben	87
A.4	Aussagen	90
A.5	Logstruktur für MFacets	93
A.6	Logstruktur für MobileFacets	96

Abbildungsverzeichnis

2.1	Ausschnitt und Vereinfachung der Klassifikation der Lebewesen nach Aristoteles [Tun09].	6
2.2	Dynamische Taxonomie mit angedeuteter Reduktion [ST09].	7
2.3	Facettierte Taxonomie (vereinfacht) [ST09].	10
2.4	Taxonomie für MobileFacets und MFacets	11
2.5	Camelis [Fer08].	15
2.6	FacetLens [LSR ⁺ 09].	16
3.1	Hauptansicht und Navigation in FaThumb [KG06].	20
3.2	Navigation und Auswahl in FaThumb [KG06].	21
3.3	Filterregion und Ergebnisliste in FaThumb [KG06].	22
3.4	Facetten- und Karten-Tab in MobileFacets [Kle10].	23
3.5	Ergebnislisten- und Galerie-Tab in MobileFacets [Kle10].	24
3.6	Auswahl einer Facette im Facetten- und Ergebnislisten-Tab in MobileFacets [Kle10].	26
4.1	Architektur des MFacets-Systems mit Unterstützung von MobileFacets.	29
4.2	Verteilungsdiagramm der Architektur des MFacets-Systems.	30
4.3	Erste Version an Mockups (1).	33
4.4	Erste Version an Mockups (2).	34
4.5	Letzte Version an Mockups (1).	35
4.6	Letzte Version an Mockups (2).	36
4.7	Screenshots von MFacets (1).	39
4.8	Screenshots von MFacets (2).	41
4.9	Sequenzdiagramm der Kommunikation zwischen Client, Server und den Datenrepositories.	44
4.10	Änderungen in MFacets	51
5.1	Kontinuierliche Likert-Skala zur Bewertung der Aussagen in der Evaluation.	59

5.2	Erfahrung mit und Nutzung des Smartphones der Probanden...	62
5.3	Boxplot zur benötigten Bearbeitungszeit der Aufgaben der Evaluation.	64
5.4	Boxplot zur benötigten Klickzahl der Aufgaben der Evaluation.	66

Tabellenverzeichnis

3.1	Vergleich zwischen FaThumb, MFacets und MobileFacets.	28
5.1	Auswertung der Aufgaben der Evaluation nach Zeit in Sekunden.	65
5.2	Auswertung der Aufgaben der Evaluation nach Klicks.	67
5.3	Auswertung der Aussagen der Evaluation.	68

Einleitung

1.1 Motivation

Der stetige Wachstum der Verkaufszahlen an Smartphones bedeutet eine immer größer werdende Verbreitung der Nutzung des mobilen Internets [BIT11a, BIT11b, Gar11, Kur11]. Im Gegensatz zur Nutzung des Internets an einem PC müssen Smartphones bei der Darstellung von Inhalten einen kleineren Bildschirm und eingeschränkte Interaktionsmöglichkeiten berücksichtigen. Deshalb genügt es nicht, Desktop-Anwendungen auf Smartphones einfach nur kleiner darzustellen, sondern man muss besonders die Navigation durch die Anwendungen anpassen. Hier ist in erster Linie zu beachten, dass aktuelle Smartphones, wie z.B. das Apple iPhone 4S, Samsung Galaxy S II oder HTC Desire HD, in der Regel einen Touchscreen zur Eingabe besitzen, welcher bei PCs, ausgenommen Tablet-PCs, nicht als Standard gilt.

Der Einsatz des Internets geschieht bei Smartphones nicht wie beim PC eher über den Browser, sondern immer häufiger über Applikationen, kurz Apps. Diese Apps zeichnen sich dadurch aus, dass Inhalte für den Benutzer auf Smartphone-Bildschirme optimiert und gegebenenfalls um spezielle, für das Handiumfeld nützliche Funktionen erweitert werden. Da das Handy in der heutigen Gesellschaft ein ständiger Begleiter ist und auch zunehmend GPS-fähige Smartphones angeboten werden, bieten sich vor allem ortsbezogene Dienste als Apps an [SNE06].

Ein Beispielszenario für einen ortsbezogenen Dienst lässt sich an Hand von *Points-of-Interests* (POI) beschreiben. Wenn man z.B. eine fremde Stadt besucht, könnte es den Benutzer interessieren, welche Sehenswürdigkeiten, Geschäfte oder Restaurants in seiner näheren Umgebung liegen. In diesem Fall ist es vorzuziehen, dem Benutzer nicht jegliche POIs anzuzeigen, sondern diese in Form von Kategorien zu sortieren und anzubieten. Somit können für den Benutzer uninteressante Kategorien schnell ausgeblendet werden und er kann sein erwünschtes Ziel schneller finden. Eine solche Suche und Navigation nach Informationen in Kategorien bzw. Facetten nennt man auch *facettierte Suche* [HEE⁺02, Hea06, Hea08, OR06].

Mobile Anwendungen zur facettierten Suche sind keine Neuigkeit. In [KG06] wird eine Anwendung namens FaThumb mit neuartiger Navigation zur facettierten Suche vorgestellt. Allerdings verwendet FaThumb einen geschlossenen Datensatz und die Facettenavigation erfolgt auf Grund der Zeit der Entwicklung über die Nummerntasten des Smartphones. mSpace Mobile [WRS⁺05] unterstützt das mobile Internet und semantische Daten und beschreibt die facettierte Suche an Hand von verschiedenen Fenstern, die unterschiedliche Informationen beinhalten. Ein ausgewähltes Fenster wird hier zwar zum besseren Lesen vergrößert, dennoch wirkt die Aufteilung und Anordnung der einzelnen Fenster auf Smartphone-Bildschirmen unübersichtlich. Weiter läuft mSpace Mobile auf Personal Digital Assistants (PDAs) und benötigt daher einen Stift zur Eingabe auf dem Touchscreen. MobileFacets [Kle10] und Mobile Cultural Heritage Guide [vAWvH10] setzen auf Listen zur Darstellung verschiedener Facetten, so dass die Navigation sich auf das Auswählen von und Scrollen nach Informationen beschränkt. All diese Variationen zur facettierten Suche auf mobilen Endgeräten haben wegen ihrer Eigenheiten Vor- und Nachteile, was sich direkt auf die Benutzbarkeit der Anwendung auswirkt.

1.2 Zielsetzung

Diese Diplomarbeit verfolgt zwei Ziele. Das erste Ziel ist es, die Idee von FaThumb aufzugreifen und in einer neuen Anwendung, MFacets, auf aktuelle Bedürfnisse und Smartphones anzupassen. Dazu gehört zum einen die Anbindung an öffentlich zugängliche, semantischen Daten über das mobile Internet und zum anderen die Umsetzung der Benutzeroberfläche auf ein Smartphone mit Touchscreenunterstützung. Da MobileFacets für diese Diplomarbeit zur Verfügung steht, werden zur Erleichterung der Implementation die Architektur des Gesamtsystems mit Client, Server und Datenrepositories und einige Funktionalitäten der Anwendung übernommen. In MobileFacets dienen DBpedia¹, GeoNames², Eventful³, Upcoming⁴ und Flickr⁵ als Datenrepositories, welche für MFacets ebenfalls übernommen und um Qype⁶ erweitert werden. Diese Quellen werden auf dem Server aufbereitet, so dass sich die Größe an zu ladenden Daten für das Smartphone und die Bearbeitungszeit der Anfragen reduzieren. Zusätzlich zur facettierten Suche wird MFacets wie MobileFacets sowohl eine Kartenansicht zur Darstellung von POIs und eine Galerie zum Browsen von Bildern enthalten.

Das zweite Ziel dieser Diplomarbeit ist die Durchführung einer summarischen Evaluation [GH99] zu den beiden Applikationen MFacets und

¹ siehe <http://dbpedia.org/About>

² siehe <http://www.geonames.org/>

³ siehe <http://eventful.com/>

⁴ siehe <http://upcoming.yahoo.com/>

⁵ siehe <http://www.flickr.com/>

⁶ siehe <http://www.qype.com/>

MobileFacets bezüglich der Benutzbarkeit. Die Benutzbarkeit einer Anwendung lässt sich an Hand der Effektivität, Effizienz und Zufriedenheit der Anwendung bemessen. Zum Messen dieser Eigenschaften werden die Probanden der Evaluation in zwei Gruppen eingeteilt, in welcher sie wohl ausgewählte und spezifizierte Aufgaben mit einer der beiden Anwendungen bearbeiten und bestimmte Aussagen im Anschluss auf einer kontinuierlichen Likert-Skala bewerten. Auf diese Art und Weise erhält man von MFacets und MobileFacets zwei voneinander unabhängige Messergebnisse, welche sich auswerten und vergleichen lassen. Die gewonnenen Erkenntnisse aus der Auswertung und die Interpretation dieser werden in dieser Diplomarbeit veröffentlicht. Um diese Zielsetzung zu erfüllen, gliedert sich diese Diplomarbeit wie folgt:

In Kapitel 2 werden verwandte Arbeiten zur facettierten Suche diskutiert. Hier wird zum einen das Klassifikationsschema für die facettierte Suche in Form von Taxonomien vorgestellt und zum anderen werden generelle Prinzipien, Funktionalitäten und Bestandteile einer facettierten Suche an Hand von Beispielen beschrieben. Im darauffolgenden Kapitel 3 wird eine genauere Blick auf die mobilen Anwendungen FaThumb und MobileFacets geworfen, um die Anforderungen für MFacets zu definieren. Das gesamte MFacets-System wird dann in Kapitel 4 in seinen Einzelheiten vorgestellt bevor in Kapitel 5 die summative Evaluation zwischen MFacets und MobileFacets bezüglich der Benutzbarkeit präsentiert wird. Kapitel 6 wird diese Diplomarbeit mit einem Fazit, Ausblick und Danksagungen abschließen.

Verwandte Arbeiten

In diesem Kapitel wird der Bezug von dieser Diplomarbeit zu verwandten Arbeiten hergestellt. Zuerst wird die Klassifikation von Daten an Hand von Taxonomien und Facetten näher erläutert, welche zum Teil in den später betrachteten mobilen Anwendungen verwendet werden. Daraufhin wird ein Überblick über die verschiedenen Aspekte der Umsetzung der facettierten Suche geschaffen.

2.1 Taxonomien und Facetten

Um facettierte Suche umzusetzen oder zu verwenden, müssen durchsuchbare Daten nach einer gewissen Struktur geordnet sein. Diese benötigte Ordnung folgt einem Klassifikationsschema, so dass es nahelegt eine entsprechende Datenstruktur bei der Implementation der facettierten Suche einzusetzen. Eine solche Strukturierung von Daten kann auf verschiedene Arten umgesetzt werden, z.B. in Taxonomien, RDF oder Ontologien. Diese Diplomarbeit wird näher auf Taxonomien eingehen, so dass für Anwendungen auf Basis von RDF [BLCC⁺06] oder Ontologien [Ss08] auf die entsprechende Literatur verwiesen wird. Da die ursprüngliche Taxonomie für eine facettierte Suche nicht optimal ausgelegt ist, werden in den folgenden Abschnitten zusätzlich weitere Entwicklungen wie die dynamische Taxonomie und die facettierte Taxonomie erläutert. Anschließend wird gezeigt, welche Variation in den evaluierten mobilen Anwendungen Verwendung findet.

2.1.1 Taxonomie

Das Wort *Taxonomie* lässt sich von den griechischen Worten *táxis*, zu deutsch *Ordnung*, und *nômos*, zu deutsch *Gesetz*, ableiten. Demnach wird in einer Taxonomie eine Ordnung geschaffen, eine Ordnung jeglicher Sachen wie z.B. für Länder oder für Autos. Eine triviale Ordnung wäre eine nach dem Alphabet, was aber wenig hilfreich ist, wenn man z.B. nach einem Nachbarland oder

nach einem Auto einer gewissen Geräteklasse verschiedener Hersteller sucht. Die Ordnung in Taxonomien wird an Hand von Taxa hergestellt, unter Taxa versteht man in der Regel verschiedene Klassen. Daher kann man eine Taxonomie auch als ein Klassifikationsschema bezeichnen [Tun09]. Diese Klassen werden hierarchisch angeordnet, wie in einer Baumstruktur, so dass Klassen beliebig viele Unterklassen, aber nur genau eine Oberklasse, mit Ausnahme der obersten Klasse (auch Wurzelknoten), besitzen. Somit existiert immer genau ein Pfad von einer Klasse zum Wurzelknoten. Eine Ober- und eine Unterklasse stehen immer in einer *is-a*-Beziehung zueinander. Die zu ordnenden Sachen oder Daten werden dann jeweils genau einer Klasse zugeordnet. Diesen Vorgang der Zuordnung bezeichnet man auch als Klassifizieren.

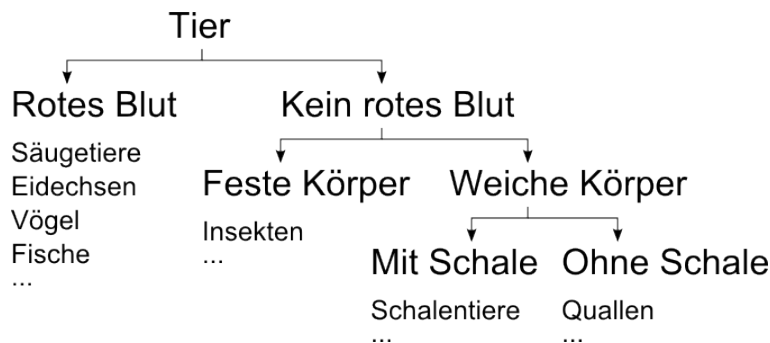


Abb. 2.1. Ausschnitt und Vereinfachung der Klassifikation der Lebewesen nach Aristoteles [Tun09].

Eine solche Ordnung einer Taxonomie ist erstmals von Aristoteles (384 - 322 v. Chr.) auf die Klassifikation von Lebewesen angewandt worden. In Abb. 2.1 sieht man einen Ausschnitt aus dieser Taxonomie. Man erkennt hier klar die strikt hierarchische Struktur, da jede Klasse maximal einer Oberklasse zugehörig ist. Außerdem sieht man, dass die Lebewesen jeweils genau einer Klasse zugeordnet sind. Da es sich bei dem Ausschnitt um eine Vereinfachung der eigentlichen Taxonomie handelt, wird nicht unbedingt ganz klar, in welcher Ebene der Taxonomie die Lebewesen den Klassen zugeordnet werden. Man könnte von einer Beschränkung der Klassifizierung nur auf die untersten Klassen (auch Blattknoten) ausgehen. In der Tat ist dies allerdings nicht der Fall, so dass Daten jeglichen Klassen zugeordnet werden können.

Diese Eigenschaften einer Taxonomie sorgen nicht nur für eine übersichtliche Ordnung nach diversen Kriterien, sondern ziehen auch ein paar Nachteile mit sich. Die Zuordnung eines Datums zu genau einem Knoten hat zur Folge, dass dieses nur über genau einen Pfad zu erreichen ist. Zum Beispiel liegt Deutschland in Europa und ist Mitglied in der Europäischen Union. Hat man nun eine Taxonomie mit zwei solchen Knoten, wie sollte man Deutschland nun zuordnen? Außerdem hat eine größere Anzahl an Daten auch eine größere und

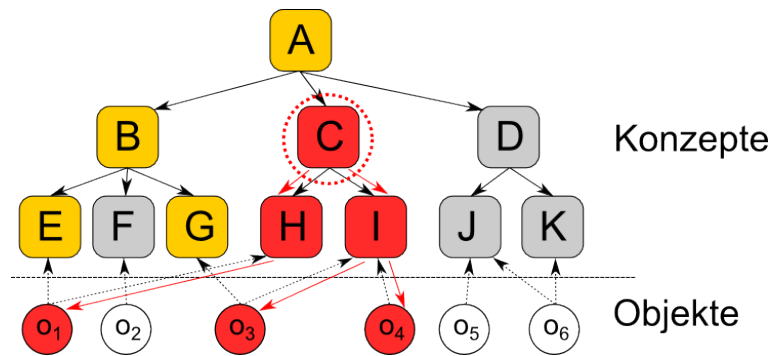


Abb. 2.2. Dynamische Taxonomie mit angedeuteter Reduktion [ST09].

komplexere Taxonomie zur Folge, welche je nach Suche nicht immer komplett benötigt wird. Eine Lösung zu diesen Problemen wird im nächsten Abschnitt aufgezeigt.

2.1.2 Dynamische Taxonomie

Dynamische Taxonomien unterscheiden sich von gewöhnlichen Taxonomien hauptsächlich in einem Punkt, Daten können mehreren Klassen zugeordnet werden [ST09]. Aus diesem Grund werden dynamische Taxonomien auch als multidimensionale Taxonomien bezeichnet. Dieser Unterschied führt zu bestimmten Vorteilen in der Verwendung von dynamischen Taxonomien.

Ein trivialer Vorteil ist, dass die Zuordnung eines Datums nicht mehr auf eine Klasse beschränkt ist. Somit können Daten, die mehrere, verschiedene Informationen beinhalten, auch verschiedenen Klassen zugeordnet werden, so dass die Entscheidung auf eine Klasse wegfällt.

Ein weiterer Vorteil ist der, dass durch die zusätzlich möglichen Zuordnungen mehr Informationen in einer Taxonomie vorhanden sind. Diese weiteren Informationen können verwendet werden, um die Daten zu einer bestimmten Klasse weiter zu bearbeiten, z.B. zu reduzieren durch Filter.

In Abb. 2.2 werden die verschiedenen Eigenschaften und Möglichkeiten von dynamischen Taxonomien illustriert. Was zuvor als Klasse bzw. Datum bezeichnet worden ist, wird nun abstrahiert zu Konzept bzw. Objekt. Weiter erkennt man in der Abbildung, dass es sich um eine multidimensionale Taxonomie handelt, da die Objekte o_1 , o_2 und o_3 jeweils zwei Konzepten zugeordnet sind. Um die zusätzlichen Informationen durch die Multidimensionalität ausnutzen zu können, muss die Taxonomie näher betrachtet werden.

Dynamische Taxonomien lassen sich reduzieren, um einen kompakteren Ausschnitt ohne überflüssige Informationen zu erhalten. Eine solche Reduktion, in [ST09] auch *zoom-in* genannt, benötigt vier Schritte:

Die Reduktion wird immer von einem bestimmten Konzept ausgehend vorgenommen. Dazu wird der *Fokus* auf ein Konzept gelegt. Im Beispiel liegt der Fokus auf Konzept C .

Im nächsten Schritt werden die dem fokussiertem Konzept zugehörigen Objekte ermittelt, die *interessanten Objekte*. Hierzu werden die Objekte gesucht, die entweder direkt dem Konzept oder einem seiner Kinder zugeordnet sind. Im Beispiel ist C kein Objekt zugeordnet, dafür hingegen o_1 zu H und o_3 und o_4 zu I . Die restlichen Objekte werden nun nicht mehr beobachtet.

Daraufhin werden die *verwandten Konzepte* identifiziert. Dies sind die Konzepte, denen die interessanten Objekte zugeordnet sind inklusive deren Eltern. Im Beispiel sind dies von den Objekten ausgehend in erster Ebene E , G , H und I , danach B und C und schließlich A .

Im letzten Schritt wird die *Reduktion* vollzogen und alle nicht-verwandten Konzepte und nicht-interessanten Objekte werden ausgeblendet.

Nun hat man eine kompaktere Taxonomie mit der es sich mit Bezug auf Konzept C einfacher arbeiten lässt. Wie zuvor erwähnt können auf Grund der Multidimensionalität die Objekte zu einem Konzept an Hand weiterer Konzepte gefiltert werden. Welche Filter überhaupt möglich sind, lässt sich am Beispiel gut erkennen. Die trivialen Filter bilden die Kinder des im Fokus stehenden Konzepts, also H und I . Weitere mögliche Filter bilden die restlichen verwandten Konzepte E , G , B und A . Somit wird nur noch die reduzierte Taxonomie benötigt.

Auf diese Art und Weise werden mit dynamischen Taxonomien bestimmte Suchen im Gegensatz zu klassischen Taxonomien einfacher (Reduktion) umgesetzt oder gar erst ermöglicht (Multidimensionalität). Dennoch haben dynamische Taxonomien den Nachteil, dass alle Informationen in einer oftmals großen Taxonomie liegen und erst nach Durchführung einer Reduktion sich die Handhabbarkeit der Taxonomie verbessert. Wie man dieses Problem für facetiierte Suche konzeptionell lösen kann, wird im nächsten Abschnitt gezeigt.

2.1.3 Facettierte Taxonomie

Die Idee der facetiierten Taxonomie ist nicht neu und wurde schon 1933 von Ranganathan in seiner Colon-Klassifikation umgesetzt [Ran65, Tun09]. Ranganathan hat für Büchereien ein Kodierungsschema um Bücher und Artikel zu klassifizieren entwickelt. Eine solche kodierte Klassifikation besteht aus einer Hauptkategorie, kodiert als Großbuchstabe, und einer Kombination zwischen fünf grundlegenden, unabhängigen Taxonomien, welche bei Verwendung als Interpunktionszeichen kodiert werden. Aus den Interpunktionszeichen, speziell dem Doppelpunkt (engl. *colon*), lässt sich auch der Name der Colon-Klassifikation ableiten. Die fünf Taxonomien werden als P-M-E-S-T zusammengefasst und beziehen sich auf: Individualität (**P**ersonality), Material (**M**atter), Energie (**E**nergy), Raum (**S**pace) und Zeit (**T**ime). Die einzelnen Konzepte der Taxonomien werden als Kombination

von Buchstaben und Ziffern kodiert. Um die Colon-Klassifikation zu verdeutlichen, wird folgendes Beispiel¹ verwendet:

Ein Artikel zur *Diagnostik der Lungentuberkulose in Frankreich 1989* wird kodiert als: L,45:421:3.53.N89

In die Einzelteile aufgespalten bedeutet dies:

- Als Hauptkategorie: Medizin (L)
- Aus der P-Taxonomie: Lunge (,45)
- Aus der M-Taxonomie: nichts entnommen, da kein Semikolon ;
- Aus der E-Taxonomie: Tuberkulose (:421) und Diagnostik (:3)
- Aus der S-Taxonomie: Frankreich (.53)
- Aus der T-Taxonomie: 1989 (.N89)

Ranganathan hat zwar für seine Colon-Klassifikation facettierte Taxonomien verwendet, aber diese nicht so benannt. Deshalb werden nun facettierte Taxonomien formal erläutert und die zugehörigen Begrifflichkeiten eingeführt, die auch im weiteren Verlauf dieser Diplomarbeit zum Einsatz kommen.

Wie schon angedeutet verwenden facettierte Taxonomien zur Klassifikation der Objekte im Gegensatz zu dynamischen Taxonomien nicht eine große Taxonomie, sondern mehrere von einander unabhängige Taxonomien [ST09]. Um diese Taxonomien klein und kompakt zu halten, sind diese bestenfalls so konzipiert, dass sie jeweils eigene Domänen abdecken und untereinander keine Schnittmenge besitzen, also orthogonal zu einander stehen. Ranganathans P-M-E-S-T Taxonomien sind ein Beispiel für solche unabhängigen Taxonomien. Weitere Beispiele findet man in speziellen Taxonomien für Veranstaltungen oder für Sportarten.

In Abb. 2.3 wird eine vereinfachte Variante für eine facettierte Taxonomie verdeutlicht. Es wird hier sofort klar, dass nicht mehr eine, sondern mehrere (hier zwei) Taxonomien zur Klassifikation verwendet werden. Die Unabhängigkeit und Orthogonalität der Taxonomien wird durch die separaten Kästen angedeutet. Vom Namen abgeleitet werden die Konzepte der Taxonomien auch Facetten genannt und Kinder als Sub- und Eltern als Superfacetten bezeichnet. Die Eigenschaft der Multidimensionalität lässt sich an Hand der Zuordnung der Objekte o_2 und o_3 zu jeweils zwei Facetten erkennen. Weiter sieht man an Objekt o_3 , dass Objekte jeglichen Facetten in der Hierarchie und auch Facetten verschiedener Taxonomien zugeordnet werden können.

Diese Merkmale sorgen dafür, dass facettierte Taxonomien im Gegensatz zu dynamischen Taxonomien kompakter, übersichtlicher und skalierbarer sind. Ein Nachteil ist jedoch der mögliche zeitliche Mehraufwand bei der Konzeption der einzelnen Taxonomien und die Behandlung des Falles, wenn

¹ Das Beispiel entstammt dem Wikipedia Artikel zur Colon-Klassifikation, siehe <http://de.wikipedia.org/wiki/Colon-Klassifikation>

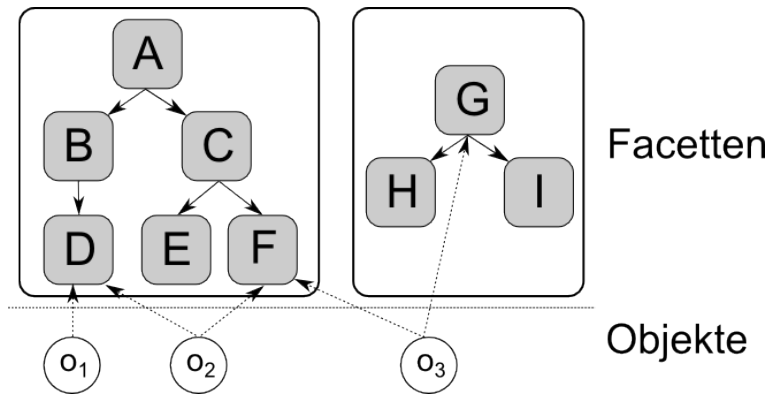


Abb. 2.3. Facettierte Taxonomie (vereinfacht) [ST09].

Taxonomien eine gemeinsame Facette besitzen. Letzteres tritt allerdings erst bei der Implementation einer facettierten Taxonomie ein.

An dieser Stelle sind verschiedene Varianten an Taxonomien für die Verwendung einer facettierten Suche vorgestellt worden. In den betrachteten mobilen Anwendungen MobileFacets und MFacets wird jedoch keine „reine“ Version der erwähnten Taxonomien verwendet, weshalb die genaue Umsetzung in den Anwendungen Gegenstand des nächsten Abschnitts ist.

2.1.4 Realisierung in MobileFacets und MFacets

In den mobilen Anwendungen MobileFacets und MFacets wird zur Umsetzung der facettierten Suche dieselbe Taxonomie zu Grunde gelegt. Diese Taxonomie ist eine Variation aus dynamischer und facettierter Taxonomie, welche in Abb. 2.4 dargestellt ist. Es handelt sich hierbei um eine facettierte Taxonomie mit vier unabhängigen Taxonomien, die über eine fiktive Wurzelfacette zu einer dynamischen Taxonomie verbunden sind.

Die unabhängigen Taxonomien ergeben sich aus den verschiedenen Datenquellen, für die eine facettierte Suche realisiert wird. Wie man aus der Abbildung entnehmen kann, gibt es jeweils eine Taxonomie für Örtlichkeiten (*Place*), Personen (*Person*), Veranstaltungen (*Event*) und Organisationen (*Organisation*). Wie man hier sieht, handelt es sich um verschiedene Domänen, so dass keine Überschneidung untereinander vorliegt. Diese Taxonomien bzw. ihre Facetten werden zur Klassifikation der Daten aus den Datenquellen verwendet. Solche Daten werden im Folgenden wie auch in der Abbildung als Ressourcen bezeichnet. Die Ressourcen können wie erwartet mehreren Facetten zugeordnet werden, allerdings liegen diese Facetten in der Regel in derselben Taxonomie. Nur in einigen Fällen werden Ressourcen Facetten aus *Person* und *Event* zugeordnet, was durch die Herkunft der Daten begründet wird.

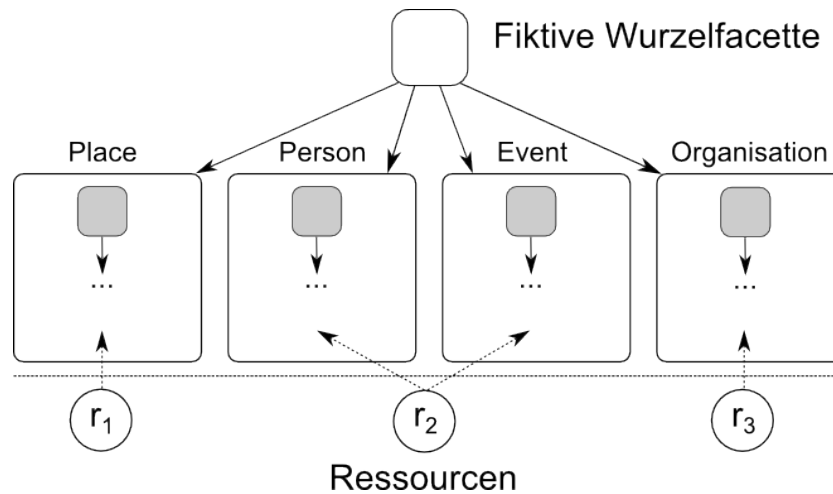


Abb. 2.4. Taxonomie für MobileFacets und MFacets

Das Hinzufügen einer fiktiven Wurzelfacetten lässt sich an Hand der Implementation in den Anwendungen erläutern. Die unabhängigen Taxonomien werden auf Grund ihrer hierarchischen Struktur als Baum implementiert. Wenn in den Anwendungen nun eine Facette näher betrachtet werden möchte, dann werden ihre Subfacetten dargestellt, z.B. werden in erster Ebene die Facetten *Place*, *Person*, *Event* und *Organisation* dargestellt. Um die Implementation der Navigation zu erleichtern wird dieser ersten Ebene eine nullte Ebene vorgeschoben, um einen an Stelle von vier Bäumen einzusetzen. Somit werden in der Anfangsansicht die Facetten der vier Taxonomien dargestellt.

Es sei noch zu erwähnen, dass diese Taxonomie ohne Reduktion implementiert worden ist, da man sich entschlossen hat, nach Fokussierung oder Auswahl einer Facette den Ergebnisraum an Ressourcen durch die Auswahl einer weiteren Facette nicht zu verringern, sondern zu erweitern. Die Ressourcen mehrerer Facetten werden also nicht geschnitten, sondern vereinigt. Welche Vor- und Nachteile ein solcher Ansatz hat und weitere Designmöglichkeiten für eine facettierte Suche werden im nächsten Unterkapitel vorgestellt.

2.2 Facettierte Suche

Unter *facettierter Suche* (auch *facettierte Browsen*) versteht man eine explorative Suche, d.h. ein Datensatz wird über Kategorien bzw. Facetten erkundet. Wie dieser Datensatz strukturiert und aufgebaut werden kann, wurde in 2.1 diskutiert. In diesem Unterkapitel werden verschiedene Varianten der Implementation der facettierten Suche vorgestellt. Es wird gezeigt, welche grundsätzlichen Prinzipien und Funktionalitäten die facettierte Suche umsetzen kann bzw. soll. Weiter wird die facettierte Suche in ihre Einzelteile

aufgespalten und es wird veranschaulicht, welche Designmöglichkeiten für die jeweiligen Bestandteile existieren. Dies wird beispielhaft an Hand von Anwendungen verdeutlicht.

2.2.1 Prinzipien und Funktionalitäten

Wie bereits erwähnt, ermöglicht die facettrierte Suche das Erforschen eines Datensatzes. Damit dieses Erforschen gut umgesetzt wird, gilt es gewisse Prinzipien einzuhalten und verschiedene Funktionalitäten zur Navigation anzubieten [ST09, Fer08, Pol09]. Im Folgenden werden zuerst zwei Prinzipien und danach fünf Funktionalitäten erläutert.

Prinzip der Uneingeschränktheit der Suche bei multidimensionaler Klassifikation

Dieses Prinzip besagt, dass es für eine facettrierte Suche unerheblich sein soll, zu welchem Zeitpunkt während der Suche oder in welcher Reihenfolge verschiedene Facetten ausgewählt werden. Demnach darf für die Verknüpfung von Facetten per boolescher Operatoren die Reihenfolge der Facetten keine Rolle spielen. Dies bewirkt, dass bei gleicher Auswahl von Facetten eine Anwendung immer die gleichen Ressourcen als Ergebnis liefert.

Poka Yoke Prinzip

Mit *Poka Yoke* (japanisch, zu deutsch *unglückliche Fehler vermeiden*) bezeichnet man im Kontext der facettierten Suche das Prinzip, dass keine leeren Ergebnismengen auftreten dürfen. Dieses Prinzip gilt es hauptsächlich bei einer mehrfachen Auswahl an Facetten zu berücksichtigen. Implementiert eine Anwendung die Verknüpfung von Facetten als Schnittmenge der zugeordneten Ressourcen, so kann daraus eine leere Menge resultieren. Um dem Vorzubeugen darf eine solche Anwendung nach Auswahl einer Facette nur noch die Facetten darstellen, die zu keiner leeren Schnittmenge führen. Eine weitere Lösungsmöglichkeit ist das „Ausgrauen“ von Facetten, so dass diese zwar dargestellt, aber nicht mehr ausgewählt werden können. Dieses Prinzip verringert die Möglichkeit der Frustration eines Benutzers, da dadurch „Sackgassen“ in der Erkundung vermieden werden.

Nach diesen Prinzipien werden nun die Funktionalitäten zur Navigation in einer facettierten Suche vorgestellt, die allerdings nicht alle von jeder Anwendung umgesetzt werden.

Zoom-in

Mit *Zoom-in* bezeichnet man den Vorgang der Verfeinerung einer Suche, also entweder die Auswahl einer weiteren Facette oder einer Subfacette. Per Definition sind einer Subfacette weniger bis gleich viele Ressourcen zugeordnet, so dass sich die Ergebnismenge verringert. Dies ist auch bei

einer UND-Verknüpfung mehrerer Facetten der Fall. Demzufolge wird in den Datensatz hineingezoomt.

Zoom-out

Wie der Name es schon andeutet, läuft ein *Zoom-out* entgegen eines *Zoom-ins*. Um die Ergebnismenge zu vergrößern, wird von einer Facette ausgehend eine Superfacette ausgewählt oder bei mehreren Facetten eine Facette abgewählt. Folglich zoomt man im Datensatz heraus.

Shift

In einem *Shift* wird bei der Auswahl mehrerer Facetten eine Facette durch eine andere ersetzt, um eine ähnliche oder verwandte Ergebnismenge zu erhalten. Ein *Shift* kann durch die Kombination von *Zoom-in* und *Zoom-out* bzw. *Zoom-out* und *Zoom-in* erreicht werden. Um dies zu verdeutlichen, kann man sich folgendes Beispiel vorstellen. Man betrachtet die Facetten *deutsch* und *Sportler*. Wenn man nun die Facette *Schauspieler* hinzufügt (*Zoom-in*) und *Sportler* abwählt (*Zoom-out*), so hat man einen *Shift* von *deutschen Sportlern* auf *deutsche Schauspieler* vollzogen. Für die andere Kombination kann man *deutsch* durch *europäisch* (*Zoom-out*) und dann durch *englisch* (*Zoom-in*) ersetzen, um einen *Shift* auf *englische Sportler* durchzuführen.

Pivot

Ein *Pivot* kommt beim Austausch einer kompletten Suchanfrage durch eine verwandte Suchanfrage vor. Man bezeichnet zwei Suchanfragen als verwandt, wenn sie mindestens eine gemeinsame Facette unter den verwandten Facetten besitzen (siehe verwandte Konzepte unter 2.1.2). Um einen solchen Austausch zu vollziehen, geht man beispielhaft wie folgt vor. Geht man wiederum von den Facetten *deutsch* und *Sportler* aus, so erhält man unter den Ergebnissen z.B. Dirk Nowitzki (Basketballspieler in der NBA bei den Dallas Mavericks). Eine verwandte Suchanfrage wäre die nach den Facetten *amerikanisch* und *Basketballspieler*. Diese beiden Suchanfragen haben *Dallas Mavericks* als gemeinsame verwandte Facette, welche man auch *Pivot* nennt.

Querying by Examples

Unter *Querying by Examples* (zu deutsch *Suche an Hand von Beispielen*) versteht man Suchanfragen, die aus einer beliebigen Menge an Ressourcen abgeleitet werden können. Dazu werden ausgewählte Ressourcen und deren Facetten näher betrachtet, so dass aus den gemeinsamen Facetten eine Suchanfrage erstellt werden kann. In der Regel wird hierzu eine beliebige Suchanfrage ausgeführt und daraufhin eine neue Suchanfrage an Hand von bestimmten Ressourcen aus der Ergebnismenge hergeleitet. Zum Beispiel kann man aus der Ergebnismenge zu der Anfrage nach

Sportlern die Ressourcen Sebastian Vettel und Michael Schumacher auswählen, um die Suche nach *Formel 1* und *Weltmeister* zu erhalten.

Dies sind die Funktionalitäten, die von Anwendungen zur Navigation in der facettierten Suche verwendet werden können. Wie diese in Anwendungen implementiert sind und welche Möglichkeiten in der Realisierung existieren, wird im nächsten Abschnitt dargelegt.

2.2.2 Bestandteile der facettierten Suche

Wenn man eine facettierte Suche implementieren möchte, so muss man sich verschiedene Gedanken bezüglich des Designs machen, z.B. wie man Facetten darstellt, wie man zwischen ihnen navigiert oder wie man eine oder mehrere Facetten auswählt. Grundsätzliche Designfragen zur facettierten Suche werden in [Hea06, Hea08] diskutiert. Im Folgenden werden verschiedene Bestandteile einer facettierten Suche vorgestellt und an Hand von Beispielen wird gezeigt, welche Ansätze zur Realisierung verfolgt werden können.

Facettendarstellung und Navigation

Das Hauptaugenmerk jeder facettierten Suche ist die Navigation zwischen den Facetten. Direkt damit verbunden ist auch die Präsentation der Facetten, also wie diese dargestellt und angeordnet sind. Hier werden nun verschiedene Herangehensweisen zur Darstellung und Navigation gezeigt.

Baum

Da eine facettierte Suche auf Taxonomien ausgeführt wird, bietet es sich an, zur Navigation die Hierarchie einer Taxonomie direkt wieder zu geben. Hierzu bietet sich ein expandierender Baum an, bestenfalls mit einer Funktion um einzelne Äste auf- und zuzuklappen wie dies auch in Abb. 2.5 zu sehen ist. Ein Baum eignet sich gut, um nur bestimmte Äste detailliert zu zeigen und um schnell zwischen den Ebenen der Taxonomie zu navigieren. Ein Nachteil kommt jedoch bei großen Taxonomien zum Vorschein, wenn der Baum nicht mehr auf dem gesamten Bildschirm darstellbar ist. Dann ist wie in Abb. 2.5 ein vertikaler Scrollbalken notwendig, bei großer Expansion der Äste eventuell auch ein horizontaler Scrollbalken. Beispiele für die Umsetzung eines Baums zur facettierten Suche findet man in [Fer08, TLH11, TAP08].

Liste

Eine Alternative zur Darstellung und Navigation stellt die Liste dar. Hier werden die Facetten einer Ebene kompakt und daher Platz sparend aufgelistet, so dass sich dies besonders für kleinere Bildschirme, wie z.B. bei mobilen Endgeräten, eignet. Eine Liste unterstützt in der Regel das Scrollen, kann aber auch als Seite bis zum Bildschirmende dargestellt

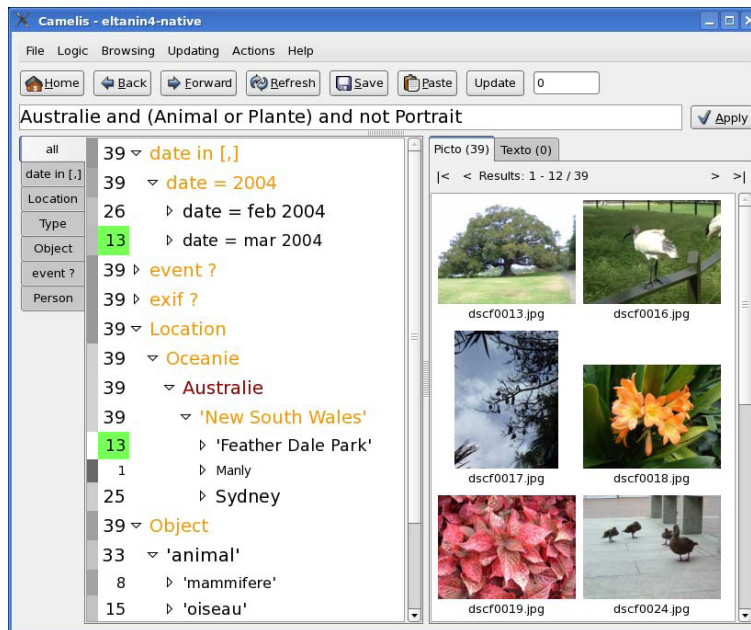
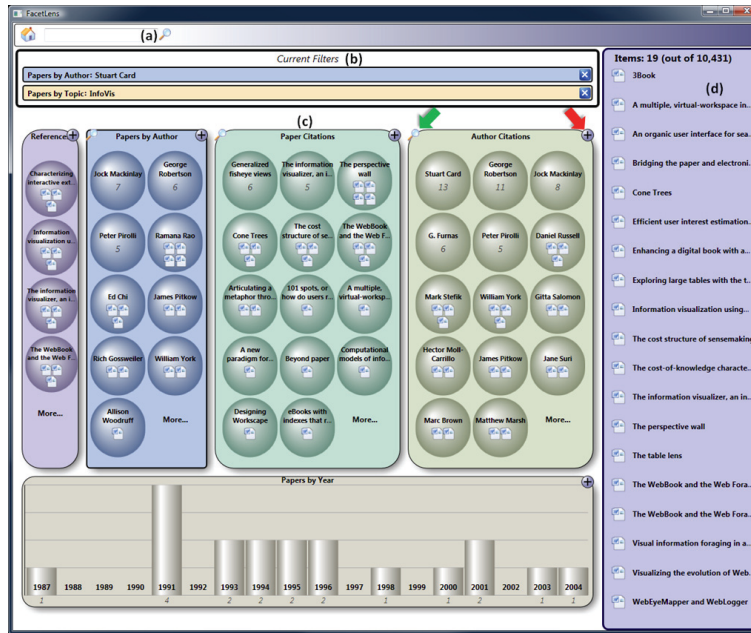


Abb. 2.5. Camelis [Fer08].

werden. Im letzteren Fall muss eine Möglichkeit zur Anzeige nicht dargestellter Facetten angeboten werden, z.B. durch einen *Weiter-* oder *Mehr-*Button (siehe auch Abb. 2.6) oder durch Angabe von Seitenzahlen. Ein Nachteil der Liste ist, dass durch sie die Hierarchie der Facetten nicht visualisiert wird, weshalb wenigstens die aktuelle Superfacette angegeben werden sollte. Möchte man zusätzlich den Pfad bis zur aktuellen Facette veranschaulichen, dann bieten sich *Breadcrumbs* an. Ein Beispiel für eine Anwendung mit einer scrollbaren Liste mit *Breadcrumbs* ist die in 3.2 vorgestellte Anwendung *MobileFacets* [Kle10]. Eine Liste als Seite wird z.B. in [EHTA11] verwendet.

Gruppierung

Wenn eine facettierte Taxonomie als Grundlage zur facettierten Suche dient, dann ist es sinnvoll, die einzelnen Taxonomien voneinander getrennt anzuzeigen. Dies lässt sich gut in einer Gruppierung von Facetten umsetzen, wie es z.B. auch in Abb. 2.6 illustriert wird. Hierbei können die Facetten einer Gruppierung auf unterschiedliche Weise dargestellt werden, z.B. als anklickbare Objekte in einem Grid wie in Abb. 2.6 oder die in 3.1 vorgestellte Anwendung *FaThumb* [KG06]. Weitere Möglichkeiten sind als Baum wie in [SVH07] oder als Liste wie in [HHMT09]. Bei größeren Hierarchien bietet es sich auch an, die Expansion per Zoom zu implementieren [DFW08, GMR07]. Wenn pro Taxonomie nur eine

Abb. 2.6. FacetLens [LSR⁺09].

Facette auswählbar sein soll, lässt sich dies einfach als *Drop-Down Liste* wie in [HEZ10, MDD09] umsetzen.

Anzahl an Ressourcen

Bei einer facettierten Suche ist der Benutzer weniger an den Facetten, sondern eher an den Ressourcen interessiert. Daher ist es nützlich, die Anzahl an Ressourcen, die einer Facette zugeordnet sind, anzuzeigen. Am einfachsten wird dies durch die Annotation der Facette mit der jeweiligen Anzahl gelöst. Zusätzlich zu diesem Ansatz wird in Abb. 2.6 die Zahl auch durch Symbole ersetzt. Verschiedene Größenordnungen können auch graphisch wiedergegeben werden, z.B. als Cluster [FZEB07] oder als Balken [ZM04].

Auswahl von Facetten

Ein weiterer wichtiger Aspekt der facettierten Suche ist das Auswählen von Facetten, also das Mitteilen der Anwendung zur Veränderung der Ergebnismenge. Grundsätzlich lässt sich die Auswahl in drei Kategorien unterscheiden: die einfache Auswahl, die mehrfache Auswahl und die Bereichsauswahl. Diese Kategorien werden nun näher erläutert.

Einfache Auswahl

Die einfache Auswahl bezieht sich nicht unbedingt auf das Auswählen insgesamt einer einzigen Facette, sondern eher auf das Auswählen einer Facette aus einer Taxonomie. Dies macht genau dann Sinn, wenn die Ressourcen nie mehr als einer Facette aus einer Taxonomie zugeordnet sind, also wenn zwei Facetten aus einer Taxonomie sich gegenseitig ausschließen. Dies kann wie zuvor erwähnt durch *Drop-Down Listen* [HEZ10, MDD09] umgesetzt werden. Weiter kann auch ein Klick auf eine Facette in einem Baum als einfache Auswahl realisiert werden [TLH11, vAWvH10].

Mehrfache Auswahl

In der Regel unterscheidet eine mehrfache Auswahl im Aussehen nicht, aus welcher Taxonomie eine Facette stammt, so dass die Auswahl einheitlich gestaltet wird. Das Aus- und Abwählen lässt direkt an der Facette nach dem *Checkbox*-Prinzip umsetzen. So kann neben der Facette eine Checkbox [FZEB07] dargestellt werden oder auch ein Plus-/Minus-Symbol [HHMT09]. Alternativ kann eine Facette auch als markiert oder nicht-markiert dargestellt werden [KG06]. Eine weitere Möglichkeit ist die Verwendung eines *Hinzufügen*-Buttons, welches eine weitere Ansicht benötigt, in der alle ausgewählten Facetten aufgelistet sind. Dort findet man auch einen entsprechenden *Löschen*-Button zum Abwählen [Kle10, EHTA11]. Eine solche Übersicht der ausgewählten Facetten ist auch beim *Checkbox*-Prinzip nützlich. Wenn jedoch dort auch eine Option zum Abwählen der Facette angeboten wird, so muss darauf geachtet werden, dass die Checkbox, das Symbol oder die Markierung der Facette dementsprechend aktualisiert wird. Wie die Facetten miteinander verknüpft werden, spielt zwar bei der einfachen Auswahl auch eine Rolle, kommt aber bei der mehrfachen Auswahl eher zum Tragen. Häufig werden die Facetten immer gleichermaßen miteinander verknüpft, also nur per *UND* oder per *ODER*. Möchte man aber auch Kombinationen zulassen, so lässt sich dies in einem Textfeld wie in Abb. 2.5 umsetzen.

Bereichsauswahl

In manchen Fällen macht es Sinn, nicht nur eine oder mehrere Facetten, sondern auch einen Bereich auswählen zu können, z.B. bei einer Facette zu Preisen, Zeiten oder Längenangaben. Hierzu ist es nötig, eine Unter- und Obergrenze anzugeben, beispielsweise durch ein Textfeld. Um allerdings Fehleingaben zu vermeiden, lässt sich die Bereichsauswahl besser an Hand eines *Sliders* realisieren [GMR07]. Ein Slider hat den Vorteil, dass nur Positionen angeboten werden können, die auch wirklich als Facette existieren. Intern muss eine solche Auswahl als Auswahl mehrerer Facetten implementiert werden, so dass die Anwendung auch die erwarteten Ergebnisse liefert.

Darstellung der Ergebnisse

Eine facettierte Suche dient des Findens von gesuchten Ressourcen. Demzufolge müssen diese auch dargestellt und dem Nutzer präsentiert werden. Eine simple Darstellungsmöglichkeit bietet die Liste, entweder wie zuvor erwähnt durchwegs scrollbar [KG06, Kle10] oder im Seitenformat [SVH07, MDD09]. Hier können die Ressourcen nach dem Alphabet oder einem anderen Kriterium sortiert oder neu geordnet werden. Im Gegensatz zu einer einzigen Liste kann eine Gruppierung der Ressourcen die Zugehörigkeit zu bestimmten ausgewählten Facetten besser wiedergeben und die Suche so eventuell vereinfachen [DFW08, GMR07]. Weiter kann eine besondere Anordnung der Ressourcen die Beziehungen untereinander widerspiegeln und so dem Nutzer mehr Informationen mitteilen [FZEB07]. *Pivot* oder *Querying by Examples* Funktionalitäten (siehe auch 2.2.1) werden sinngemäß bei den Ressourcen angeboten, wie man z.B. in [WRS⁺05, Fer08] sehen kann.

Suchfeld

Obwohl die facettierte Suche als explorative Suche konzipiert ist, kann in vielen solchen Anwendungen dennoch auch eine direkte Suche durchgeführt werden. Dies kann einerseits direkt zur gesuchten Ressource führen [LSR⁺09, MDD09], andererseits kann dies auch als Einstiegspunkt zur Anzeige spezieller Facetten dienen [HHMT09, FZEB07], also dass nur Facetten aus dem Kontext der Texteingabe dargestellt werden. Für beide Fälle wird üblicherweise ein Inputfeld zur Texteingabe benötigt. Eine weitere Möglichkeit der Verwendung eines Suchfelds ist die Suche in bereits vorhandenen Facetten [TLH11, HvOH06]. Dadurch lässt sich die Navigation in den Facetten vereinfachen bzw. abkürzen. Dabei kann eine solche Suche über alle Facetten, nur ausgewählte Facetten oder nur in bestimmten Taxonomien erfolgen. Zur Fehlerreduzierung bietet sich die Unterstützung der Autovervollständigung bei der Texteingabe an.

Dies schließt die Betrachtung der unterschiedlichen Bestandteile der facettierten Suche ab. Nach dieser Vorstellung von verschiedenen Klassifikationen an Hand von Taxonomien und der grundsätzlichen Beschreibung der Prinzipien und Funktionalitäten sowie dem Aufbau einer facettierten Suche folgt im nächsten Kapitel eine genauere Untersuchung zweier mobiler Anwendungen zur facettierten Suche, FaThumb und MobileFacets, und es wird gezeigt, in welchem Zusammenhang sie zur zu entwickelnden Anwendung MFacets stehen.

Anforderungen und Analyse

Wie eingangs in der Einführung erwähnt, werden in dieser Diplomarbeit die beiden mobilen Anwendung zur facettierten Suche *MFacets* und *MobileFacets* bezüglich der Benutzbarkeit evaluiert. *MobileFacets* ist das Produkt der Diplomarbeit von Alexander Kleinen [Kle10], welche das Ziel hat, eine mobile Anwendung zur facettierten Suche von Grund auf neu zu entwerfen. *MFacets* wird in dieser Diplomarbeit entwickelt und wird einerseits Teile der Architektur von *MobileFacets* zur schnelleren Implementierung wiederverwenden. Andererseits beruht *MFacets*' Benutzeroberfläche konzeptionell auf der facettierten Suche der Anwendung *FaThumb* [KG06]. *FaThumb* ist für Handys mit Nummernblock ausgelegt und arbeitet auf einem geschlossenen Datensatz. Daher hat *MFacets* das Ziel, *FaThumb* zu modernisieren und an aktuelle Gegebenheiten, wie z.B. Smartphones mit Touchscreeneingabe oder Anschluss an öffentlich zugängliche Daten durch das mobile Internet, anzupassen. Bevor jedoch *MFacets* in Kapitel 4 vorgestellt wird, gilt es im weiteren Verlauf zuerst die Anwendungen *FaThumb* und *MobileFacets* zu erläutern.

3.1 FaThumb

FaThumb [KG06] ist eine mobile Anwendung zur facettierten Suche auf einem Datensatz mit ca. 39.000 Einträgen (entnommen aus den Yellow Pages der Metropole Seattle) und zeichnet sich durch eine neuartige Navigation durch die Facetten aus. Wie man in Abb. 3.1 sieht, werden bis zu acht Facetten in Form eines 9er-Grid angeordnet, welches die neun Tasten des Nummernblocks eines Handys nachempfunden ist. *FaThumb* läuft auf *Piccolo.NET*¹ und ist für Handys ohne Touchscreenfähigkeit konzipiert, weshalb die Facetten über die einzelnen Nummerntasten angewählt werden können, d.h. ein Klick z.B. auf die Nummer 3 bewirkt einen Klick auf die rechte obere Facette.

¹ siehe <http://www.cs.umd.edu/hcil/piccolo/>

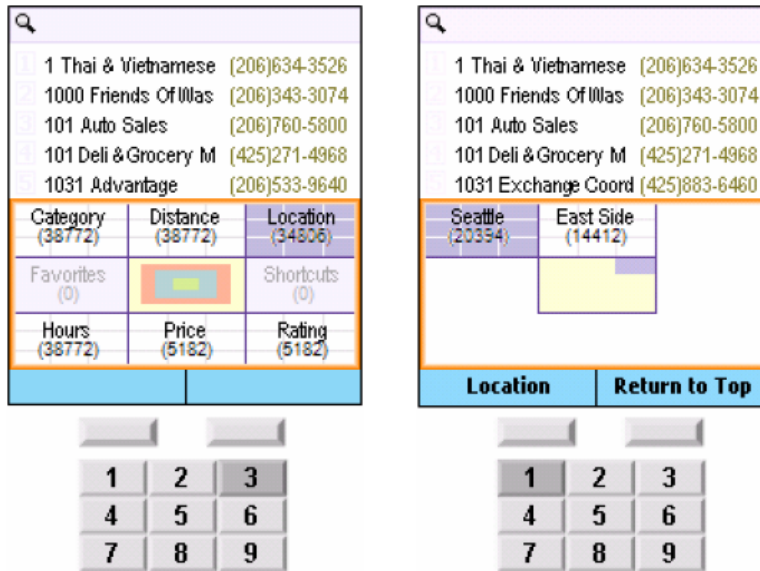
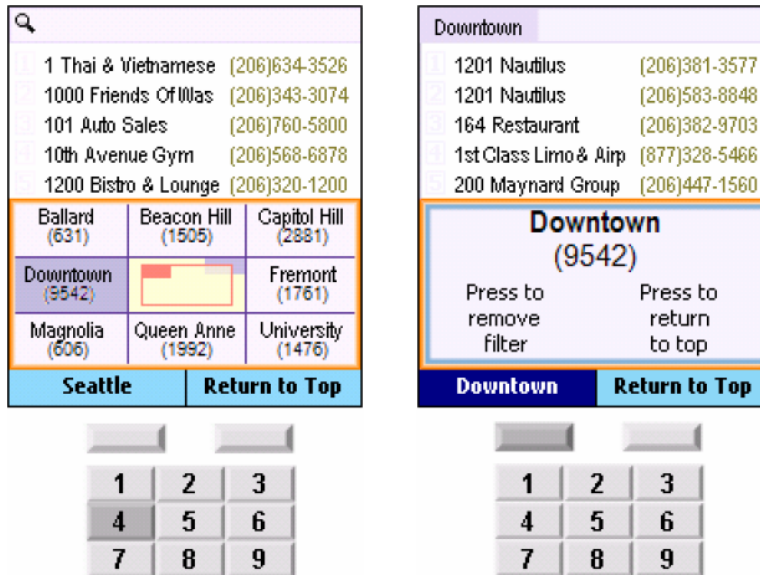


Abb. 3.1. Hauptansicht und Navigation in FaThumb [KG06].

In der Anfangsansicht von FaThumb (siehe Abb. 3.1(a)) bilden nur sechs Felder Facetten ab, nämlich die obere und untere Reihe. Die mittlere Reihe verwendet zwei Felder um Favoriten und Abkürzungen anzuzeigen. Unter Favoriten versteht man hier komplett getätigte Suchen mit Auswahl mehrerer Facetten und unter Abkürzungen zuvor aufgesuchte Facetten, die in tieferen Hierarchieebenen zu finden sind. Das mittlere Feld dient der Navigation durch die Facetten und zeigt an, welchen Pfad man bisher zurückgelegt hat. In Aktion sieht man dieses Navigationsfeld erst nach dem ersten Klick auf eine Facette wie in Abb. 3.1(b). Hier sieht man eine Markierung oben rechts im Navigationsfeld, was darauf hindeutet, dass eine Ebene zuvor auf die Facette oben rechts geklickt worden ist. Wird in der nächsten Ebene wieder eine Facette ausgewählt, so ändert sich das Navigationsfeld dementsprechend und fügt eine neue Ebene in einer anderen Farbe hinzu (siehe Abb. 3.2(a)). FaThumb unterstützt bis zu fünf Hierarchieebenen, so dass das Navigationsfeld bis zu vier Ebenen anzeigen kann.

In FaThumb sind nur die Facetten am Ende eines Pfades (Blattknoten) auswählbar, welche sich im Aussehen von anderen Facetten unterscheiden. Eine Superfacette wird mit einem Gitter im Hintergrund dargestellt (siehe Abb. 3.1), während ein Blattknoten einen normalen Hintergrund besitzt (siehe Abb. 3.2(a)). Neben dem Facettennamen wird außerdem noch die Anzahl an Ressourcen mit abgebildet. Möchte man nun eine Facette selektieren, so wird diese zuerst vergrößert angezeigt (siehe Abb. 3.2(b)) und kann daraufhin über



(a) Klick auf 4 für Downtown.

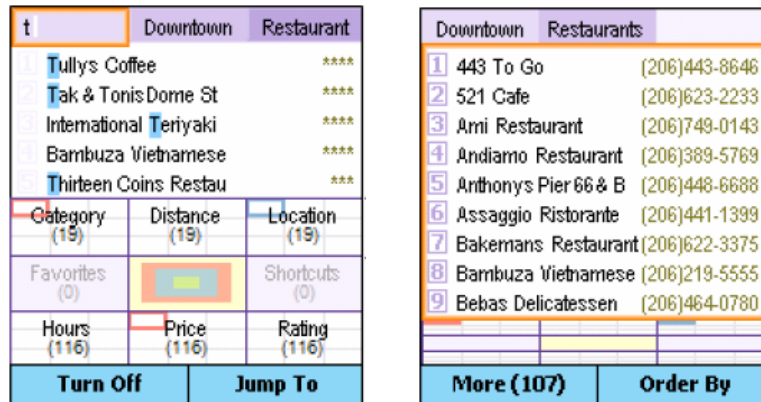
(b) Klick auf linke Optionstaste zur Auswahl der Facette.

Abb. 3.2. Navigation und Auswahl in FaThumb [KG06].

die linke Optionstaste des Handys ausgewählt werden. Nach der Auswahl wird die Facette in einer Leiste am oberen Bildschirmrand, in FaThumb Filterregion genannt, wiedergegeben und die über den Facetten dargestellten Ergebnisse verändern sich direkt entsprechend der ausgewählten Facette.

FaThumb bietet noch eine weitere Möglichkeit dem Benutzer mitzuteilen, dass Facetten ausgewählt worden sind. Wenn eine Subfacette ausgewählt ist, wird dies im Hintergrund der Superfacette angezeigt (siehe Abb. 3.3(a)). Je nach dem wo die Subfacette im 9er-Grid abgebildet wird, wird im entsprechenden Feld im Hintergrund der Superfacette eine Markierung gesetzt. Außerdem wird die Farbe der Markierung so gewählt, dass sie die Hierarchieebene der ausgewählten Facette repräsentiert. Da die Facetten in FaThumb nach dem UND-Prinzip verknüpft werden, ändert sich folgerichtig auch die Anzahl an Ressourcen pro Facette. Würde nach Auswahl einer Facette eine leere Ergebnismenge entstehen, so wird diese Auswahl nach dem Poka Yoke Prinzip durch ein Ausgrauen der Facette unterbunden.

Über die Navigationstasten des Handys (oben, unten, rechts und links) kann zwischen den Bearbeitungsregionen gewechselt werden, also der Facettennavigation, der Ergebnisliste oder der Filterregion. Der aktuelle Fokus wird durch einen orangenen Rahmen um die jeweilige Bearbeitungsregion verdeutlicht. Liegt der Fokus auf der Ergebnisliste (siehe Abb. 3.3(b)), dann wird diese expandiert um bis zu neun Ressourcen darstellen zu können. Diese



(a) Hinzufügen eines textuellen Filters.

(b) Expansion der Ergebnisliste.

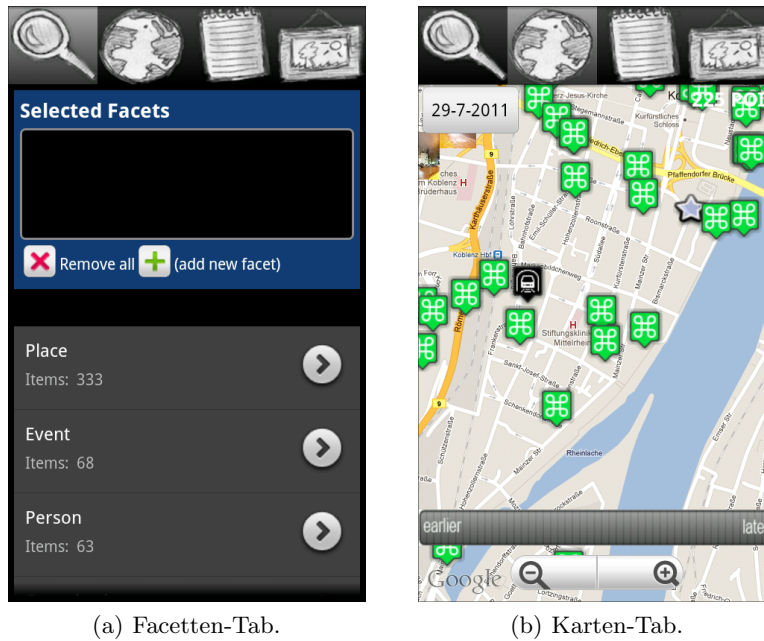
Abb. 3.3. Filterregion und Ergebnisliste in FaThumb [KG06].

Ressourcen können mit den Tasten 1-9 näher betrachtet werden und, wenn möglich, können weitere Ressourcen mit der linken Optionstaste angezeigt werden. Wie viele weitere Ressourcen dargestellt werden können, wird entsprechend visualisiert. Die vorhandenen Ergebnisse können über die rechte Optionstaste nach verschiedenen Kategorien sortiert werden, z.B. nach Alphabet, Distanz oder Bewertung. Weiterhin lassen sich die Ergebnisse auch an Hand eines textuellen Eingabefeldes filtern (siehe Abb. 3.3(a)). Dies geschieht in der Filterregion, so dass der Filtertext neben den ausgewählten Facetten angezeigt wird. Zusammen verdeutlicht dies auch, welche Bedingungen für die dargestellten Ressourcen gilt. Der Filter lässt sich auch auf die gesamte Ergebnismenge anwenden, so dass FaThumb auch eine direkte Suche nach Ressourcen unterstützt.

Hiermit ist die Erklärung der Funktionsweise von FaThumb abgeschlossen. Welche Funktionalitäten und Benutzeroberflächenelemente in MFacets auch umgesetzt werden, wird in Abschnitt 3.3 präsentiert. Vorerst wird aber noch MobileFacets im nächsten Abschnitt näher untersucht.

3.2 MobileFacets

MobileFacets [Kle10] ist ein ortsbezogener Dienst, welcher seine Daten bzw. Ressourcen aus öffentlich zugänglichen Datensätzen bezieht, und läuft auf Smartphones mit Touchscreeneingabe und Google Android Betriebssystem ab



(a) Facetten-Tab.

(b) Karten-Tab.

Abb. 3.4. Facetten- und Karten-Tab in MobileFacets [Kle10].

Version 2.1². Als Quellen dienen MobileFacets hierzu DBpedia³, GeoNames⁴, Eventful⁵, Upcoming⁶ und Flickr⁷. Diese Ressourcen lassen sich per facetierter Suche ergründen und gegebenenfalls auf einer Karte darstellen. MobileFacets verfolgt einen Tab-basierten Ansatz, d.h. dass über Tabs verschiedene Bearbeitungsbildschirme geöffnet werden können. Somit gibt es Tabs für die Auswahl der Facetten (siehe Abb. 3.4(a)), die Kartenansicht (siehe Abb. 3.4(b)), die Ergebnisliste (siehe Abb. 3.5(a)) und die Galerie zur Darstellung von Bildern (siehe Abb. 3.5(b)).

Im Facetten-Tab gibt es zwei Bereiche, zum einen die Anzeige ausgewählter Facetten und zum anderen die Facettennavigation. Die Facetten werden in einer scrollbaren Liste dargestellt und jeder Eintrag enthält sowohl den Facettennamen als auch die Anzahl an zugeordneten Ressourcen. Wird eine Facette angeklickt, so wird sie sofort zu den ausgewählten Facetten in der zugehörigen Ansicht hinzugefügt (siehe Abb. 3.6). Somit lassen sich auch Nicht-Blattknoten auswählen. Die Facettenliste zeigt maximal zwei

² siehe <http://www.android.com/>

³ siehe <http://dbpedia.org/About>

⁴ siehe <http://www.geonames.org>

⁵ siehe <http://eventful.com/>

⁶ siehe <http://upcoming.yahoo.com/>

⁷ siehe <http://www.flickr.com/>

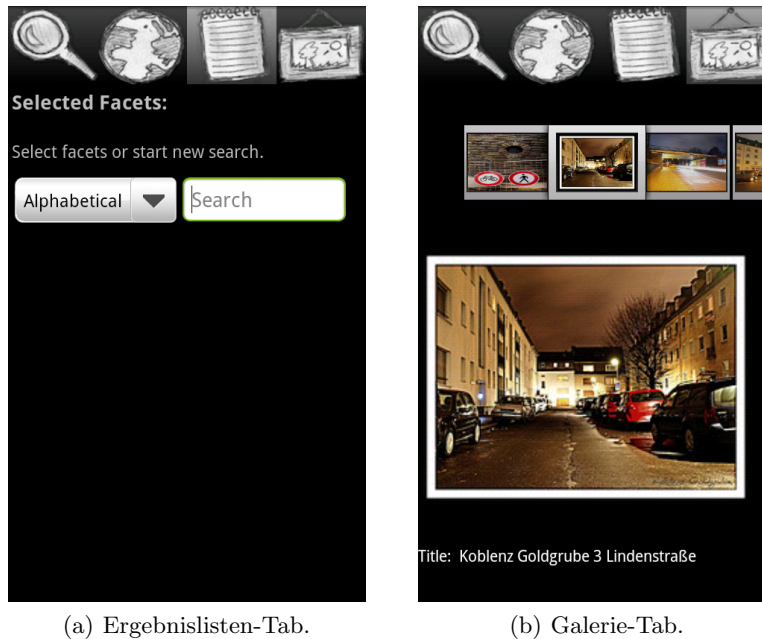


Abb. 3.5. Ergebnislisten- und Galerie-Tab in MobileFacets [Kle10].

Hierarchieebenen an und wählt man in der zweiten Ebene eine Facette aus, so werden die Ressourcen nur zu dieser Facette in derselben Liste angezeigt. Dadurch wird ein Wechsel zur Ergebnisliste, um zu einer bestimmten Ressource zu gelangen, nicht unbedingt notwendig. Möchte man mehrere Facetten auswählen, so kann man die aktuelle Facette über den *Hinzufügen-Button* (in MobileFacets als *add new facet* bezeichnet) permanent als ausgewählt belassen. Nach einem Klick auf diesen Button springt die Anwendung zurück in die Ursprungsansicht, um eine neue Suche nach einer Facette zu ermöglichen. Die aktuelle Facette wird in Form eines Breadcrumbs immer in der Mitte des Bildschirms zwischen den beiden Bereichen angegeben. Eine ausgewählte Facette kann in MobileFacets auf drei Weisen entfernt werden. Ist eine Facette nicht zu den ausgewählten permanent hinzugefügt worden, so reicht ein Klick auf die *Zurück-Taste* des Smartphones um diese implizit zu entfernen und entweder die Superfacette oder gar keine Facette auszuwählen. Einzelne Facetten können jederzeit explizit über den *Löschen-Button* zur Selbigen entfernt werden. Alle Facetten entfernt man am einfachsten über den *Alles-Löschen-Button*. Nach einem expliziten Löschen springt MobileFacets wieder in die Ursprungsansicht, so dass der Benutzer eine neue Suche starten kann.

Besitzen Ressourcen zu ausgewählten Facetten eine Geokoordinate, auch *Point-of-Interest* (POI) genannt, so können diese auf einer Karte von Google

Maps⁸ angezeigt werden (siehe Abb. 3.4(b)). Zur Darstellung dieser POIs werden Symbole verwendet, die sich je nach Facettenzuordnung einer Ressource unterscheiden. Die Detailansicht zu einer Ressource kann durch das Anwählen eines Symbols geöffnet werden. Die Karte bietet die üblichen Funktionen des Heran- und Herauszoomens über zwei Buttons am unteren Ende des Bildschirms an. Die Änderung der Karte auf eine Satellitenansicht ist über einen Menü-Eintrag nach Betätigung der *Menü-Taste* des Smartphones möglich. Ist das Smartphone GPS⁹-fähig und ist dieses auch eingeschaltet, so kann der Benutzer über einen weiteren Menü-Eintrag sich seinen aktuellen Standort anzeigen lassen. Es ist zu erwähnen, dass die ortsbezogenen Suchen von MobileFacets sich an dem Kartenausschnitt im Karten-Tab orientieren, so dass für eine Suche an einem anderen Ort dieser Ausschnitt entsprechend geändert werden muss. Wie viele POIs auf der Karte dargestellt werden, ist in der oberen rechten Ecke visualisiert. Zusätzlich wird in der oberen linken Ecke ein Datum angegeben, welches entweder durch ein Klick auf die zugehörige Fläche oder über die Zeitleiste am unteren Bildschirmrand verändert werden kann. Dies hat den Zweck, dass Veranstaltungen (in Abb. 3.4(b) als Stern gekennzeichnet) je nach zeitlicher Entfernung des Startzeitpunkts unterschiedlich angezeigt werden können. So wird der Stern zu einer noch weit entfernten Veranstaltung transparenter dargestellt als eine bald stattfindende Veranstaltung.

Die Ergebnisliste erreicht man über den dritten Tab und wird initial leer präsentiert, wie man es in Abb. 3.5(a) sehen kann. Dies liegt daran, dass in MobileFacets die ausgewählten Facetten nach dem ODER-Prinzip verknüpft werden. Da nach dem Start der Anwendung noch keine Facetten ausgewählt sind, ist die Ergebnisliste dementsprechend leer. Dies führt dazu, dass die Auswahl einer Facette die Ergebnisliste um Ressourcen erweitert und nie verringert (siehe Abb. 3.6). Somit müssen im Gegensatz zu FaThumb keine Facetten ausgegraut werden, da keine leere Ergebnisliste bei Auswahl einer Facette entstehen kann. Ist eine Facette ausgewählt, so wird sie zusammen mit der Anzahl an dargestellten Ressourcen unter der Tableiste angegeben. Die Ressourcen werden so präsentiert, dass, wenn möglich, ein Ausschnitt der Beschreibung der Ressource und die Entfernung zum Suchort angezeigt werden. Zusätzlich wird, wie auf der Karte, die Facettenzugehörigkeit der Ressource neben dem Facettennamen durch ein Symbol illustriert. Je nach Bedarf können die Ressourcen entweder nach Alphabet oder nach Distanz sortiert oder über ein Texteingabefeld gefiltert werden.

Der vierte Tab führt zur Galerie (siehe Abb. 3.5(b)), in der verschiedene Bilder zur Anfangssuche angezeigt werden. Diese Bilder werden nicht von der Auswahl von Facetten beeinflusst und bleiben so unverändert in der Galerie bestehen. Zu den Bildern wird der zugehörige Titel unterhalb des Bildes

⁸ siehe <http://maps.google.com/>

⁹ Global Positioning System

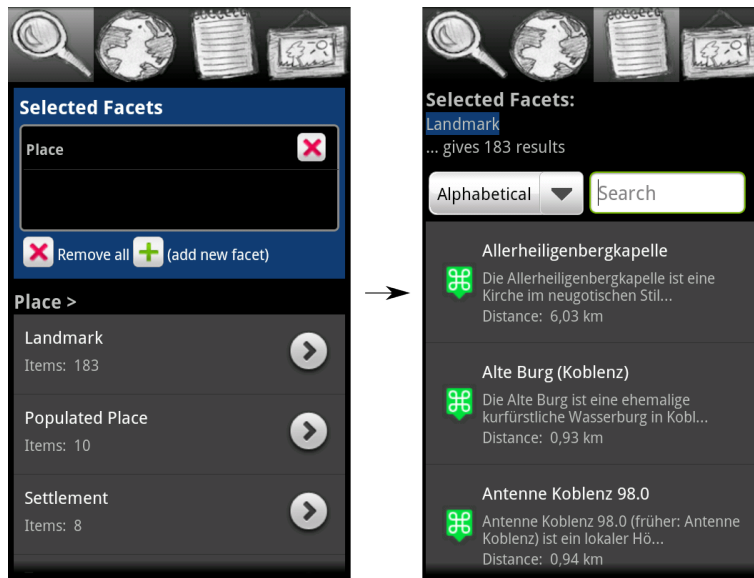


Abb. 3.6. Auswahl einer Facette im Facetten- und Ergebnislisten-Tab in MobileFacets [Kle10].

angegeben und weitere Bilder lassen sich in einer scrollbaren Miniansicht der Bilder auswählen und anzeigen.

MobileFacets bietet neben der ortsbezogenen Suche auch eine Suche nach Stichwörtern. Dazu muss die *Suchen-Taste* des Smartphones betätigt werden und das Stichwort in ein erscheinendes Suchfeld eingegeben werden. Während der Eingabe werden Vorschläge an Stichwörtern in einer Liste angezeigt, die logischerweise ausgewählt werden können, um eine entsprechende Suche zu starten.

Dies beendet die Beschreibung von MobileFacets, so dass im nächsten Abschnitt gezeigt werden kann, was aus FaThumb und aus MobileFacets zur Implementation von MFacets übernommen wird und in welchen Punkten sich die beiden Anwendungen von MFacets unterscheiden werden.

3.3 Anforderungen für MFacets

Dieser Abschnitt dient der Übersicht, welche Funktionalitäten und designtechnischen Elemente aus FaThumb und MobileFacets in MFacets umgesetzt werden und in welchen Punkten sich MFacets von diesen Anwendungen im Einzelnen abgrenzen wird. Dies wird am Ende des Abschnitts durch die Tabelle 3.1 nochmals verdeutlicht.

3.3.1 Übereinstimmungen mit FaThumb und MobileFacets

MFacets ist als Modernisierung von FaThumb konzipiert, so dass es Nahe liegt, dass die in FaThumb implementierte, neuartige Navigation durch die Facetten auch in MFacets realisiert wird. Dies bedeutet, dass MFacets den Bildschirm ebenso in einen Bereich für die Ergebnisliste und in einen Bereich für die Facettennavigation aufteilen wird. Die Facetten werden in einem 9er-Grid präsentiert, wobei auch in MFacets das mittlere Feld der Navigation und dem Zurechtfinden in der Taxonomie dienen wird. Die Expansion der Ergebnisliste und die Sortierung bzw. Filterung der zugehörigen Ressourcen wird ebenfalls in MFacets umgesetzt, allerdings nicht zwangsweise über dieselben Eingabemethoden, insbesondere da Smartphones keine rechte Optionstaste zum Aufrufen der Sortierung zur Verfügung stehen. Um den textuellen Filter in MFacets zur direkten Suche verwenden zu können, wird die Ergebnisliste auch initial mit allen Ressourcen gefüllt sein.

Um die Implementation von MFacets zu erleichtern, wird MFacets einige Eigenschaften mit MobileFacets gemein haben. Somit wird MFacets auch auf einem Smartphone mit Touchscreenunterstützung und Google Android Betriebssystem laufen. Außerdem wird MFacets ebenso auf öffentlich zugängliche Datensätze zurückgreifen, um Ressourcen für die facetthierte Suche zu erhalten. Allerdings ist schon im Vorhinein beschlossen worden, dass die Quellen der Datensätze für MFacets um Qype¹⁰ erweitert werden. Die Verknüpfung der Facetten nach dem ODER-Prinzip hat sich in MobileFacets für diese Ressourcen bewährt, so dass dies auch in MFacets umgesetzt wird. MFacets wird ebenfalls als ortsbezogener Dienst agieren können, so dass die Karte wie auch die Symbole zur POI-Darstellung und weitere Elemente und Funktionalitäten des Karten-Tabs von MobileFacets übernommen werden. Da MFacets auch Fotos als Ressourcen erhalten wird, bietet sich die Verwendung der Galerie aus MobileFacets an. Da MobileFacets schon Eingabelemente zur Sortierung und Filterung der Ressourcen in der Ergebnisliste besitzt, werden diese in MFacets ebenfalls implementiert. Des Weiteren wird MFacets die Architektur und Schnittstellen des von MobileFacets verwendeten Servers verwenden und gegebenenfalls erweitern, so dass die Möglichkeit der Suche nach Stichwörtern ebenfalls implementiert werden kann.

3.3.2 Abgrenzung von FaThumb und MobileFacets

MFacets wird sich in zwei Punkten grundsätzlich von FaThumb unterscheiden. Zum einen ist dies die Umsetzung auf einem Touchscreen-fähigen Smartphone und zum anderen die Verwendung eines offenen Datensatzes.

Ersteres bedeutet, dass jegliche Funktionalitäten mit Verbindung zu den Tasten des Nummernblocks eines Handys für MFacets obsolet werden. Diese Funktionalitäten gilt es in MFacets entweder durch einen Klick auf den Touchscreen oder durch das Hinzufügen spezieller Buttons zu realisieren. Ebenfalls

¹⁰ siehe <http://www.qype.com/>

ist die Verwendung eines Rahmens zur Verdeutlichung des fokussierten Bereichs nicht mehr notwendig, da in MFacets die Abhängigkeit zwischen Fokus und Tastendruck nicht mehr wie in FaThumb vorhanden ist.

Da in FaThumb ein geschlossener Datensatz verwendet wird, kann dieser so präpariert werden, dass eine Facette nie mehr als acht Subfacetten besitzt. Dies ist in MFacets wegen des offenen Datensatzes nicht mehr möglich, so dass MFacets die Herausforderung angehen muss, mehr als acht Subfacetten anzeigen zu können ohne das Design des 9er-Grids zu verwerfen. Die Beschränkung FaThumbs auf die Auswahl nur von Blattknoten wird für MFacets nicht übernommen, so dass die Möglichkeit für die Auswahl von Superfacetten realisiert werden muss. Auf Grund der Art der Ressourcen, wird sich MFacets von FaThumb in der Verwendung einer Karte und einer Galerie und in der Verknüpfung der Facetten nach dem ODER-Prinzip unterscheiden.

Der grundlegende Unterschied zwischen MFacets und MobileFacets ist die unterschiedliche Darstellung der Facetten und die zugehörige Navigation durch diese. Bezüglich der Navigation wird MFacets die Möglichkeit bieten mehr als nur zwei Hierarchieebenen zu erkunden, so dass das Navigationsfeld in MFacets mehr Sinn ergibt. Weiter wird sich MFacets durch die Darstellung der Ergebnisliste und der Facetten auf demselben Bildschirm von MobileFacets abgrenzen. MFacets und MobileFacets werden zwar beide die Facetten nach dem ODER-Prinzip verknüpfen, dennoch wird die Ergebnisliste in MFacets anfangs mit allen Ressourcen gefüllt angezeigt und das ODER-Prinzip erst nach der Auswahl der ersten Facette anwenden. Um möglichst viele Ressourcen in der Ergebnisliste oberhalb der Facettennavigation anzuzeigen, wird MFacets keine Tabs zum Wechseln zur Karte oder Galerie verwenden. Dies wird über Menü-Einträge möglich sein. Da MFacets auf FaThumb beruht und dort keine Breadcrumbs verwendet werden, stellt dies einen weiteren Unterschied zu MobileFacets dar.

Nach dieser Beschreibung der Gemeinsamkeiten und Unterschiede zwischen MFacets und den Anwendungen FaThumb und MobileFacets, welche zur Übersicht in Tabelle 3.1 nochmals aufbereitet ist, folgt im nächsten Kapitel die genaue Beschreibung der Entwicklung von MFacets.

	FaThumb →	MFacets	MobileFacets
Datensatz	geschlossen	offen	offen
Facettenanordnung	9er-Grid	9er-Grid	Liste
initiale Ergebnisliste	voll	voll	leer
Orientierung	Navigationsfeld	Navigationsfeld	Breadcrumbs
Wechsel des Bearbeitungsbildschirms	Navigations-tasten	Klick oder Menü	Tabs

Tabelle 3.1. Vergleich zwischen FaThumb, MFacets und MobileFacets.

MFacets

MFacets stellt die mobile Anwendung dar, die im Zuge dieser Diplomarbeit entwickelt worden ist. Wie schon in Kapitel 3 ausgiebig erläutert worden ist, beruht MFacets unter dem Gesichtspunkt der Benutzeroberfläche auf der mobilen Anwendung FaThumb und passt diese an aktuelle Gegebenheiten an. Um die Implementation zu erleichtern, werden bestimmte Funktionalitäten von MobileFacets übernommen. Daher beschäftigt sich dieses Kapitel mit der Entwicklung von MFacets. Zuerst wird in 4.1 die Architektur des kompletten MFacets-Systems näher beleuchtet, also das Zusammenspiel zwischen Client, Server und den Datensätzen der öffentlich zugänglichen Quellen. Daraufhin wird in 4.2 näher auf die Realisierung des Clients eingegangen, bevor die Serverseite in 4.3 präsentiert wird. Zuletzt schließt 4.4 dieses Kapitel ab, indem gezeigt wird, inwieweit MFacets und die Vergleichsanwendung MobileFacets verändert und erweitert werden müssen, um die Basis für die Durchführung einer erfolgreichen Evaluation zu legen.

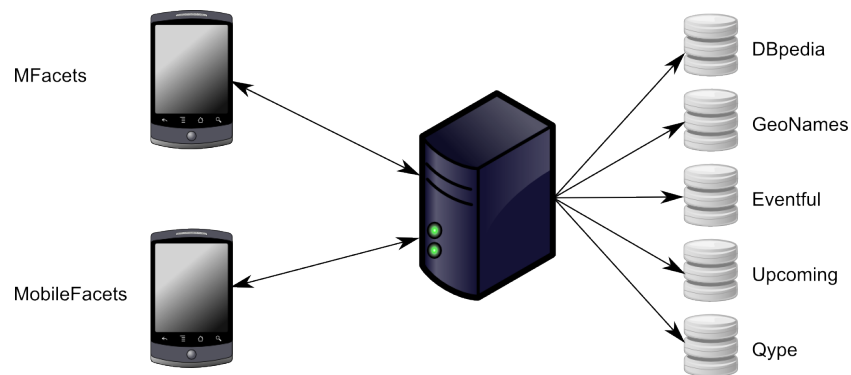


Abb. 4.1. Architektur des MFacets-Systems mit Unterstützung von MobileFacets.

4.1 Architektur

Die grundsätzliche Architektur des MFacets-Systems wird in Abb. 4.1 illustriert und lässt sich in drei Schichten einteilen, die Client-, Server- und Datenschicht. Die Clientschicht besteht aus den mobilen Anwendungen MFacets und MobileFacets, welche beide mit demselben Server kommunizieren. Dieser Server verarbeitet die Anfragen und entweder antwortet er direkt oder er leitet diese an die Datenrepositories der Datenschicht weiter. Die einzelnen Datenrepositories liefern dem Server die Ressourcen, welcher alle Ressourcen sammelt und in einer einzelnen Antwort an die jeweiligen Clients zurückliefert.

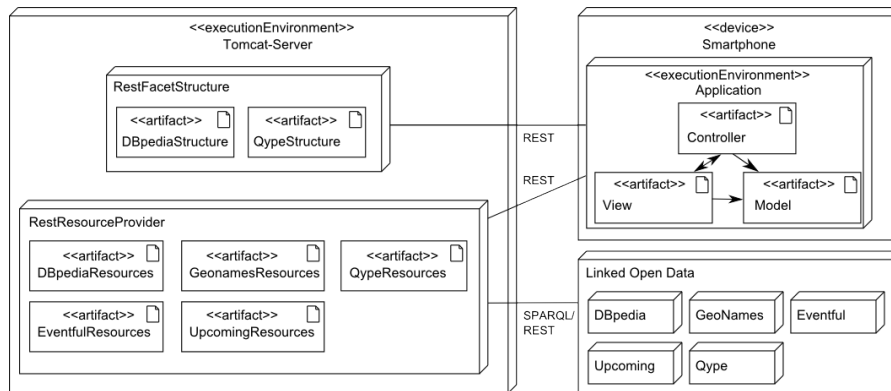


Abb. 4.2. Verteilungsdiagramm der Architektur des MFacets-Systems.

Um einen detaillierteren Blick auf diese Architektur zu erhalten, lässt sich diese auch in einem Verteilungsdiagramm darstellen (siehe Abb. 4.2). Hier sieht man, dass die Kommunikation zwischen der Anwendung und dem Tomcat¹-Server per REST² und zwischen Server und den Datenrepositories per REST oder SPARQL³ vonstatten geht. REST hat den Vorteil, dass die Verbindung zum Server über HTTP⁴ aufgebaut werden kann, so dass Parameter zur Anfrage einfach per HTTP GET übergeben werden können. Welche Parameter vom Server verstanden werden und was sie im Einzelnen bewirken, wird später in 4.3 vertieft. Dort wird auch näher erläutert, wie der Server die Datenrepositories getrennt anfragt, die Ressourcen erhält, die Taxonomie (hier *Facettenstruktur*) erstellt und die notwendigen Informationen an die Clients weiterleitet. Wozu die Anwendung die Facettenstruktur benötigt,

¹ siehe <http://tomcat.apache.org/>

² Representational State Transfer, siehe auch <https://www.ibm.com/developerworks/webservices/library/ws-restful/>

³ SPARQL Protocol And RDF Query Language, siehe auch <http://www.w3.org/TR/rdf-sparql-query/>

⁴ Hypertext Transfer Protocol

wird in Abschnitt 4.2.3 dargelegt. In der Abbildung erkennt man auch, dass die Anwendung nach dem Model-View-Controller-Prinzip⁵ aufgebaut ist. Dieses Prinzip sorgt für eine übersichtliche Trennung des Datenmodells von der Benutzeroberfläche und der Programmsteuerung. Wie dies genau umgesetzt ist, wird nun im nächsten Abschnitt beschrieben.

4.2 Client

Sowohl MFacets als auch MobileFacets sind nach dem Model-View-Controller-Prinzip entwickelt. Dies bedeutet, dass die Klassen des Datenmodells getrennt von den Klassen zur Benutzeroberfläche und den Klassen zur Programmsteuerung betrachtet und erstellt werden. Diese Trennung unterstützt zum einen die Übersichtlichkeit der Implementation und zum anderen sorgt sie dafür, dass Änderungen oder Erweiterungen in den einzelnen Komponenten erleichtert werden. Diese jeweiligen Aspekte werden im Folgenden näher vorgestellt. Zuvor sei noch zu erwähnen, dass MFacets in JAVA mit der Entwicklungsumgebung Eclipse⁶ in der Version 3.6.1 und dem Android SDK⁷ in der Revision 10 implementiert worden ist.

4.2.1 Datenmodell

Das Datenmodell von MFacets wird durch drei Klassen repräsentiert, einer Klasse für Facetten, genannt **Facet**, einer Klasse für Ressourcen, genannt **ResourceWrapper**, und einer Klasse zur Navigation zwischen den Facetten, genannt **FacetTreeNode**. Die Bedeutung und Funktionsweise dieser Klassen wird nun erläutert.

Facet

Für die facettierte Suche sind Facetten essentiell und wird folglich in einer eigenen Klasse abgebildet. Diese Klasse ist jedoch äußerst einfach gehalten und enthält nur die nötigsten Attribute. Somit benötigt eine Facette in MFacets zu zwei Attribute. Ein Attribut für den Facettennamen und einen zum Speichern der Anzahl an Ressourcen, die dieser Facette zugeordnet sind. Dies sind genau die Attribute, die in der Facettennavigation angezeigt werden.

ResourceWrapper

Eine Ressource ist im Vergleich zur Facette etwas komplexer, da diese Klasse Attribute und Methoden für verschiedene Arten an Ressourcen zur Verfügung stellen muss. So gibt es spezifische Attribute, die nur auf einen Ort oder auf eine Person oder auf eine Veranstaltung zutreffen.

⁵ siehe auch <http://www.martinfowler.com/eaDev/uiArchs.html>

⁶ siehe <http://www.eclipse.org/>

⁷ siehe <http://developer.android.com/>

Dennoch haben alle Ressourcen zwei Attribute gemein. Sie alle besitzen einen Namen und mindestens einen Facettennamen, dem sie zugeordnet sind. Um einen Überblick zu geben, was für Attribute gesetzt werden können, werden nun einige aufgezählt und vorgestellt.

Eine POI kennzeichnet sich an Hand einer *Geokoordinate* aus, welche neben der *Distanz* zum Suchort gesetzt werden kann. Da eine POI auf der Karte dargestellt werden kann, wird in der Ressource das zugehörige *Symbol* festgelegt. In manchen Fällen kann eine POI eine *Adresse* besitzen und wenn die POI eine Veranstaltung ist, dann sind der *Start-* und *Endzeitpunkt* interessant. Eine Ressource kann auch eine Person darstellen. In dem Fall ist es möglich, der Ressource einen *Geburtstag*, *Geburtsort*, *Todestag* und *Todesort* zuzuteilen. Unabhängig vom Typ der Ressource kann diese um eine *Homepage*, eine kurze *Zusammenfassung* oder eine *Fotoadresse* ergänzt werden.

FacetTreeNode

Um die Taxonomie der Facetten zu repräsentieren kann mittels der Klasse **FacetTreeNode** ein entsprechender Baum aufgebaut werden (siehe auch Abschnitt 2.1.4 für die verwendete Taxonomie). Dazu werden von einer Wurzelfacette ausgehend weitere Facetten dem Baum hinzugefügt. Um das Hinzufügen umzusetzen, werden von der Klasse Methoden angeboten, die dem **FacetTreeNode** eine Facette und einen Elternknoten zuordnen. Passend hierzu ist auch das Entfernen eines Kindknotens implementiert. Weiter ist es möglich, zu einem Knoten ein bestimmtes Kind an Hand eines Facettennamens oder alle Kinder zurückzugeben. Außerdem macht es Sinn, die Tiefe eines Knotens im Baum für das Navigationsfeld anbieten zu können.

4.2.2 Benutzeroberfläche

MFacets orientiert sich in seiner Benutzeroberfläche an FaThumb, welche aber, wie man in Abschnitt 3.3.2 sieht, nicht exakt übernommen werden kann. Dies liegt insbesondere daran, dass auf Grund des geschlossenen Datensatzes in FaThumb jede Facette maximal 8 Subfacetten besitzt, was aber für MFacets nicht vorausgesetzt werden kann. Deshalb hat man sich entschlossen, vor Beginn der Implementation Mockups zu entwerfen, die das Design von MFacets wiedergeben sollen. Nach dem Entwurf einer Mockup-Version von MFacets sind diese auf Unzulänglichkeiten analysiert worden und in einer neuen Version verbessert worden. Insgesamt sind drei Versionen verschiedener Situationen in der MFacets-Anwendung entstanden. Da sich die zweite und dritte Version jedoch sehr ähneln, werden im Folgenden nur die erste und dritte Version präsentiert, bevor die Implementation vorgestellt wird. Die erwähnten Mockups sind mit GIMP 2.6.7⁸ erstellt worden.

⁸ siehe <http://www.gimp.org/>

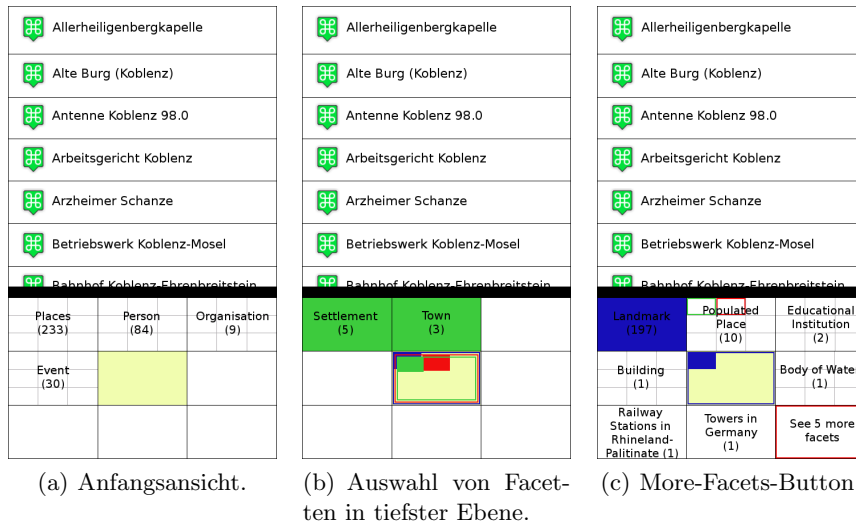


Abb. 4.3. Erste Version an Mockups (1).

Erste Version der Mockups

Abb. 4.3(a) zeigt die Anfangsansicht nach initialer Suche. Es wird hier deutlich, dass die Hauptaspekte von FaThumb übernommen worden sind, nämlich die Teilung des Bildschirms in Ergebnisliste und Facettennavigation und eben jene neuartige Navigation durch die Facetten in einem 9er-Grid. Die Facetten sind wie in FaThumb so gestaltet, dass neben dem Facettennamen die zugehörige Anzahl an Ressourcen in runden Klammern angegeben wird. Facetten mit Subfacetten, in Abb. 4.3(a) alle Facetten, werden auch in MFacets mit einem Gitter im Hintergrund und Blattknoten, z.B. *Towers in Germany* in Abb. 4.3(c), mit neutralen Hintergrund dargestellt. Sind in einer Ebene weniger als acht Facetten anzuzeigen, so werden nicht verwendete Felder des 9er-Grids leer wiedergegeben. In der Mitte des 9er-Grids ist das Navigationsfeld zu sehen, welches in Abb. 4.3(b) den gewählten Pfad widerspiegelt. Wird eine Facette ausgewählt, so verfärbt sich dessen Hintergrund gemäß der Hierarchieebene der Facette, wie man in Abb. 4.3(b) und 4.3(c) sehen kann. In letzterer Abbildung erkennt man, dass die Auswahl einer Subfacette im Hintergrund einer Superfacette im entsprechenden Feld signalisiert wird. Auch hier wird die unterschiedliche Hierarchieebene der ausgewählten Subfacette durch die Verwendung unterschiedlicher Farben gekennzeichnet.

Wie das Problem von mehr als acht Facetten in einer Hierarchieebene gelöst wird, kann in den Abbildungen 4.3(c) und 4.4(a) eingesehen werden. Wenn dieser Fall eintritt, werden im 9er-Grid nur sieben Facetten dargestellt und das letzte Feld unten rechts dient als Button zu einer neuen, sogenannten *More-Facets-Ansicht*. Der *More-Facets-Button* gibt hierzu an, wie viele

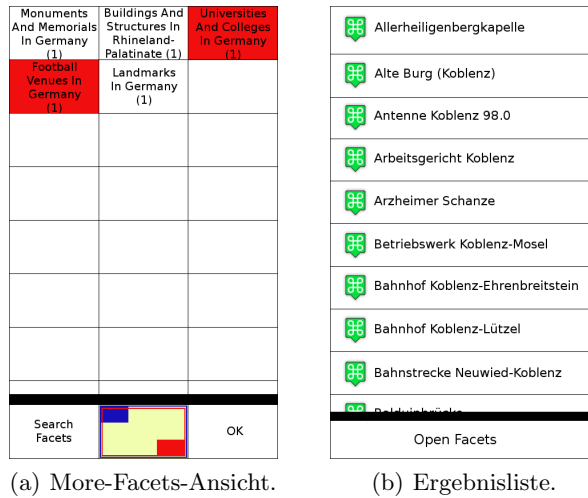
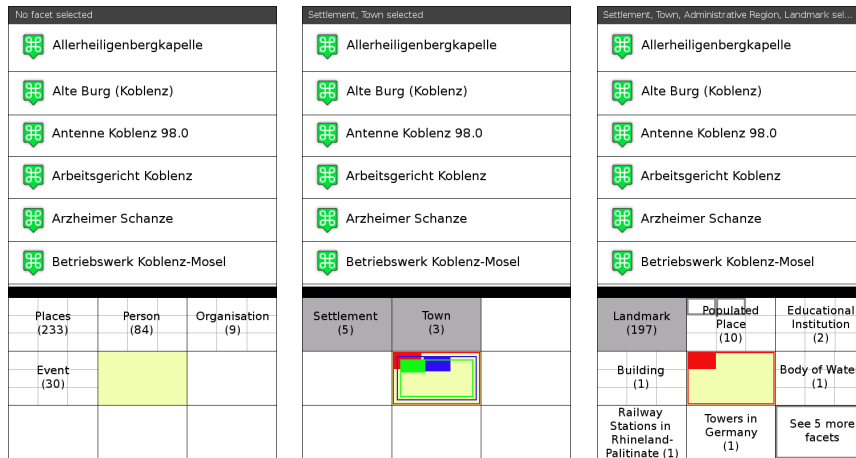


Abb. 4.4. Erste Version an Mockups (2).

Facetten in der zu öffnenden Ansicht angezeigt werden. Zur More-Facets-Ansicht hat es zwei Überlegungen zur Umsetzung und Anzeige der übrigen Facetten gegeben. Da die Anzahl der darzustellenden Facetten in dieser Ansicht im Vorhinein unbekannt ist, bietet sich ein scrollbares Element zur Darstellung an. Die Überlegung ist nun gewesen, ob man hierzu eine normale Liste wie in MobileFacets oder eine Variante mit drei Feldern pro Zeile ähnlich zur Hauptansicht verwenden soll. Auf Grund der Einheitlichkeit und der möglichen Anzeige von drei Mal mehr Facetten hat man sich für letztere Variante entschlossen. Um die Einheitlichkeit zu verdeutlichen, hat man sich dazu entschieden, das Navigationsfeld auch in der More-Facets-Ansicht anzuzeigen. Um einen Bruch beim Scrollen zu vermeiden, wird das Navigationsfeld nicht in der Mitte, sondern am unteren Ende des Bildschirms platziert. Um die Lücken neben dem Navigationsfeld zu füllen, hat man sich entschieden eine Suche nach Facetten anzubieten und über einen Button ein Zurückspringen zur Hauptansicht zu signalisieren. Ist eine Facette in der More-Facets-Ansicht ausgewählt worden, so wird eine Auswahl auch im More-Facets-Button über einen hinzugefügten Rahmen angezeigt (siehe Abb. 4.3(c)).

Zuletzt wird in Abb. 4.4(b) noch die Expansion der Ergebnisliste illustriert, welche durch einen Klick auf die Selbige erzielt werden kann. Diese Ergebnisliste ähnelt der Ergebnisliste in MobileFacets, doch um mehr Ressourcen ohne Scrollen anzuzeigen, insbesondere ohne vorherige Expansion, wird eine kompaktere Darstellung ohne Ausschnitt der Beschreibung der Ressource gewählt. Um von der expandierten Ergebnisliste wieder zur Ursprungsansicht zu gelangen, wird ein entsprechender Button angeboten.



(a) Anfangsansicht mit Facetteninformationsleiste. (b) Auswahl von Facetten in tiefster Ebene. (c) More-Facets-Button.

Abb. 4.5. Letzte Version an Mockups (1).

Dies beendet die Beschreibung der Mockups in der ersten Version. Als nächstes wird die letzte Version der Mockups so präsentiert, dass die Unterschiede zwischen den Versionen aufgezeigt und begründet werden.

Letzte Version der Mockups

Wenn man sich das erste Mockup in Abb. 4.5(a) anschaut, fällt auf, dass am oberen Bildschirmrand eine Leiste hinzugekommen ist, die *Facetteninformationsleiste*. In der ersten Version an Mockups hat eine Wiedergabe der ausgewählten Facetten ähnlich der Filterregion in FaThumb gefehlt, so dass ausgewählte Facetten nun dort angegeben werden. Da in dieser Abbildung noch keine Facetten ausgewählt sind, wird durch einen entsprechenden Text darauf hingewiesen, dass zumindest zukünftig ausgewählte Facetten an dieser Stelle wiedergegeben werden.

In Abb. 4.5(b) fallen im Vergleich zu Abb. 4.3(b) in der ersten Version der Mockups einige Unterschiede auf. Zum einen werden ausgewählt Facetten nicht mehr verschieden farbig, sondern nur noch grau markiert. Dies ist beschlossen worden, da dem Benutzer ohne Erläuterung die Bedeutung der unterschiedlichen Farben nicht sofort klar sein und auf Anheb für Verwirrung sorgen könnte. Außerdem reicht dem Benutzer die Information der Auswahl einer Facette für die facettierte Suche, so dass die Information der genauen Hierarchieebene einer Facette als weniger relevant angesehen werden kann. Dies wirkt sich auch auf die Markierungen im Hintergrund von Superfacetten und des More-Facets-Buttons aus, wie es in Abb. 4.5(c) ersichtlich ist. Ebenso

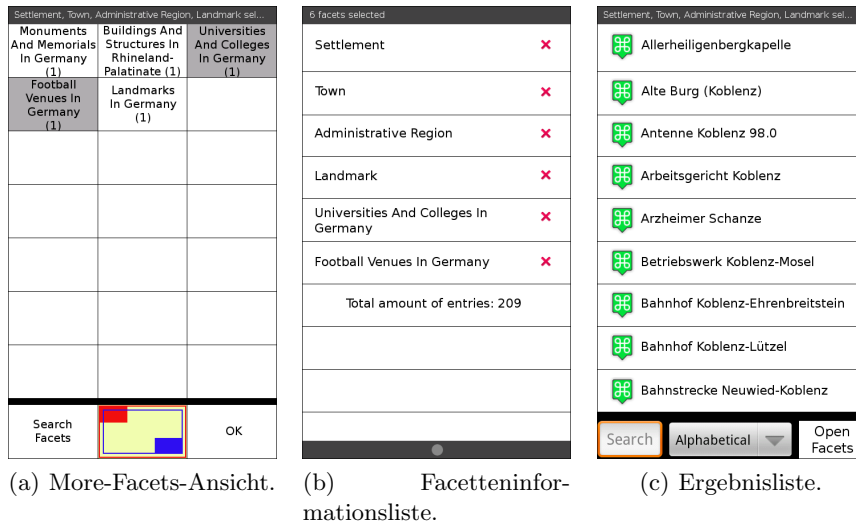


Abb. 4.6. Letzte Version an Mockups (2).

ist diese Markierung etwas breiter gestaltet, um sich mehr vom Hintergrund und Gitter abzusetzen. Weiter sieht man in Abb. 4.5(b), dass die Ebenen in der Navigationsansicht einen größeren Abstand zu einander haben und die Farbreihenfolge geändert worden ist. Der größere Abstand sorgt dafür, dass sich die Ebenen bei entsprechenden Pfaden weniger überlappen und so besser voneinander zu unterscheiden sind. Die Änderung der Farbreihenfolge ist deshalb gewählt worden, um keine grüne Ebene auf eine rote Ebene folgen zu lassen, was sich sonst für Nutzer mit einer Rot-Grün-Schwäche als Nachteil erweisen würde. Man erkennt weiterhin, dass nach Auswahl mindestens einer Facette sich der Text in der Facetteninformationsleiste ändert und die ausgewählten Facetten angibt. Werden mehr Facetten ausgewählt als es die Leiste in einer Zeile anzeigen kann, so werden durch drei Punkte am Ende der Zeile auf weitere ausgewählte Facetten hingewiesen, wie es auch in Abb. 4.5(c) und 4.6(a) dargestellt ist. Letztere Abbildung zur More-Facets-Ansicht dient lediglich der Vollständigkeit, dass die bereits erwähnten Veränderungen auch in dieser Ansicht greifen.

Abb. 4.6(b) zeigt einen in der letzten Version der Mockups hinzugekommenen Bearbeitungsbildschirm, nämlich die *Facetteninformationsliste*. Wenn man die zugehörige Leiste hinunterzieht, so erscheint diese Liste mit allen ausgewählten Facetten. Somit wird sichergestellt, dass der Benutzer auch die ausgewählten Facetten sehen kann, die eventuell in der Leiste nicht mehr angezeigt werden können. Außerdem wird in der Liste im letzten Eintrag angegeben, wie viele Ressourcen unter Berücksichtigung der ausgewählten Facetten in der Ergebnisliste angezeigt werden. Weiter ist neben dem Facettennamen ein *Löschen-Button* dargestellt, um ein Entfernen einer Facette zu

ermöglichen. Somit wird dem Benutzer neben dem direkten Abwählen einer Facette eine Alternative geboten, um zum Löschen nicht zur Facette navigieren zu müssen. Wenn man die Facetteninformationsleiste wieder hochzieht, so verschwindet dieser Bearbeitungsbildschirm wieder.

In der letzten Abbildung 4.6(c) der Mockups lassen sich zwei Unterschiede zur ersten Version ausmachen. Einerseits ist es nun möglich die Ressourcen zu filtern und andererseits zu sortieren. Diese Eingabefelder sind direkt aus MobileFacets entnommen und werden neben dem Button zum Öffnen der Facettenavigation am unteren Ende des Bildschirms platziert.

Dies schließt die Entwurfsphase mit der Erstellung von Mockups ab, so dass nun näher auf die Implementation der Benutzeroberfläche eingegangen werden kann.

Implementation der Benutzeroberfläche

Da MFacets auf einem Google Android Betriebssystem laufen wird, werden zur Erstellung der Benutzeroberfläche entsprechende Werkzeuge des Android SDK verwendet. Um die Benutzeroberfläche zu erstellen, bietet das Android SDK eine entsprechende Ansicht in Eclipse an, um mit einfachen Klicks ein Layout zusammen zu stellen. Dieses Layout mit all seinen Elementen wird in einer oder mehreren XML⁹-Dateien gespeichert. Um einzelne Layout-Elemente in Methoden aufrufen und verändern zu können, kann jedem Element eine ID zugeteilt werden, die über die Android spezifische Klasse `R` erreichbar ist. In Android wird ein Bearbeitungsbildschirm in der Regel durch eine `Activity` repräsentiert, die wiederum mit einer Layout-XML-Datei verbunden ist. Um zwischen Activities zu wechseln, verwendet man `Intents` und die einzelnen Layout-Elemente basieren auf der Klasse `View`.

In MFacets gibt es verschiedene Activities. Die meisten davon sind aus MobileFacets übernommen und zwei Activities sind speziell für MFacets entwickelt worden. Da die einzelnen Activities aus MobileFacets im Detail in [Kle10] beschrieben sind, werden diese hier nur in einem Überblick wiedergegeben:

DatePickerActivity Diese Activity ermöglicht es, das Datum oben links in der Karte manuell zu verändern.

GalleryActivity Die Galerie sammelt alle aus Flickr erhaltende Fotos und bildet sie ab. Über dem angezeigten Foto werden alle Fotos in einer Miniaturansicht dargestellt zwischen denen der Benutzer scrollen kann.

HelpActivity Die `HelpActivity` wird in der Kartenansicht über das Menü aufgerufen und erklärt die Bedeutung der verschiedenen Symbole der POIs.

⁹ Extensible Markup Language

InteractiveMapActivity Diese Activity ermöglicht mit Hilfe der `InteractiveMapView` die Verwendung der Karte. Sie regelt, welche Elemente wo auf der Karte dargestellt werden und bietet spezielle Kartenfunktionalitäten, wie z.B. das Zoomen, an.

PreferencesActivity Mit dieser Activity lassen sich Einstellungen für die Anwendung angeben. Die Einstellungsseite lässt sich über das Menü aufrufen und man kann auswählen, welche Quellen durchsucht und wie viele Bilder maximal von Flickr erhalten werden sollen. Die möglichen Einstellungen werden in einer separaten XML-Datei angegeben.

ResourceInspectorActivity Um Details einer Ressource aufzurufen, wird diese Activity benötigt. Zum einen stellt sie die Informationen zu einer Ressource übersichtlich dar und falls eine Ressource eine POI darstellt, dann ermöglicht sie zum anderen auch einen Sprung zur POI auf der Karte.

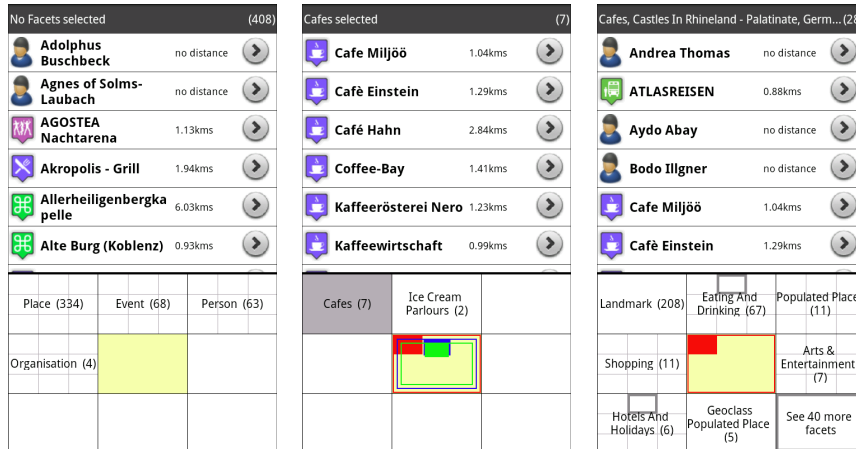
SearchActivity Die `SearchActivity` ermöglicht eine Suche nach Stichwörtern und wird über die *Suchen-Taste* des Smartphones geöffnet. Neben dem Ausführen der eigentlichen Suche werden dem Benutzer nach Eingabe eines Stichworts auch mögliche weitere Stichwörter angeboten, welche ebenso angewählt werden können.

WebViewActivity Besitzt eine Ressource einen Link zu einer Homepage, so kann diese in einer einfachen Ansicht innerhalb der Anwendung über diese Activity angesehen werden.

Dies sind alle Activities, die aus `MobileFacets` übernommen und gegebenenfalls etwas angepasst worden sind. Als nächstes werden nun die beiden `MFacets` spezifischen Activities vorgestellt.

StartActivity

Die `StartActivity` ist eine unsichtbare Activity und dient als Einstiegspunkt für die Anwendung. Ihr Zweck ist es, die Initialisierung und Startreihenfolge der notwendigen Activities zu regeln. Um die initiale Ortssuche der Anwendung ausführen zu können, ist es vonnöten, dass die Geokoordinaten des Kartenausschnitts zu ermitteln. Dazu wird aus dieser Activity die `InteractiveMapActivity` per Intent gestartet und nach Initialisierung wieder aufgerufen. Danach wird die `MFacetsActivity` zur Hauptansicht gestartet. Neben diesen Aufgaben ermittelt die `StartActivity` auch die Displaygröße des Smartphones, welche zur richtigen Darstellung der Facettenavigation nötig ist. Da es in Android notwendig ist, eine Activity zum Starten anzugeben, wird die `StartActivity` als solche an entsprechender Stelle deklariert.



(a) Anfangsansicht. (b) Auswahl von Facetten in tiefster Ebene. (c) More-Facets-Button.

Abb. 4.7. Screenshots von MFacets (1).

MFacetsActivity

Die `MFacetsActivity` stellt die Hauptansicht von MFacets dar und setzt die in den Mockups entworfenen Bearbeitungsbildschirme um. Sobald die initiale Suche abgeschlossen ist, werden die Ergebnisliste mit Ressourcen und die Facettennavigation mit Facetten gefüllt (siehe Abb. 4.7(a)). In dieser Abbildung erkennt man auch einige Unterschiede zu den Mockups. Einerseits sieht man oben rechts in der Facetteninformationsleiste ein neu hinzugekommenes Element, was zu jeder Zeit die Anzahl an Ressourcen in der Ergebnisliste wiedergibt. Andererseits werden den Ressourcen nun auch eine Entfernung zum Suchort und ein Button zum Öffnen der Detailansicht der Ressource zugeordnet. Besitzt eine Ressource keine Entfernung, z.B. wenn es sich um eine Person handelt, dann wird dies textuelle angegeben.

Die Ansicht zur Facettennavigation ist so gestaltet, dass sie $2/5$ der Bildschirmhöhe belegt. Somit besitzt eine Facette die Breite von $1/3$ der Bildschirmbreite und die Höhe von $2/15$ der Bildschirmhöhe. Da diese Dimensionen nur bei einem Bildschirm im Hochformat eine benutzerfreundliche Nutzung ermöglichen, ist die `MFacetsActivity` so definiert, dass der Bildschirminhalt für einen Bildschirm im Querformat nicht transformiert wird. Eine Facette wird mittels zwei Views realisiert, die `FacetView` und die `FacetTextView`. In Ersterer wird die Verbindung zwischen dem Benutzeroberflächenelement und dem Datenmodell der Facette hergestellt und Letztere ist für die Darstellung des Texts zuständig. Da das Android SDK keine simple Möglichkeit bietet, Text auf einem beschränkten Raum über mehrere Zeilen anzuzeigen, ist die `FacetTextView` mit Hilfe des

android-textview-multiline-ellipse-Projekts¹⁰ implementiert worden. Diese View sorgt auch dafür, dass, falls ein Facettenname länger als das zugehörige Feld ist, der Name durch drei Punkte abgekürzt wird und dennoch die Anzahl an Ressourcen weiterhin angezeigt werden. Soll ein Nicht-Blattknoten ausgewählt werden, so ist dies über einen langen Klick auf die Facette möglich. Der Hintergrund der Facette wird über eine `Drawable` implementiert, der `FacetBackgroundDrawable`. Diese regelt die richtige Darstellung des Hintergrunds einer Facette, also z.B. ob ein Gitter gemalt oder der gesamte Hintergrund markiert werden muss (siehe Abb. 4.7(b) und 4.7(c)).

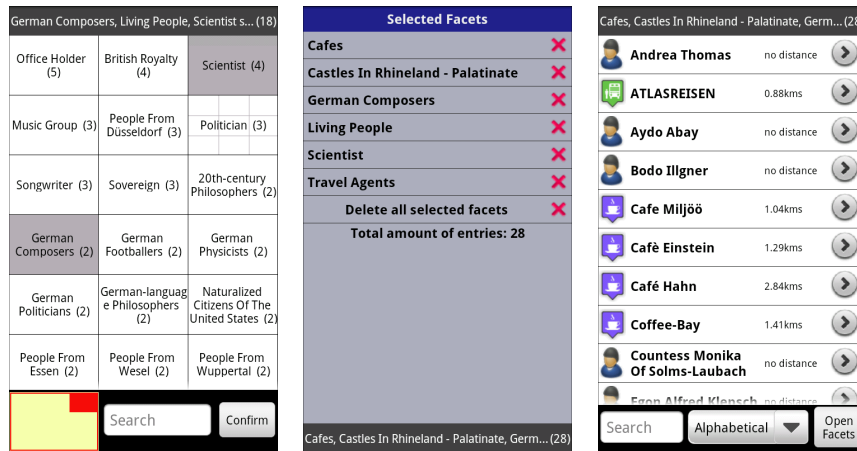
Das Navigationsfeld wird über die `NavigationView` realisiert, welche nur dazu benötigt wird, den Hintergrund mit der `NavigationBackgroundDrawable` anzeigen zu können. Mit dieser `Drawable` können die einzelnen Ebenen und die Facetten des Pfades angezeigt werden (siehe Abb. 4.7(b)). In Abb. 4.7 sieht man auch, wie die Verknüpfung der Facetten sich auf die Ergebnisliste auswirkt. Zuerst werden alle Ressourcen bei keiner ausgewählten Facette angezeigt, nach Auswahl einer Facette wird eine Schnittmenge gebildet und bei Auswahl weiterer Facetten wird die Ergebnisliste um die entsprechenden Ressourcen erweitert.

Abb. 4.8(a) zeigt die More-Facets-Ansicht, welche über den More-Facets-Button geöffnet wird. Es hat den Anschein, dass dieser Bearbeitungsbildschirm über eine weitere Activity implementiert ist, in der Tat wird jedoch nur die Sichtbarkeit von Elementen verändert. Ist man in der normalen Ansicht, so ist die More-Facets-Ansicht auf unsichtbar gestellt. Nach Klick auf den More-Facets-Button wird die normale Ansicht auf unsichtbar und die More-Facets-Ansicht auf sichtbar geändert. So entsteht ein flüssiger Wechsel zwischen diesen beiden Bearbeitungsbildschirmen. In dieser Abbildung sieht man auch einen kleinen Unterschied zum entsprechenden Mockup. Im Mockup ist das Navigationsfeld mittig am unteren Ende des Bildschirms dargestellt gewesen, nun ist es in der eigentlichen Anwendung links platziert. Dies liegt an der Texteingabe zum Filtern der Facetten, welcher in der Entwurfsphase noch als Button dargestellt gewesen ist und nun in der Anwendung so platziert mehr Text anzeigen kann.

Die in Abb. 4.8(b) illustrierte Facetteninformationsliste hat sich gegenüber der Mockup-Version etwas verändert. Um sich besser von der Ergebnisliste abzuheben, ist ein farbiger Hintergrund für die Liste gewählt worden. Außerdem ist nun auch ein Eintrag hinzugefügt worden, der das Löschen aller Facetten ermöglicht. Die Liste wird weiterhin durch Ziehen der Facetteninformationsleiste erreicht. Da das Android SDK keine triviale Möglichkeit zum Ziehen einer Leiste von oben nach unten bietet, ist dies mit Hilfe des *android-misc-widgets*-Projekts¹¹ implementiert worden.

¹⁰ siehe <http://code.google.com/p/android-textview-multiline-ellipse/>

¹¹ siehe <http://code.google.com/p/android-misc-widgets/>



(a) More-Facets-Ansicht. (b) Facetteninformationsliste. (c) Ergebnisliste.

Abb. 4.8. Screenshots von MFacets (2).

Der Vollständigkeit halber wird in Abb. 4.8(c) auch die realisierte Ergebnisliste verbildlicht. Wie die Anzeige der More-Facets-Ansicht ist auch die Expansion der Ergebnisliste über die Veränderung der Sichtbarkeit entsprechender Elemente umgesetzt.

Hiermit wird die Beschreibung der Benutzeroberfläche von MFacets abgeschlossen. Somit folgt im nächsten Unterabschnitt die Erläuterung der Programmsteuerung.

4.2.3 Programmsteuerung

Die Programmsteuerung hat die Aufgabe, Benutzerinteraktionen zu verarbeiten, Abläufe zu regeln, das Datenmodell anzureichern und die Kommunikation mit dem Server abzuwickeln. Einige Funktionalitäten hierzu sind aus Mobile-Facets entnommen und angepasst worden, wie z.B. das Auslesen der Antwort des Servers, andere hingegen sind hinzugekommen, wie z.B. die Bereitstellung der Subfacetten zu einer Facette für die Facettenavigation. Um diese Funktionalitäten übersichtlich zu gliedern, ist die Programmsteuerung in drei Manager unterteilt, dem `ActivitiesManager`, dem `FacetManager` und dem `ResourceManager`. Diese Manager werden nun vorgestellt.

ActivitiesManager

Der `ActivitiesManager` hat die grundlegenden Aufgaben, dass die notwendigen `Activities` in der richtigen Reihenfolge initialisiert werden. Somit wird sicher gestellt, dass Methoden anderer Klassen auf benötigte Werte zugreifen können ohne Gefahr zu laufen, dass diese wegen fehlender

Initialisierung noch nicht vorhanden sind. Außerdem werden in diesem Manager alle gestarteten Activities gesammelt, so dass beim Schließen der Anwendung auch alle Activities geschlossen werden.

FacetManager

Wie der Name schon vermuten lässt, werden im **FacetManager** Methoden mit Zusammenhang zu Facetten angeboten. Noch vor der initialen Suche wird der **FacetManager** benötigt, um vom Server die Facettenstruktur zu erhalten. Die Facettenstruktur ist deshalb notwendig, um die Antwort des Servers bezüglich der Bereitstellung von Ressourcen möglichst gering zu halten. Würde man der Anwendung die Struktur nicht separat mitteilen, so müsste bei einer Ressource nicht nur die zugeordneten Facetten, sondern auch der Pfad jeder dieser Facetten zur Wurzel angegeben werden. Um diesen Mehraufwand und mögliche Redundanzen zu vermeiden, hat man sich für diese Lösung entschieden. Die Facettenstruktur wird vom Server in Form eines JSON¹²-Strings gesendet, da JSON eine einfache Möglichkeit bietet mit geringem Aufwand strukturierte Daten zu erstellen und zu verarbeiten. Da für die Facettenstruktur nur ein Baum benötigt wird, bietet sich diese Variante durchaus an. JSON wird von JAVA nicht unterstützt, so dass zur Erstellung und Verarbeitung von JSON das JSON-Paket von Douglas Crockford¹³ verwendet wird.

Im **FacetManager** wird auch die Kommunikation zum Server hinsichtlich der Bereitstellung der Ressourcen umgesetzt. Es lässt sich eventuell darüber streiten, ob dies nicht besser im **ResourceManager** implementiert werden sollte. Da aber in MobileFacets das Konzept eines **ResourceManagers** fehlt, die Serverantwort zu den Ressourcen auch Facetten enthält und im **FacetManager** schon die Kommunikation zum Server in Bezug auf die Facettenstruktur abgewickelt wird, ist auch diese Kommunikation hier implementiert. Neben dem Aufbau der Kommunikation ist vor allem die Verarbeitung der Antwort wichtig für MFacets. Hier wird dafür gesorgt, dass Facetten und Ressourcen mit den entsprechenden Informationen erstellt werden und so zur Anzeige in MFacets zur Verfügung stehen. Um die Wartezeit zwischen Anfrage an und Antwort vom Server zu verdeutlichen, wird ein Ladebildschirm angezeigt.

Der **FacetManager** ist auch dafür zuständig, dass je nach Benutzereingabe immer die richtigen Facetten in der Facettennavigation und in der More-Facets-Ansicht angezeigt werden. Dies gilt nicht nur für die Facettennamen, sondern auch für den Hintergrund der Facetten. Daher werden im **FacetManager** die vom Benutzer ausgewählten Facetten gespeichert und an die jeweiligen Elemente der Benutzeroberfläche, wie z.B. die Facetteninformationsliste, weitergereicht. Vom **FacetManager** ist hier besonders zu beachten, dass bei Auswahl einer Facette die Hintergründe

¹² JavaScript Object Notation, siehe auch <http://json.org/>

¹³ siehe <https://github.com/douglascrockford/JSON-java>

ihrer Superfacetten bis zur Wurzel je nach Position der Facette im 9er-Grid aktualisiert werden. Eine entsprechende Aktualisierung ist auch im Fall der Abwahl einer Facette durchzuführen. Weiter wird ebenfalls der aktuelle Pfad bis zur Wurzel gespeichert, um die richtige Anzeige des Navigationsfelds zu gewährleisten.

ResourceManager

Der **ResourceManager** speichert, wie der Name schon sagt, alle Ressourcen, insbesondere die Ressourcen zu den ausgewählten Facetten. Er sorgt somit dafür, dass einerseits die Ergebnisliste und andererseits die Karte die richtigen Ressourcen erhält. Hier ist zu beachten, dass die Karte nur POIs, also nur Ressourcen mit Geokoordinate, erhält. Außerdem werden im **ResourceManager** die Zuordnungen von Symbol und Ressource erstellt. Um dies zu erreichen, werden Symbole zu bestimmten Facetten fest zugeteilt, so dass an Hand der Facetten einer Ressource die Ressource ein Symbol bekommt. Existiert keine Zuordnung zwischen einer Facette und einem Symbol, so wird der Vorgang so lange mit der Superfacette wiederholt bis ein Symbol gefunden ist. Da mit der Hinzunahme von Qype zu den Datenrepositories auch weitere interessante Facetten hinzugekommen sind, werden die Symbole aus *MobileFacets* dementsprechend ergänzt. Diese Symbole sind der *map icons collection*¹⁴ entnommen.

Dies beendet nicht nur die Beschreibung der Umsetzung des Model-View-Controller-Prinzips in *MFacets*, sondern auch die Beschreibung des Clients selbst. Im nächsten Abschnitt gilt es nun, die Beschreibung des *MFacets*-Systems zu komplettieren und die Serverseite vorzustellen.

4.3 Server

Wie man in Abb. 4.1 am Anfang des Kapitels sehen kann, ist der Server eine wichtige Komponente im *MFacets*-Systems. Der Server erhält und verarbeitet Anfragen von den Clients, sorgt für die Aufbereitung von Ressourcen aus den Datenrepositories und gibt die relevanten Informationen an die Clients weiter. Realisiert wird dies durch RESTful Webservices auf einem Tomcat-Server. In Abb. 4.9 erkennt man, dass es sich um genau zwei Webservices handelt. Einer dient der Erstellung der Facettenstruktur und einer zur Bereitstellung der Ressourcen. Die Facettenstruktur wird deshalb auf dem Server, welcher unter <http://mfacets.west.uni-koblenz.de:8080/MFacets-Server/> zu erreichen ist, erstellt, so dass bei einer Änderung dieser keine Aktualisierung der Anwendung notwendig ist. Wie diese erstellt wird und wie der Server die Ressourcen aus den Datenrepositories erhält, wird nach der Vorstellung der Datenrepositories in den folgenden Unterabschnitten gezeigt.

¹⁴ siehe <http://mapicons.nicolasmollet.com/>

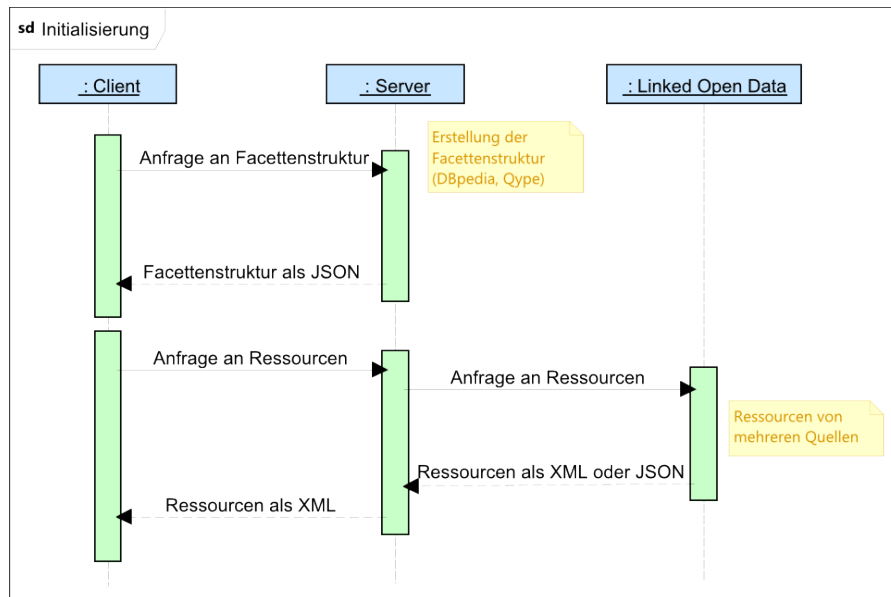


Abb. 4.9. Sequenzdiagramm der Kommunikation zwischen Client, Server und den Datenrepositories.

4.3.1 Datenrepositories

In Abschnitt 3.2 zur Beschreibung von MobileFacets ist schon erwähnt worden, dass DBpedia, GeoNames, Eventful, Upcoming und Flickr als Datenrepositories verwendet werden. Für das MFacets-System wird Qype denen hinzugefügt. Dieser Unterabschnitt soll nun einen Überblick über diese Datenrepositories schaffen und sie kurz vorstellen.

DBpedia

DBpedia¹⁵ ist das Produkt einer Gemeinschaft, die sich das Ziel gesetzt hat, strukturierte Informationen aus Wikipedia¹⁶ zu extrahieren und diese Informationen über das Internet anzubieten. Diese Informationen sind nicht nur strukturiert, um gezielt Anfragen nach Informationen ausführen zu können, sondern auch zu weiterführenden Wikipedia-Artikeln verlinkt. Die Daten, die von DBpedia angeboten werden, sind in RDF¹⁷-Tripeln aufbereitet und bieten sich somit besonders für das semantische Web an. Um einen besseren Überblick über den DBpedia-Datensatz zu erhalten, werden nun kurz ein paar Daten dazu aufgelistet:

¹⁵ siehe <http://wiki.dbpedia.org/About>

¹⁶ siehe <http://wikipedia.org/>

¹⁷ Resource Description Framework, siehe auch <http://www.w3.org/RDF/>

- 672 Millionen RDF-Tripel, 286 Millionen RDF-Tripel aus der englischen Wikipedia
- mehr als 3,5 Millionen „Dinge“
- 364.000 Personen, 462.000 Orte, 99.000 Musikalben, 54.000 Filme, 16.500 Videospiele, 148.000 Organisationen, 148.000 Spezien und 5.200 Krankheiten
- 1.850.000 Verweise zu Bildern, 5.900.000 Verweise zu externen Homepages, 6.500.000 externe Verweise zu anderen RDF Datensätzen und 632.000 Wikipedia Kategorien

DBpedia ist das einzige Datenrepository, welches über eine HTTP-Schnittstelle per SPARQL angefragt wird.

GeoNames

GeoNames¹⁸ ist eine Datenbank mit Informationen zu geographischen Orten. Im Zusammenhang des MFacets-Systems dient GeoNames der Zuordnung von Geokoordinaten zu Ortsnamen. Diese Ortsnamen werden für DBpedia verwendet, um Ressourcen zu den gefundenen Orten zu finden. So können neben Ressourcen zum Ort selbst auch Ressourcen mit Verbindung zum Ort, wie z.B. Personen mit entsprechenden Geburtsort, gefunden werden. Der Datensatz von GeoNames beinhaltet mehr als 10 Millionen geographische Namen, 2,8 Millionen bewohnte Orte und 5,5 Millionen Alternativname. Zur Anreicherung dieses Datensatzes beruht sich GeoNames auf verschiedenste Quellen, welcher für den Server per REST angefragt werden kann.

Eventful

Eventful¹⁹ hat das Ziel, Benutzern möglichst viele Veranstaltung in der Nähe anzubieten. Dazu ist es für den Nutzer möglich, Veranstaltungen zu entdecken, zu teilen oder zu erstellen. Laut eigenen Angaben besitzt Eventful 18 Millionen registrierte Benutzer und pro Monat durchstöbern mehr als 20 Millionen Benutzer mehr als 4 Millionen lokale Veranstaltungen. Als Veranstaltungen gelten unter anderem Filmvorstellungen, Musikkonzerte und Veranstaltungen zu Sport oder für die Familie. Der Webservice von Eventful wird vom Server per REST verwendet um an Ressourcen zu gelangen.

Upcoming

Mit Upcoming²⁰ steht dem Server ein weiterer Datenrepository mit Bezug zu Veranstaltungen zur Verfügung. Upcoming ist von Yahoo!²¹ erworben worden

¹⁸ siehe <http://www.geonames.org/>

¹⁹ siehe <http://eventful.com/>

²⁰ siehe <http://upcoming.yahoo.com/>

²¹ siehe <http://yahoo.com/>

und lebt wie Eventful von Nutzer generiertem Inhalt. Da die in Upcoming erstellten Veranstaltungen nach gewissen Kategorien unterschieden werden, kann dies für die facettierte Suche in MFacets genutzt werden. Auch Upcoming wird vom Server per REST angesprochen um weitere Ressourcen zu Veranstaltungen zu erhalten.

Flickr

Flickr²² ist ein weiterer von Yahoo! erworbener Dienst, der es Nutzern ermöglicht, digitale Fotos und Videos hochzuladen und diese mit anderen Nutzern zu teilen. Die Gemeinschaft von Flickr umfasst 51 Millionen registrierte Nutzer, welche pro Tag 4,5 Millionen Fotos hochladen²³. Besucht wird Flickr von knapp 80 Millionen einzigartigen Benutzern, wovon 20 Millionen aus den USA stammen. Im Gegensatz zu den anderen Datenrepositories wird Flickr nicht vom Server, sondern direkt von der Anwendung per REST angesprochen. Dies ist schon so in MobileFacets umgesetzt worden, da die Anwendung die Fotos selbst herunterlädt und somit die Kommunikation über den MFacets-Server einen Umweg darstellen würde.

Qype

Mit Qype²⁴ ist im MFacets-System ein neues Datenrepository hinzugekommen. Qype hat das Ziel, über Kritiken und Bewertungen den Nutzern Empfehlungen zu jeglichen Einrichtungen anzubieten und so zu einem besseren Kundendienst zu führen. Zu diesen Einrichtungen gehören unter anderem Restaurants, Dienstleistungen, Ärzte oder Behörden, welche von Nutzern bewertet werden. Laut eigenen Angaben wird in Qype alle 32 Sekunden eine Bewertung abgegeben, so dass Qype auf über 20 Millionen Besucher im Monat kommt. Für MFacets ist es von Vorteil, dass die bewerteten Einrichtungen zum einen gut strukturiert erfasst sind und diese zum anderen über Geokoordinaten per REST angefragt werden können.

4.3.2 Webservice zur Facettenstruktur

Der Webservice zur Facettenstruktur hat im MobileFacets-System noch nicht existiert, so dass dieser neu zu entwickeln ist. In MobileFacets werden die Facetten, die angezeigt werden können, fest in eine Klasse geschrieben. Dies hat den Nachteil, dass, falls dies nicht gründlich gemacht wird, existente Facetten nicht berücksichtigt werden und dass es bei einer Erweiterung oder Änderung der Facetten zu großen Änderungen in besagter Klasse führen kann. Da für

²² siehe <http://www.flickr.com/>

²³ siehe <http://advertising.yahoo.com/article/flickr.html>

²⁴ siehe <http://www.qype.com/>

MFacets beschlossen worden ist, Qype als weitere Quelle zu den Datenrepositories hinzuzufügen, hat man sich dazu entschlossen, die Facettenstruktur in der Anwendung dynamisch generieren zu lassen. Hierzu ist etwas Vorarbeit zu leisten, indem man die verwendete Facettenstruktur der Datenrepositories untersucht. Bei näherer Betrachtung stellt man fest, dass sowohl Eventful als auch Upcoming keine Facettenstruktur besitzen und die Kategorien der Veranstaltungen alle auf einer Ebene anzusehen sind. Da aus GeoNames keine Ressourcen ermittelt werden, gilt es nur noch DBpedia und Qype genauer zu analysieren.

DBpedia bietet die Möglichkeit, das verwendete Schema als Ontologie²⁵ in OWL²⁶ herunterzuladen. Diese OWL-Datei kann analysiert werden, so dass man die Facettenstruktur erhält. Um diese Analyse zu vereinfachen, wird ein `OWL2FSParser` des `TwoUse-Toolkits`²⁷ verwendet. Allerdings muss für diesen Parser die Ontologie in einer gewissen Syntax vorliegen, der OWL 2 Functional-Style Syntax²⁸. Eine Transformation zu dieser Syntax ist z.B. mit dem Programm `Protégé`²⁹ möglich. Zu dieser Extraktion der Facettenstruktur aus der Ontologie wird das zugehörige Schema in der Version 3.6 verwendet.

Qype bietet im Gegensatz zu DBpedia keine Ontologie oder ähnliches um die verwendete Facettenstruktur zu erlangen. Stattdessen gibt es die Möglichkeit, alle Facetten über die Qype API³⁰ zu erhalten. Allerdings erhält man über die API nicht die komplette Facettenstruktur auf Anhieb, sondern man muss sie aufbauen. Hierzu ruft man die Wurzelfacette auf, welche Informationen zu den Kindern erhält. Diese Information speichert man und ruft daraufhin die einzelnen Kinder auf und wiederholt diesen Prozess bis die komplette Struktur aufgebaut worden ist. Um den Aufbau dieser Facettenstruktur nicht für jede Client-Anfrage neu zu generieren, geschieht dies ein Mal und die aufgebaute Facettenstruktur wird in einer Datei auf dem Server gespeichert.

Diese beiden Facettenstrukturen liegen dem Webservice im JSON-Format vor. Da Qype nur Orte als Ressourcen besitzt, wird die Qype Facettenstruktur der Facette *Place* aus der DBpedia Facettenstruktur untergeordnet. Dieses zusammengefügte JSON-Dokument wird nach Aufruf des Webservices über `http://mfacets.west.uni-koblenz.de:8080/MFacets-Server/RestFacetStructure` dem Client als Antwort geliefert.

²⁵ Eine Ontologie kann in diesem Zusammenhang als eine Art Wissensbasis verstanden werden. Für weitere Information siehe <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

²⁶ Web Ontology Language, siehe auch <http://www.w3.org/TR/owl2-overview/>

²⁷ siehe <http://code.google.com/p/twouse/>

²⁸ siehe <http://www.w3.org/TR/owl-syntax/>

²⁹ siehe <http://protege.stanford.edu/>

³⁰ Application Programming Interface, zu deutsch Programmierschnittstelle

4.3.3 Webservice zur Bereitstellung der Ressourcen

Der Webservice zur Bereitstellung der Ressourcen auf dem MFacets-Server unterscheidet sich nur geringfügig vom selbigen auf dem MobileFacets-Server, so dass der Ablauf der Anfragen an die Datenrepositories nur um die Anfrage an Qype erweitert wird. Da die Funktionsweise detailliert in [Kle10] beschrieben ist, wird hier ein Überblick über den Ablauf gegeben und nur die Anfrage an Qype genauer beschrieben.

Ablauf

Um den Webservice zur Bereitstellung der Ressourcen³¹ richtig verwenden zu können, müssen bestimmte Parameter per HTTP GET übergeben werden. Um einen Missbrauch des Webservices zu verhindern, muss für jede Anfrage der Parameter `m` gesetzt sein. `m` kann hier den Wert 1 oder 2 annehmen, wodurch unterschieden wird, ob eine Stichwort- oder Ortssuche durchgeführt werden soll. Außerdem sollten die Parameter `search-dbpedia`, `search-events` und `search-qype` auf `true` gesetzt sein, um die entsprechenden Datenrepositories in der Suche einzuschließen.

Wird eine Stichwortsuche ausgeführt, so können zwei Fälle eintreten. Entweder wird das Stichwort, welches per `keyword` übergeben wird, als Ort mit Geokoordinate unter Zuhilfenahme von GeoNames erkannt oder nicht. Im ersten Fall wird eine Ortssuche mit der ermittelten Geokoordinate ausgeführt, ansonsten werden DBpedia, Eventful und Upcoming mittels des Stichworts angefragt. Qype wird vom Server nicht angefragt, da Qype eine Stichwortsuche nur für Nutzer eines Pro-Accounts zulässt.

Wird eine Ortssuche ausgeführt, so sind weitere Parameter notwendig. Per `north`, `south`, `west` und `east` werden die Geokoordinaten übergeben, um den Kartenausschnitt der Anwendung zu repräsentieren. Diese Parameter können für DBpedia und Qype für eine Bereichssuche genutzt werden, wobei für DBpedia wiederum GeoNames zur Ermittlung von Ortsnamen verwendet wird. Eventful und Upcoming unterstützen keine Bereichssuche, sondern nur eine Umkreissuche. Hierzu wird der Mittelpunkt des Kartenausschnitts mit `long` und `lat` und der Durchmesser des Umkreises in Kilometern mit `distance` übergeben.

Nach Erhalt der Antworten der Datenrepositories werden diese so aufbereitet, dass nur eine einzelne Antwort an den Client geschickt werden muss. Wie eine Anfrage an ein Datenrepository abläuft, wird im Folgenden beispielhaft an Qype gezeigt.

Anfrage an Qype und Erstellung der Antwort

Mit Ausnahme von DBpedia, welches mit SPARQL über eine HTTP-Schnittstelle angesprochen wird, werden alle Datenrepositories per REST angefragt.

³¹ siehe <http://mfacets.west.uni-koblenz.de:8080/MFacets-Server/RestResourceProvider>

Das bedeutet, dass auf den Datenrepositories über eine bestimmte Webadresse mit spezifizierten Parametern eine Suche ausgeführt werden kann. Im Fall von Qype lautet die Adresse `http://api.qype.com/v1/bounding_boxes/`, woran die Beschreibung des Bereichs durch die vom Client erhaltenen Parameter `south`, `west`, `north` und `east` folgt. Zuletzt benötigt Qype noch Parameter zur Identifizierung des Nutzers (`consumer_key`), für die Sprache (`lang`), welche auf Englisch gestellt ist, und zur Angabe an maximal zu erhaltenden Ressourcen (`per_page`). Diese Anzahl wird mit 100 auf das von Qype unterstützte Maximum gesetzt.

Nach Ausführung dieser Suche wird vom Datenrepository eine Antwort an den Server gesendet. Im Fall von GeoNames, Eventful und Upcoming ist diese Antwort ein XML-Dokument, im Fall von Qype ein JSON-Dokument, welche beide als gleichwertig, aber mit unterschiedlicher Formatierung angesehen werden können. Die Aufgabe des Servers ist es nun, diese Antworten zu analysieren und nur die Informationen temporär zu speichern, die vom Client benötigt werden. Hierzu wird pro erhaltener Ressource eine Ressource im Datenmodell abgebildet, im Fall von Qype ist dies ein `QypeWrapper`. Nachdem alle Antworten vom Server erhalten worden sind, wird die Antwort in Form eines XML-Dokuments an den Client verfasst. Diese Antwort enthält nur noch die notwendigen Informationen der Ressourcen, die allesamt vom Client verstanden werden und so als Grundlage für die facettierte Suche dienen können.

Hiermit wird die Beschreibung des Servers und somit des kompletten MFacets-Systems abgeschlossen. Nun sind die beiden mobilen Anwendungen MFacets und MobileFacets prinzipiell bereit für die Evaluation. Dennoch müssen diese Anwendungen noch unter gewissen Gesichtspunkten vorbereitet werden. Welche dies sind und was genau geändert werden muss, ist Bestandteil des nächsten Abschnitts.

4.4 Vorbereitung der Anwendungen auf die Evaluation

Um eine erfolgreiche Evaluation zu gewährleisten, müssen die mobilen Anwendungen so vorbereitet werden, dass die Probanden bei beiden Anwendungen die gleichen Voraussetzungen auffinden. Weiter müssen die Anwendungen auch in der Lage sein, den Anforderungen an die Evaluation zu genügen. Daher wird im Folgenden einerseits aufgezeigt, welche Änderungen an den Anwendungen vorgenommen, und andererseits mit welchen Funktionalitäten sie erweitert worden sind.

4.4.1 Änderungen an den Anwendungen

Die Evaluation zwischen MobileFacets und MFacets setzt voraus, dass beide Anwendungen vergleichbar sind. In 5.1.1 wird gezeigt, dass zwei Aspekte der

Anwendungen verglichen werden. Diese sind zum einen der Ansatz der initial vollen Ergebnisliste von MFacets gegenüber der leeren Ergebnisliste von MobileFacets und zum anderen die unterschiedliche Darstellung und Navigation in den Facetten. Deshalb müssen sonstige unterschiedliche Funktionalitäten entweder beseitigt oder angepasst werden. Im weiteren Verlauf werden die Änderungen an den einzelnen Anwendungen erläutert.

Änderungen in MobileFacets

Wenn man MobileFacets und MFacets näher betrachtet, stellt man fest, dass unterschiedlich viele Hierarchieebenen der Taxonomien angezeigt werden können. In MobileFacets werden nur zwei Ebenen dargestellt, die Erste dient zur Auswahl der gewünschten Taxonomie und die Zweite zur Auswahl einer Facette. In MFacets hingegen ist es möglich, wenn die Taxonomie es zulässt, innerhalb von vier Ebenen nach der richtigen Facette zu suchen. Dieser Unterschied muss für die Evaluation angepasst werden, da sonst die Pfade zu den angegebenen oder gesuchten Facetten verschieden sind und die Probanden somit eine im Vorhinein unterschiedliche Vorgehensweise zur Bearbeitung der Aufgaben haben. Mit diesem Unterschied wäre also ein Vergleich der Anwendungen diesbezüglich nicht möglich.

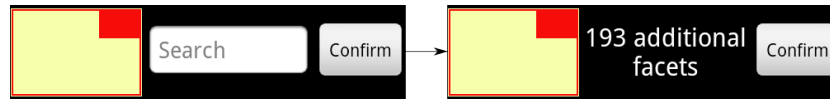
Um dieses Problem zu beheben, kann man entweder die im Hintergrund liegende Taxonomie von MobileFacets an MFacets anpassen oder umgekehrt. Im letzteren Fall hätten die darzustellenden Facetten allerdings nur in der ersten Ebene Subfacetten, so dass die Anzeige von vorhandenen Subfacetten als Gitter im Hintergrund der Facetten in MFacets für die zweite Ebene nicht präsent wäre. Deshalb hat man sich für die erste Variante entschieden und die Darstellung von Facetten auf bis zu vier Ebenen in MobileFacets geändert. Um diese Änderung zu realisieren, müssen die Taxonomien für MobileFacets durch die Taxonomien für MFacets ausgetauscht werden. Außerdem müssen für MobileFacets auch Methoden bereitstehen, um festzustellen, wo sich ein Benutzer in der Taxonomie aufhält, so dass immer die richtigen Facetten während dem Browsen angezeigt werden. Um diesen Austausch der Taxonomien einfacher in MobileFacets zu integrieren, werden auch weitere Methoden mit Verbindung zu Facetten aus MFacets übernommen. Benutzeroberflächenspezifische Methoden zu Facetten, wie z.B. die Auswahl des richtigen Hintergrunds einer Facette, sind von dieser Integration logischerweise nicht betroffen. In diesem Zusammenhang wird auch der Ablauf des Starts der Anwendung abgeändert, indem vor der initialen Suche wie bei MFacets die Taxonomie bzw. Facettenstruktur vom Server erfragt wird. Demzufolge muss auch die Serveradresse in MobileFacets aktualisiert werden.

Ein weiterer Unterschied zwischen beiden Anwendungen liegt darin, dass MFacets Daten eines weiteren Datensatzes unterstützt, nämlich die von Qype stammenden Ressourcen. Dementsprechend müssen die Methoden zur Verarbeitung der Antwort des Servers zu einer Suche angepasst werden, so dass

die Informationen einer Qype Ressource verstanden werden und in die Anwendung richtig integriert werden kann. Dazu gehört auch, dass die Icons zu den Ressourcen um die aus MFacets neu hinzukommenden Icons erweitert und adäquat zugeordnet werden. Um diese Integration zu erleichtern, wird die Klasse zur Abbildung einer Ressource aus MFacets unverändert übernommen.

People From Koblenz (14)	People From The Rhine Province (13)	Living People (12)	People From Koblenz (14)	People From The Rhine Province (13)	Living People (12)
		German Immigrants To The United States (6)			German Immigrants To The United States (6)
Artist (11)			Artist (11)		
Military Person (6)	Athlete (5)	See 193 more facets	Military Person (6)	Athlete (5)	See more facets...

(a) Änderung des More Facets-Buttons.



(b) Austausch des Filters durch die Angabe der Anzahl an Facetten.

Abb. 4.10. Änderungen in MFacets

Änderungen in MFacets

Für die Evaluation muss an MFacets nicht viel verändert werden, da MFacets alle Funktionalitäten von MobileFacets unterstützt. Allerdings wird in MFacets mit der Suche nach Facetten eine zusätzliche Funktion implementiert. Das Filtern der Facetten ist zwar nur in der More-Facets-Ansicht verfügbar, dennoch führt dies dort im Gegensatz zum einfachen Scrollen zu einem schnelleren Finden von Facetten, falls der Name dem Nutzer bekannt ist. Aber genau dieses Scrollen soll evaluiert werden, so dass dieser Filter aus MFacets entfernt werden muss. Da ein reines Entfernen des Filters am unteren Rand der More-Facets-Ansicht eine leere Fläche verursachen würde, hat man sich dazu entschlossen, diese Fläche mit der Anzahl an Facetten in dieser Ansicht zu füllen. Diese Anzahl wird in MFacets schon im Titel des More-Facets-Buttons in der normalen Ansicht dargestellt. Um diese Redundanz zu vermeiden, wird die Anzahl für die Evaluation aus dem Titel entfernt. Diese beiden Änderungen werden in Abb. 4.10 illustriert.

4.4.2 Erweiterungen an den Anwendungen

In Kapitel 5 wird gezeigt, dass die Evaluation der Benutzbarkeit das Messen und Bewerten der Effizienz verlangt. Diese Messung wird an Hand der Schnelligkeit der Bearbeitung der Aufgaben vollzogen. Dabei bezieht sich die Schnelligkeit einerseits auf die benötigte Zeit und andererseits auch auf die benötigten Bearbeitungsschritte. Die Bearbeitungsschritte kann man in diesem Zusammenhang mit den benötigten Klicks gleichsetzen. Um die Zeit zu messen, könnte man eine Stoppuhr verwenden. Allerdings erfordert dies für den Leiter der Evaluation eine hohe Aufmerksamkeit, um ungenaue Messungen zu vermeiden, und stellt somit eine mögliche Fehlerquelle dar. Außerdem könnte die offensichtliche Zeitmessung den Probanden un- oder gar bewusst beeinflussen. Auch das genaue Beobachten der Bearbeitungsschritte zur Ermittlung der Klickzahl kann als weitere mögliche Fehlerquelle betrachtet werden. Deshalb hat man sich dazu entschlossen, die Zeit und Klickzahl über Logeinträge zu messen, d.h. dass spezielle Aktionen der Anwendung zusammen mit dem Zeit des Auftretens in einer Textdatei festgehalten werden. Zur Auswertung werden dann diese Einträge analysiert, um die Anfangs- und End-einträge zu den einzelnen Aufgaben zu ermitteln. Dieses Auswerten wird durch eine gewisse Struktur der Logeinträge erleichtert, welche im Folgenden näher beleuchtet wird.

Die Logeinträge folgen einer Grammatik, die in EBNF³² konzipiert ist und vollständig im Anhang A.5 für MFacets und in A.6 für MobileFacets eingesehen werden kann. Diese Vorgehensweise unterstützt einerseits die Übersicht beim Analysieren der Einträge und andererseits bei der Implementation und Integration in die beiden Anwendungen. Pro Textzeile wird ein Eintrag erstellt, welcher so aufgebaut ist, dass zuerst immer der Zeitpunkt des Eintrags angegeben wird. Danach folgen Informationen, die man in zwei Kategorien einteilen kann. Es können entweder allgemeine Informationen zur initialen Suche festgehalten oder aktive Bearbeitungsschritte des Probanden wiedergegeben werden.

Zu den allgemeinen Informationen gehört, dass der Ort bzw. der Bereich der Suche an Hand seiner Geokoordinaten und die zu verwendenden Datenquellen angegeben werden. Außerdem werden nach der Antwort des Servers Informationen zum erhaltenen Datensatz niedergeschrieben. Dazu gehört die Gesamtzahl an Ressourcen und Facetten aufgeteilt nach Datenquelle und eine Art „Füllstand“ der Facetten. Da Ressourcen mehreren Facetten zugeordnet werden können, ist es interessant zu sehen, wie viele Ressourcen einer einzelnen Facette zugewiesen sind. Mit diesen Informationen lässt sich feststellen, inwieweit die Probanden mit denselben Ressourcen bzw. Facetten zu arbeiten haben, um eine grundsätzliche Vergleichbarkeit der Evaluationsgruppen zu bestätigen oder zu widerlegen. Ansonsten dienen sie eher der Beschreibung der Evaluation als dem Vergleich der einzelnen Aspekte beider Anwendungen.

³² Extended oder Erweiterte Backus-Naur-Form

Daher haben die Grammatiken der Logeinträge für MFacets und MobileFacets diesen Teil gemein.

Für die Bewertung der Schnelligkeit in den einzelnen Aufgaben sind die Informationen zu den Bearbeitungsschritten bedeutsamer. Durch diese Logeinträge lassen sich alle Tätigkeiten des Probanden nachvollziehen. Da sich einige Interaktionen in den Anwendungen unterscheiden, gilt dies auch für die zugehörigen Grammatikregeln. Gemeinsame Interaktionen hingegen sind die Betätigung der Tasten des Smartphones, also die *Menü-*, *Zurück-* oder *Suchen-Taste*. Gleiche Regeln existieren auch bei der Auswahl einer Ressource, der Sortierung der Ressourcen oder bei der Texteingabe im Filter zu den Ressourcen. Weiter ist die Regel zu einem Klick auf einen einfachen Button, z.B. ein *OK-Button*, gleich aufgebaut, so dass die genaue Bezeichnung des Buttons in den jeweiligen Methoden der Anwendungen bestimmt wird. Unterschiedliche Regeln gibt es jedoch bei speziellen Buttons wie dem *Löschen-Button* in der Facetteninformationsliste in MFacets oder den *Löschen-* und *Hinzufügen-Buttons* in MobileFacets, um diese Aktionen unter den Logeinträgen besser wahrnehmen zu können. Ebenso wird der Klick auf eine Facette von beiden Anwendungen anders behandelt. In MobileFacets reicht es aus, den Namen der Facette festzuhalten. In MFacets wird aber zusätzlich noch zwischen einem normalen Facettenklick und einem Klick in der More-Facets-Ansicht unterschieden. Zudem wird, wenn nötig, auch angegeben, wenn durch den Klick eine Facette aus- oder abgewählt wird. Wie zu erwarten ist, besitzen die Eigenheiten der Anwendungen auch eigene Grammatikregeln. Hierzu gehört in MFacets das Öffnen und Schließen der Facetteninformations- und Ergebnisliste und in MobileFacets das Öffnen eines Tabs.

Hiermit ist die Beschreibung der Grammatiken zu den Logeinträgen abgeschlossen. Um sich die Analyse von Logeinträgen besser vorstellen zu können, soll dieses kurze Beispiel näher betrachtet werden:

1	2011/09/15-17:32:03: Click on Facet Shopping
2	2011/09/15-17:32:04: Click on Facet Book Shops , selected
3	2011/09/15-17:32:22: Click Back key to MFacets, Place
4	2011/09/15-17:32:25: Click on Facet Hotels And Holidays
5	2011/09/15-17:32:28: Click on Facet Hotels
6	2011/09/15-17:32:29: Click on Facet 3 Star Hotels , selected

Beispiellogeinträge für MFacets.

Dieses Beispiel zeigt die Auswahl von und Navigation zwischen zwei Facetten in MFacets. Man erkennt, dass es sich um sechs Bearbeitungsschritte bzw. Klicks handelt und dafür 26 Sekunden benötigt worden sind. Die Selektion einer Facette erkennt man an dem Schlüsselwort *selected*. Außerdem sieht man an Hand des Klicks auf die *Zurück-Taste* oder dem Klick auf eine Facette ohne Selektion, dass zwischen verschiedenen Hierarchieebenen gewechselt worden ist. Durch solche Logeinträge lassen sich somit die einzelnen

Bearbeitungsschritte der Probanden einfach nachvollziehen, um die Effizienz der Aufgabenbearbeitung messen und bewerten zu können.

Dieser Abschnitt hat nun gezeigt, welche Maßnahmen notwendig sind, um die beiden mobilen Anwendungen vergleichbar zu machen und für die Erfassung der benötigten Messwerte zur Evaluation vorzubereiten. Was diese Evaluation beinhaltet, wie sie durchgeführt wird und welche Ergebnisse zu verzeichnen sind, wird im nächsten Kapitel ausführlich vorgestellt.

Summative Evaluation

Bisher konzentrierte sich diese Diplomarbeit auf die facettierte Suche und deren Umsetzung in mobilen Anwendungen, insbesondere MobileFacets und MFacets. In diesem Kapitel wird die im Rahmen der Diplomarbeit durchgeführte summative Evaluation zwischen den besagten beiden mobilen Anwendungen präsentiert.

Die Evaluation verfolgt das Ziel, die Anwendungen bezüglich ihrer Benutzbarkeit zu vergleichen. Da die Benutzbarkeit von Anwendungen nicht so einfach wie z.B. eine Zeitdauer mit einer Stoppuhr oder eine Länge mit einem Maßband gemessen werden kann, wird die Evaluation gemäß dem Standard ISO 9241-110:2006¹ aufgebaut. Demnach gilt es, in der Evaluation Aufgaben von Probanden durchführen zu lassen und diese nach Effektivität, Effizienz und Zufriedenheit zu bewerten. Um Effektivität zu messen, kann überprüft werden, ob ein Proband eine Aufgabe erfolgreich abgeschlossen hat. Für die Effizienz lässt sich die Schnelligkeit der Bearbeitung einer Aufgabe messen. Die Zufriedenheit kann jedoch nicht direkt gemessen werden, weshalb Probanden mehrere Fragen nach *IsoMetrics* [GH99] beantworten werden. In *IsoMetrics* werden zwei Varianten an Fragestellungen angeboten, *IsoMetrics^S* für eine summative und *IsoMetrics^L* für eine formative Evaluation. Da es sich bei der Evaluation um zwei fertige, zu vergleichende Anwendungen handelt, wird folglich *IsoMetrics^S* als Grundlage gewählt.

Somit gestaltet sich der Aufbau dieses Kapitels wie folgt. Nachdem der Aufbau der Evaluation in 5.1 erläutert worden ist, wird in 5.2 die Durchführung beschrieben, worauf schließlich in 5.3 die Auswertung der Evaluation zuzüglich von Erkenntnissen und Schlussfolgerungen aus der Evaluation präsentiert werden.

¹ siehe auch http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38009

5.1 Konzeption der Evaluation

Die Evaluation ist nach dem *Zwischen-den-Gruppen-Prinzip* (engl. *between-group*) entworfen, d.h. dass eine Gruppe die Evaluation mit MFacets und die andere Gruppe dieselbe Evaluation mit MobileFacets durchführt. Dies bedeutet, dass man zwar mehr Probanden für die gesamte Evaluation benötigt als beim *Innerhalb-der-Gruppe-Prinzip* (engl. *within-group*), dafür lassen sich aber in der selben Zeit mehr Aufgaben behandeln [LFH10]. Da der Vergleich der Anwendungen an Hand verschiedener Aspekte, also verschiedener Aufgaben, im Vordergrund steht, hat man sich für diesen Ansatz entschieden.

Die Evaluation lässt sich in zwei Abschnitte unterteilen, zum einen in die Aufgabenbearbeitung und zum anderen in den Kommentarteil. Die Ausführung von Aufgaben dient zum Messen der Effektivität und Effizienz und durch die Bewertung von Aussagen und die Angabe freier Kommentare lässt sich auf das Maß der Zufriedenheit schließen. Diese beiden Teile werden nun näher erläutert.

5.1.1 Aufgaben

Die Evaluation beinhaltet die Bearbeitung von insgesamt sieben Aufgaben, welche unterschiedliche Zwecke erfüllen und von beiden Anwendungen erfolgreich bearbeitet werden können. Diese Aufgaben sind jeweils voneinander unabhängig ausführbar, so dass, falls eine Aufgabe von einem Probanden falsch oder nicht gelöst werden kann, alle Aufgaben bearbeitet und somit auch bewertet werden können. Dennoch werden die Aufgaben dem Probanden zusammenhängend in Form einer Stadterkundung präsentiert, um ein reales Anwendungsszenario zu simulieren. Um die Intention bzw. den zu vergleichenden Aspekt der einzelnen Aufgaben aufzuzeigen, reicht die grobe Erläuterung der Aufgaben, so dass an dieser Stelle die genauen Aufgabestellungen unerheblich sind. Daher sei hier bei Bedarf auf den Anhang A.3 verwiesen. Somit folgt nun die Darstellung der Aufgaben.

Aufgabe 1: Suche nach Ressource

Die erste Aufgabe ist als eine einfache zu bearbeitende Aufgabe konzipiert. Es wird vorgegeben, welche Facette auszuwählen ist und welchen Pfad der Proband zu wählen hat. Daraufhin muss der Proband aus der Ergebnisliste die richtige Ressource wählen. Diese Aufgabe ist deshalb so einfach gehalten, dass in der Bearbeitung durch beide Anwendungen keine Unterschiede festzustellen sind. Demnach verfolgt diese Aufgabe den Zweck die Gruppen auf Vergleichbarkeit zu überprüfen. Für die Auswertung und Interpretation der Evaluation ist es wichtig, dass die Gruppen voneinander unabhängig sind (siehe 5.3). Lässt sich bei der Bewertung dieser Aufgabe ein Unterschied feststellen, so deutet dies entweder auf einen Unterschied zwischen den Gruppen, z.B. Probanden mit Erfahrung vs. Probanden ohne Erfahrung mit mobilen Anwendungen,

oder auf eine Eigenheit der Anwendung hin. Gegebenenfalls muss dies bei Auftritt eines Unterschieds näher untersucht werden. Da mit Hilfe einer solchen Aufgabe die Grundlage für die Bewertung dieser und folgender Aufgaben gelegt wird, wird diese auch als *Baseline* Aufgabe oder Test bezeichnet. Im späteren Verlauf bei der Auswertung werden die Aufgaben an Hand von Hypothesen verglichen. Die Hypothese für diese Aufgabe lautet folglich:

Hypothese 1 *Es gibt keinen Unterschied zwischen MFacets und Mobile-Facets.*

Aufgabe 2: Suche nach Facette

In dieser Aufgabe wird dem Probanden eine Ressource vorgegeben, die er durch die Auswahl einer geeigneten Facette finden soll. Hierbei ist die zu tätigende Navigation zur Facette einfach gehalten, so dass diese Aufgabe als weiterer Baseline Test dient. Demzufolge lautet die Hypothese für diese Aufgabe wiederum:

Hypothese 2 *Es gibt keinen Unterschied zwischen MFacets und Mobile-Facets.*

Aufgabe 3: Direkte, textuelle Suche

Die dritte Aufgabe ist die erste Aufgabe, welche nicht mehr als Baseline Test gilt. Nachdem der Proband jegliche bisher ausgewählten Facetten entfernt hat, soll er eine Ressource finden, indem er einen Teil des Namens in das Eingabefeld zum Filtern der Ergebnisliste eintippt. Hier kommt der Unterschied zum Vorschein, dass bei MFacets die Ergebnisliste bei Auswahl keiner Facette alle Ressourcen darstellt und MobileFacets nur die Ressourcen zu ausgewählten Facetten anzeigt. Daher ist für diese Aufgabe mit einem besseren Abschneiden für MFacets zu rechnen, so dass sich folgende Hypothese aufstellen lässt:

Hypothese 3 *Die volle Liste von MFacets führt schneller zum Ziel als die leere Liste von MobileFacets.*

Aufgabe 4: Suche nach Facetten und Ressourcen

Diese Aufgabe behandelt das allgemeine Browsen, d.h. dem Probanden ist weder der Weg zu einer Facette noch die zu suchende Ressource bekannt. Hierzu soll der Proband angegebene Facetten finden, welche in der Hierarchie am niedrigsten liegen, so dass der Proband selbst entscheiden muss, welche höher liegende Facetten er näher untersuchen möchte. Eine gefundene Facette soll dann ausgewählt und eine dem zugehörige Ressource angegeben werden. Um eine eventuelle Streuung der Messwerte beim Auffinden einer einzigen Facette bzw. Ressource zu verringern, soll der Proband vier Facetten bzw. Ressourcen finden. Mit dieser Aufgabe wird der Hauptunterschied zwischen MFacets und MobileFacets untersucht und zwar die verschiedene Darstellung der Facetten und die damit verbundene unterschiedliche Navigation durch

die Facetten. Da MFacets im Grid auf Anhieb mehr Facetten darstellt als MobileFacets in der Liste, wird von einem schnelleren Browsen bei MFacets ausgegangen. Deshalb lässt sich diese Hypothese ableiten:

Hypothese 4 *Das Browsen im Grid von MFacets ist schneller als in der Liste von MobileFacets.*

Aufgabe 5: Suche nach Facetten

Die fünfte Aufgabe verhält sich ähnlich zur zweiten Aufgabe. Es wird eine Ressource vorgegeben, aber nicht unter welcher Facette diese zu finden ist. Dennoch ist der Ablauf dieser Aufgabe etwas anders gestaltet. Dem Probanden werden drei Facetten vorgegeben, die er nach einander untersuchen soll, ob die gesuchte Ressource der Facette zugewiesen ist. Dabei soll der Proband die Facette nicht in tieferen Hierarchieebenen, sondern in derselben Ebene unter vielen Facetten suchen. Dafür bedient sich die Aufgabe der *Person* Taxonomie, welche zwar viele Facetten, aber eine flache Hierarchie besitzt. Somit muss der Proband die Facetten unter mehr als 100 Facetten in einer Ebene finden. Um die angegebenen Facetten zu finden, muss der Proband in beiden Anwendungen wenig scrollen, dennoch sollte die gleichzeitige Darstellung von mehr Facetten in MFacets als in MobileFacets einen Vorteil erzielen. Demzufolge lautet die Hypothese dieser Aufgabe:

Hypothese 5 *Das Finden von Facetten unter vielen Facetten geschieht bei MFacets schneller als bei MobileFacets.*

Aufgabe 6: Suche nach Ressourcen

Diese Aufgabe wird ähnlich zur ersten und zur vorherigen Aufgabe bearbeitet. Wie bei der ersten Aufgabe wird die Facette und der Weg dorthin vorgegeben, so dass der Proband die entsprechende Ressource angeben soll. Zusätzlich werden wie bei der fünften Aufgabe drei Facetten unter vielen in der *Person* Taxonomie zu suchen sein. Der Unterschied zu Aufgabe 5 ist jedoch der, dass diese Facetten im Grid bzw. in der Liste weiter unten zu finden sind, so dass mehr Scrollen notwendig ist. Hierdurch sollte der Vorteil von MFacets wegen der gleichzeitigen Darstellung von mehr Facetten als MobileFacets noch stärker zu tragen kommen. Daher lässt sich diese Hypothese aufstellen:

Hypothese 6 *Das Finden von Facetten unter sehr vielen Facetten geschieht bei MFacets schneller als bei MobileFacets.*

Aufgabe 7: Suche nach Facetten und Ressourcen

Die letzte Aufgabe der Evaluation bezieht sich wie die vierte Aufgabe auf das allgemeine Browsen. Auch hier werden wieder zu vier Facetten entsprechende Ressourcen gesucht, aber bei genauerer Betrachtung sind die Facetten etwas schwieriger zu finden als in Aufgabe 4. Das liegt daran, dass es weniger klar sein sollte, unter welchen Facetten sich die gesuchten Facetten befinden. Dennoch gilt vom Prinzip her für Aufgabe 7, was auch für Aufgabe 4 gilt, so dass die Hypothese wie folgt aussieht:

Hypothese 7 *Das Browsen im Grid von MFacets ist schneller als in der Liste von MobileFacets.*

5.1.2 Kommentare

Nach den Aufgaben folgt der Kommentarteil der Evaluation. Hier wird zur Bewertung der Benutzbarkeit die Zufriedenheit der Probanden näher untersucht. Dazu sollen die Probanden einerseits Aussagen bewerten und, wenn möglich, danach noch frei formulierte Kommentare abgeben.

Die Aussagen sind gemäß *IsoMetrics*^S [GH99] ausgewählt und werden um anwendungsspezifische Aussagen erweitert. Insgesamt gilt es von den Probanden der MFacets-Gruppe 29 Aussagen zu bewerten, während die Probanden der MobileFacets-Gruppe nur 28 Aussagen bewerten. Dieser Unterschied lässt sich durch das Hinzufügen einer speziellen Aussage zur MFacets-Anwendung erklären, welche sich auf die gleichzeitige Verwendung von Facetten und Ergebnisliste auf demselben Bildschirm bezieht und in der späteren Auswertung nicht verglichen werden kann.

Die Aussagen lassen sich in drei Kategorien einteilen und werden folgendermaßen präsentiert. Zuerst werden acht Aussagen allgemein zur Anwendung getätigt, wie z.B. „Die Anwendung erfordert keine überflüssigen Eingaben.“ oder „Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.“. Darauf folgen neun Aussagen mit Bezug zu den Facetten oder der Ergebnisliste, wie z.B. „Das Auswählen mehrerer Facetten ist einfach.“ oder „Die Suche nach Ressourcen in der Ergebnisliste ist intuitiv.“. Zuletzt folgen elf Aussagen zur Benutzeroberfläche und zu Funktionalitäten, wie z.B. „Die Anwendung erschwert meine Aufgabenbearbeitung durch eine uneinheitliche Gestaltung.“ oder „Ich kann die Rückmeldungen, die ich von der Anwendung erhalte, eindeutig dem auslösenden Vorgang zuordnen.“. Wie man an den aufgeführten Beispielaussagen sieht, werden Aussagen auch negativ ausgedrückt, so dass nicht nur positive Bewertungen zu erwarten sind. Da an dieser Stelle nur ein Ausschnitt aller Aussagen wiedergegeben worden ist, sei hiermit auf den Anhang A.4 für die komplette Übersicht verwiesen.

trifft nicht zu |—————| trifft voll zu

Abb. 5.1. Kontinuierliche Likert-Skala zur Bewertung der Aussagen in der Evaluation.

Nach *IsoMetrics*^S werden die Aussagen mittels einer *Likert-Skala* von 1 für *trifft nicht zu* bis 5 für *trifft voll zu* bewertet. Dies wurde für diese Evaluation nicht übernommen um die Diskussion zu umgehen, ob nun eine Skala bis fünf ausreichend ist oder eine Skala bis sieben oder gar zehn besser geeignet wäre [GM08]. Stattdessen wird eine kontinuierliche Likert-Skala verwendet, was zusätzlich dazu führt, dass für die Aussagen auch dieselben statistischen

Analysen zur Auswertung verwendet werden können wie für die Aufgaben. Eine solche kontinuierliche Likert-Skala findet man in Abb. 5.1. Auf dieser Skala können Probanden frei auswählen, an welcher Stelle sie ein Kreuz bzw. einen senkrechten Strich setzen, um die jeweilige Aussage zu bewerten. Um die Bewertungen auszuwerten, muss der Abstand des Strichs zu den Enden per Hand gemessen werden. Auf diesen Daten lassen sich die statistischen Analysen auszuführen. Zur besseren Darstellung können die Ergebnisse im Nachhinein auf eine Likert-Skala von 1-5 projiziert werden.

Als letztes folgt nach der Bewertung der Aussagen die Angabe von freien Kommentare in einem Textfeld. Diese Kommentare lassen sich in der Regel nur schwer bis gar nicht vergleichen, so dass diese eher das Stimmungsbild eines Probanden wiedergeben. Das Kommentarfeld bietet den Probanden die Möglichkeit, Anmerkungen zur Anwendung zu tätigen, speziell solche, die von den Aussagen nicht aufgegriffen worden sind, wie z.B. konkrete Verbesserungsvorschläge oder auch Angaben darüber, was besonders gefallen hat. Falls gewisse Kommentare mehrfach vorkommen, so können diese bei der Auswertung besonders begutachtet werden.

In diesem Abschnitt wurde gezeigt, wie die Evaluation aufgebaut ist und welchen Zweck die einzelnen Aufgaben, Aussagen und Kommentare besitzen. Die Durchführung dieser Evaluation ist Gegenstand des nächsten Abschnitts.

5.2 Durchführung

Im folgenden Abschnitt wird nun wiedergegeben, wie die Evaluation durchgeführt worden ist, welche Fehler oder Probleme aufgetreten sind und wie diese behoben worden sind.

Die Evaluation hat zwischen dem 12. und 26. September 2011 stattgefunden, währenddessen 28 Probanden evaluiert worden sind. Für die Durchführung der Evaluation haben die Probanden eine Aufwandsentschädigung in Höhe von pauschal 10 Euro pro Stunde erhalten. Jede Evaluation ist auf dem gleichen Smartphone ausgeführt worden, nämlich einem handelsüblichen HTC Desire mit Google Android 2.2 Betriebssystem. Bevor ein Proband die jeweilige mobile Anwendung gestartet hat, wurden auf dem Smartphone der Bildschirmschoner und das GPS abgeschaltet worden. Dies hat den Grund, dass der Proband nach einer Zeit der Nicht-Benutzung, wie z.B. während dem Lesen einer Aufgabenstellung, den Bildschirm nicht wieder anschalten muss und dass die initiale Suche der Anwendung über den gleichen voreingestellten Ort abläuft. Die Internetverbindung um den Server zu erreichen ist immer per WLAN aufgebaut worden. Nachdem ein Proband die ihm zufällig zugeteilte Anwendung gestartet hatte, sind von ihm zur Überbrückung der Wartezeit während dem Suchprozess einige allgemeine, anonyme Fragen zur Beschreibung des Probanden beantwortet worden. Bevor ein Proband mit der Aufgabenbearbeitung angefangen hat, wurde eine Einführung in die jeweilige

Anwendung vollzogen. In dieser Einführung sind die benötigten Interaktionselemente an Hand von Screenshots und einer Beispielaufgabe vorgestellt worden. Die genauen Details zur Einführung lassen sich für MFacets in Anhang A.1 und für MobileFacets in A.2 einsehen. Danach sind die eigentlichen Aufgaben der Evaluation bearbeitet worden, worauf die Bewertung der Aussagen und die eventuelle Angabe freier Kommentare gefolgt sind. Während ein paar wenigen, aber erwähnenswerten Evaluationsdurchführungen sind Probleme aufgetreten, die nun näher erläutert werden.

Eine Evaluation musste frühzeitig abgebrochen werden, da die initiale Suche selbst nach mehreren Versuchen zu einem unzureichenden Ergebnis geführt hat, d.h. für die Evaluation benötigte Ressourcen sind vom Server nicht zurück geliefert worden. Der Proband zu dieser Evaluation ist daraufhin zu einem neuen Termin eingeladen worden. Für die darauffolgenden Evaluationen ist dieses serverseitige Problem nach einem Neustart des Servers behoben gewesen.

In zwei Evaluationen ist es zu Fehleingaben der Probanden gekommen, die für die spätere Auswertung gesondert behandelt werden mussten. Ein Proband hat für die Bearbeitung einer Aufgabe eine neue Suche ausgeführt, so dass die verursachte Wartezeit für die Bewertung der Aufgabe gestrichen werden musste. Ein weiterer Proband hat die Warnung vor dem Schließen der Anwendung nicht richtig wahrgenommen und bestätigt, so dass diese neu gestartet werden musste. Für die Messung der Aufgabe musste die zusätzliche Wartezeit bis zur Weiterbearbeitung ignoriert werden. Außerdem musste diese gesamte Evaluation an Hand von zwei Logdateien ausgewertet werden.

Mit diesen Beschreibungen ist die allgemeine Durchführung der Evaluation erläutert. Daher werden nun im letzten Abschnitt dieses Kapitels die Auswertung der erfassten Messungen und die daraus resultierenden Schlussfolgerungen behandelt.

5.3 Auswertung und Interpretation der Messergebnisse

Nach der Durchführung der Evaluation gilt es, die Messergebnisse auszuwerten und entsprechend zu interpretieren. Dies ist Gegenstand dieses Abschnitts, so dass zunächst die Probanden beschrieben werden, um zu zeigen, dass es sich um zwei zufällige und unabhängige Gruppen handelt. Danach folgt die Auswertung der Effektivität, Effizienz und Zufriedenheit, um anschließend Schlussfolgerungen bezüglich der Benutzbarkeit zu ziehen.

5.3.1 Beschreibung der Probanden und des Datensatzes

Wie schon im Abschnitt 5.2 zur Durchführung erwähnt, gibt es insgesamt 28 Probanden, also 14 Probanden je Gruppe. Diese Probanden haben am Anfang der Evaluation allgemeine Fragen beantwortet, um unter anderem ihren Erfahrungswert mit Smartphones und mobilen Anwendungen festzustellen.

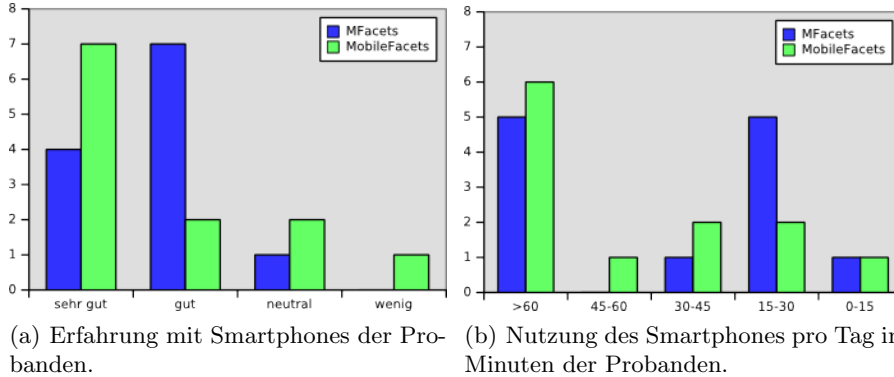


Abb. 5.2. Erfahrung mit und Nutzung des Smartphones der Probanden.

Zu diesen Fragen gehört auch die Angabe, ob man schon Erfahrung mit einem Touchscreen als Interaktionsform gesammelt hat. Nach der Auswertung der Messungen aller 28 Probanden hat sich herausgestellt, dass die Bearbeitung der Aufgaben von Probanden ohne Touchscreen-Erfahrung langsamer ausgefallen ist als die der Probanden mit Touchscreen-Erfahrung. Aus diesem Grund werden diese Probanden für die Auswertung der Evaluation nicht berücksichtigt, so dass nur noch die Messungen von 24 Probanden, zwölf je Gruppe, bewertet werden.

Die Probanden der MFacets Gruppe lassen sich zusammenfassend wie folgt beschreiben. Von den zwölf Probanden sind zwei weiblichen Geschlechts und die Altersstruktur verteilt sich zwischen 19 und 35 Jahren so, dass der durchschnittliche Proband im Mittel 25,92 und im Median 25,5 Jahre alt ist. Als Beruf haben zehn Probanden Student und zwei Doktorand angegeben. Sieben Studenten und beide Doktoranden entstammen der Informatik (Informatik, Wirtschaftsinformatik oder Informationsmanagement) und jeweils ein Student der Elektrotechnik, des Lehramts und der Kulturwissenschaft.

Auch die zwölf Probanden der MobileFacets Gruppe beinhalten zwei weibliche Personen. Der durchschnittliche Proband ist bei einer Verteilung von 22 bis 34 Jahren im Mittel 26 und im Median 25 Jahre alt. Alle zwölf Probanden sind Studenten, wovon zehn Studenten ein Studienfach der Informatik (Informatik, Computervisualistik, Wirtschaftsinformatik oder Informationsmanagement) belegen. Die zwei übrigen Studenten sind dem Studienfach des Lehramts bzw. des Magisters zuzuordnen.

Um die Erfahrung eines Probanden mit einem Smartphone widerzuspiegeln, haben die Probanden eine Selbsteinschätzung bezüglich ihrer Erfahrung und Nutzung des Smartphones pro Tag in Minuten angegeben. Dabei hatten die Probanden bei der Einschätzung, wie gut sie sich mit Smartphones auskennen, die Wahl zwischen fünf Vorgaben von *sehr gut* bis *gar nicht*. Auch bei der Angabe der Nutzung pro Tag in Minuten sind fünf Möglichkeiten

vorgegeben gewesen. Wie die Verteilung der einzelnen Gruppen aussieht, wird in Abb. 5.2 illustriert. Hier erkennt man, dass die Probanden der Gruppen sich in ihrer Erfahrung im Durchschnitt ähnlich einschätzen. Bei der Nutzung des Smartphones pro Tag liegt der durchschnittliche Wert bei den Probanden in der MobileFacets Gruppe höher als in der MFacets Gruppe. Diese beiden Verteilungen deuten auf zufällige Gruppen hin, dennoch sollte man bei der Bewertung der Messergebnisse eventuell die höhere Nutzung in der MobileFacets Gruppe berücksichtigen.

Der durchschnittlich verwendete Datensatz hat in der MFacets Gruppe im Mittel 354,58 und im Median 403 Ressourcen beinhalten. In der MobileFacets Gruppe liegt der Wert der Ressourcen im Mittel bei 370,08 und im Median bei 401,5. Der Unterschied im Mittel von ca. 4% lässt sich dadurch erklären, dass in der MFacets Gruppe drei und in der MobileFacets Gruppe nur zwei Probanden mit einem Datensatz von ca. 210 Ressourcen gearbeitet haben. Dennoch sind die für die Evaluation benötigten Ressourcen auch in diesem verringerten Datensatz vorhanden gewesen. Ein Grund dafür, dass der Server für diese Probanden weniger Ressourcen als Antwort geliefert hat, kann durch eine temporäre Überbelastung des Servers einer oder mehrerer Quellen erklärt werden. Trotz dieses Unterschieds in der Ressourcenanzahl haben die Probanden beider Gruppen mit 333,46 für MFacets und 333,33 für MobileFacets fast gleiche Facettenzahlen erhalten. Auch die Füllstände der Facetten, also wie viele Ressourcen im Mittel einer Facette zugeordnet sind, liegen in beiden Gruppen ähnlich. Hier erzielen die MFacets Gruppe im Mittel 3,70 und im Median 3,97 und die MobileFacets Gruppe im Mittel 3,78 und im Median 3,97 Ressourcen pro Facette. Aus diesen statistischen Werten lässt sich folgern, dass die Probanden beider Gruppen im Durchschnitt mit einem fast gleichen Datensatz gearbeitet haben, so dass dies die Evaluation zum Vergleich der beiden Anwendungen nicht beeinflusst.

5.3.2 Auswertung der Effektivität

Wie eingangs in diesem Kapitel beschrieben wird die Effektivität daran bemessen, ob ein Proband eine Aufgabe richtig oder falsch bearbeitet hat. Dies lässt sich für diese Evaluation schnell zusammenfassen, da jeder Proband der MFacets Gruppe jede Aufgabe erfolgreich bearbeitet hat und bei MobileFacets nur 1 3/4 Aufgaben fehlerhaft beantwortet worden sind. Diese falschen Antworten sind allerdings durch die Vorgehensweise der einzelnen Probanden zu erklären und nicht mit falschen Rückmeldungen der Anwendung. Genauer ist eine Aufgabe falsch beantwortet worden, da der Proband auf dem Weg zur richtigen Facette eine Hierarchieebene zu früh die Suche beendet und somit eine falsche Ressource als Antwort geliefert hat. Im anderen Fall hat ein Proband eine falsche Facette aufgesucht. Dennoch ist im letzten Fall die zugehörige Aufgabe zu 3/4 richtig beantwortet worden, da insgesamt vier Ressourcen zu vier unterschiedlichen Facetten gesucht worden sind. Demzufolge ergibt sich bei $12 * 7 = 84$ Aufgaben je Gruppe eine Effektivität von 100% für

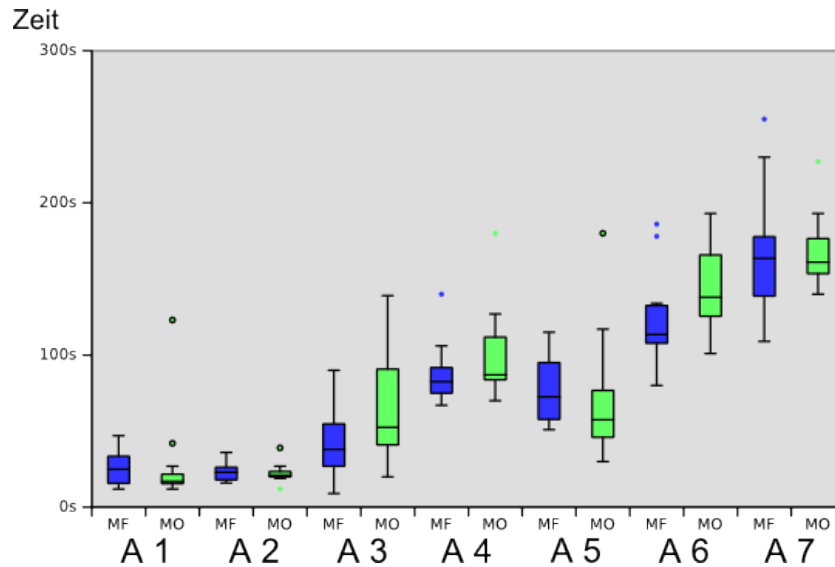


Abb. 5.3. Boxplot zur benötigten Bearbeitungszeit der Aufgaben der Evaluation.

MFacets und 98,52% für MobileFacets. Für den Vergleich der Benutzbarkeit beider Anwendungen spielt die Effektivität somit keine Rolle.

5.3.3 Auswertung der Effizienz

Die Effizienz wird an Hand der Logeinträge, die während der Aufgabenbearbeitung generiert worden sind, bewertet. Um die Messungen miteinander zu vergleichen und zu interpretieren, kann man sich statistischen Analysen bedienen. Eine Möglichkeit Daten aus zwei Gruppen zu vergleichen bietet der *t-Test für zwei unabhängige Stichproben* [LFH10]. Dieser Test kann aussagen, ob ein signifikanter Unterschied zwischen den Mittelwerten zweier Stichproben vorliegt. Allerdings sind dafür gewisse Voraussetzungen notwendig. Einerseits müssen die Stichproben unabhängig voneinander und andererseits normalverteilt sein. Da die Gruppen zufällig zusammengesetzt worden sind und jeder Proband die gleichen Voraussetzungen zur Bearbeitung gehabt hat, kann man von unabhängigen Stichproben ausgehen. Eine Überprüfung der Normalverteilung hingegen muss für die Messungen je Gruppe und je Aufgabe durchgeführt werden. Liegt keine Normalverteilung vor, so kann der *Mann-Whitney U Test* zum Vergleich verwendet werden. Bei beiden Tests werden Werte berechnet, t beim t -Test und Z beim Mann-Whitney U Test, von denen man die Wahrscheinlichkeit eines signifikanten Unterschieds ableiten kann. In dieser Auswertung ist ein signifikanter Unterschied ab einer Wahrscheinlichkeit unter 5% (Signifikanzniveau $\alpha = 0,05$) gegeben. Liegt ein signifikanter Unterschied vor, so kann man die sogenannte *Nullhypothese* ablehnen und die

Alternativhypothese annehmen. Die Nullhypothese besagt, dass die Mittelwerte gleich sind, während die Alternativhypothese von ungleichen Mittelwerten ausgeht. Wenn also ein signifikanter Unterschied auftritt, dann kann auf einen Unterschied zwischen den beiden Anwendungen geschlossen werden. Außerdem ist es üblich, einen signifikanten Unterschied mit einer Effektstärke zu bewerten [Rob11]. Zur Berechnung der Effektstärke werden hier *Cohens d* oder *Pearsons r^2* verwendet. Diese Werte sind dimensionslos und geben die Relevanz eines signifikanten Unterschieds an. Eine Effektstärke kann als klein ($0,2 \leq |d| < 0,5$ bzw. $0,01 \leq r^2 < 0,09$), mittel ($0,5 \leq |d| < 0,8$ bzw. $0,09 \leq r^2 < 0,25$) oder groß ($|d| \geq 0,8$ bzw. $r^2 \geq 0,25$) bezeichnet werden. Im Folgenden wird nun die Auswertung der Effizienz nach Zeit und nach Klicks unterschieden. Die dargestellten Berechnungsergebnisse sind mit Hilfe von Gnumeric² in der Version 1.10.16 erstellt worden.

Auswertung der Effizienz nach Zeit

Wie viel Zeit die Probanden für die einzelnen Aufgaben benötigt haben, wird in Abb. 5.3 verdeutlicht. In 5.1.1 ist gezeigt worden, dass die Aufgaben 1 und 2 als Baseline Tests gelten. Die Abbildung deutet auf keinen signifikanten Unterschied zwischen den Anwendungen in den beiden Aufgaben hin, was insbesondere durch die entsprechenden Werte in Tabelle 5.1 bestätigt wird ($P = 0,21$ bzw. $P = 0,354$). In Aufgabe 3 kann ein signifikanter Unterschied festgestellt werden ($P = 0,026$), was auch fett in der Tabelle hervorgehoben ist. Die Effektstärke in Aufgabe 3 ist für $d = -0,841$ groß und für $r^2 = 0,162$ mittel. Aufgabe 4 kann mit $P = 0,054$ fast einen signifikanten Unterschied vorweisen, was deshalb in der Tabelle kursiv dargestellt ist. Diese Aufgabe hätte eine mittlere Effektstärke mit $d = -0,508$ und $r^2 = 0,109$. Die Abbildung zu Aufgabe 5 deutet auf einen signifikanten Unterschied zu Gunsten für MobileFacets hin, was die zugehörigen Werte jedoch verneinen. In Aufgabe 6 sieht

Zeit	MFacets		MobileFacets		relativ		Z oder t	P
	Mittel	Median	Mittel	Median	Mittel	Median		
Aufgabe 1	25,833	25	28,417	17,5	91%	143%	Z = -0,808	0,21
Aufgabe 2	23,5	23	22,5	21	104%	110%	t = 0,379	0,354
Aufgabe 3	41,167	38	68,833	52,5	60%	72%	t = -2,06	0,026
Aufgabe 4	87,417	82,5	100,583	87	87%	95%	Z = -1,617	<i>0,054</i>
Aufgabe 5	77,167	72,5	69,917	57,5	110%	126%	t = 0,529	0,301
Aufgabe 6	122,583	113,5	145,417	138	84%	82%	t = -1,803	0,043
Aufgabe 7	167	163,5	168	161	99%	102%	t = -0,07	0,472
Gesamt	790,333	771,5	852,917	800	93%	96%	Z = -1,559	<i>0,06</i>

Tabelle 5.1. Auswertung der Aufgaben der Evaluation nach Zeit in Sekunden.

² siehe <http://projects.gnome.org/gnumeric/>

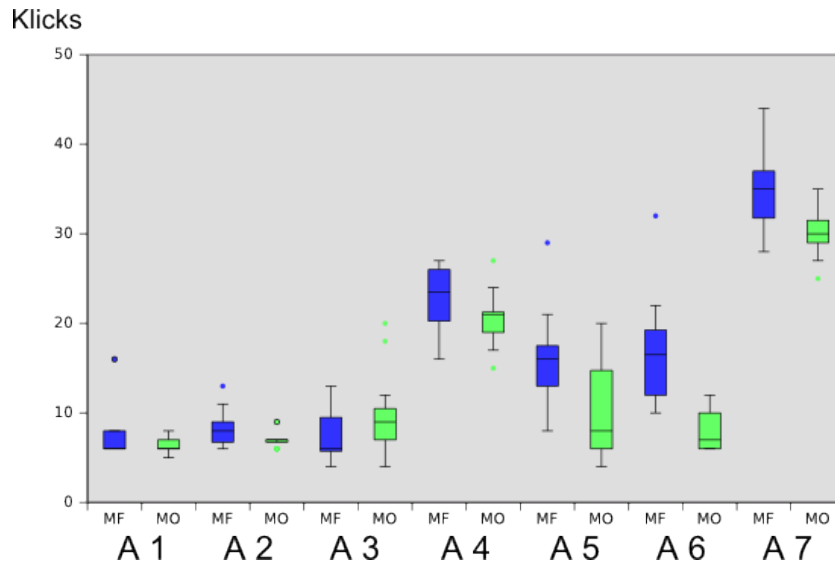


Abb. 5.4. Boxplot zur benötigten Klickzahl der Aufgaben der Evaluation.

man einen weiteren signifikanten Unterschied bei $P = 0,043$ und mittleren Effektstärken mit $d = -0,736$ und $r^2 = 0,129$. Bei Aufgabe 7 wiederum ist kein Unterschied zu sehen. In der Tabelle werden zusätzlich noch die Werte für die insgesamt benötigte Zeit präsentiert, wo bei $P = 0,06$ wieder fast ein signifikanter Unterschied vorliegt. Die Effektstärke wäre hier bei $d = -0,629$ und $r^2 = 0,101$ mittel stark. Die Gesamtzeit stellt hier nicht, wie vielleicht zu vermuten wäre, die Summe der Zeiten der einzelnen Aufgaben dar, sondern die wirklich benötigte Zeit vom ersten Klick der Aufgabe 1 bis zum letzten Klick der Aufgabe 7. Dies schließt somit neben der Bearbeitungszeit auch die Zeit zum Lesen, zum Stellen eventueller kleiner Zwischenfragen oder dem Schreiben der Antworten ein. Da manche Probanden auch nach Start der Bearbeitung einer Aufgabe nochmals auf den Aufgabentext geschaut haben, normalisiert diese Gesamtzeit individuelle Diskrepanzen zu den Durchschnittszeiten.

Auswertung der Effizienz nach Klicks

Die jeweiligen Klickzahlen der Probanden zu den einzelnen Aufgaben werden in Abb. 5.4 illustriert und die zugehörigen Werte in Tabelle 5.2 aufgelistet. Die Boxplots zu den Aufgaben 1 und 2 deuten auf eine erhöhte Klickzahl für MFacets in den Baseline Tests hin, was die Werte in der Tabelle belegen. Bei $P = 0,007$ bzw. $P = 0,009$ liegen signifikante Unterschiede vor mit Effektstärken von $d = 0,52$ (mittel) und $r^2 = 0,257$ (groß) bzw. $d = 0,827$ (groß) und $r^2 = 0,233$ (mittel). In der Abbildung lässt sich ein Unterschied in Aufgabe 3 zum Vorteil von MFacets vermuten, was aber durch die Werte nicht belegt wird.

Klicks	MFacets		MobileFacets		relativ		Z oder t	P
	Mittel	Median	Mittel	Median	Mittel	Median		
Aufgabe 1	7,417	6	6,33	6	117%	100%	Z = -2,483	0,007
Aufgabe 2	8,25	8	6,917	7	119%	114%	Z = -2,367	0,009
Aufgabe 3	7,583	6	9,917	9	76%	67%	t = -1,405	0,087
Aufgabe 4	22,5	23,5	20,5	21	110%	112%	t = 1,364	0,093
Aufgabe 5	16,333	16	10,667	8	153%	200%	Z = -2,252	0,012
Aufgabe 6	16,667	16,5	7,917	7	211%	236%	Z = -4,041	<0,001
Aufgabe 7	34,75	35	30,333	30	115%	117%	t = 2,58	0,009
Gesamt	133,583	132	104,25	105	128%	126%	t = 4,379	<0,001

Tabelle 5.2. Auswertung der Aufgaben der Evaluation nach Klicks.

Auch in Aufgabe 4 stellt man keinen signifikanten Unterschied fest. Die Aufgaben 5-7 weisen hingegen wieder signifikante Unterschiede vor. In Aufgabe 5 liegt die Wahrscheinlichkeit bei $P = 0,012$ und die Effektstärke stellt sich als groß ($d = 0,994$) bzw. mittel ($r^2 = 0,211$) heraus. Aufgabe 6 zeigt einen deutlichen Unterschied auf bei $P < 0,001$ und großen Effektstärken von $d = 1,874$ und $r^2 = 0,681$. In Aufgabe 7 wird der signifikante Unterschied durch $P = 0,009$ bekräftigt, welcher mit einer großen ($d = 1,053$) bzw. mittleren ($r^2 = 0,232$) Effektstärke einhergeht. In der Tabelle erkennt man, dass auch die insgesamt benötigte Klickzahl einen signifikanten Unterschied aufweist. Hier liegt die Wahrscheinlichkeit bei $P < 0,001$ deutlich unter dem Signifikanzniveau und der Unterschied besitzt eine große Effektstärke mit $d = 1,788$ und $r^2 = 0,466$.

5.3.4 Auswertung der Zufriedenheit

In 5.1.2 ist gezeigt worden, dass die Zufriedenheit über die Bewertung der Aussagen und die Angabe freier Kommentare gemessen werden kann. Daher wird dieser Unterabschnitt dementsprechend unterteilt.

Auswertung der Bewertung der Aussagen

Die Aussagen sind in einer kontinuierlichen Likert-Skala bewertet worden, so dass dieselben statistischen Analysen wie bei den Aufgaben durchgeführt werden können. Um genauere Berechnungen durchführen zu können, werden die wirklich gemessenen Werte verwendet. Zur Darstellung werden allerdings die resultierenden Werte in eine übliche Likert-Skala von 1-5 umgewandelt. Da *Cohens d* direkt von den Werten abgeleitet wird, wird hier nur *Pearsons r²* zur Angabe der Effektstärke verwendet. Im Folgenden werden nur die Aussagen mit Baseline Charakter oder mindestens annäherndem signifikanten Unterschied näher erläutert. Die kompletten Bewertungen können dennoch in Tabelle 5.3 eingesehen werden.

Aussagen	MFacets		MobileFacets		Z oder t	P
	Mittel	Median	Mittel	Median		
#1	1,676	1,61	1,521	1,438	t = 0,856	0,201
#2	4,178	4,219	3,679	4,01	t = 1,152	0,131
#3	3,737	3,686	3,94	3,8	t = -0,571	0,288
#4	4,051	4,41	3,86	4,2	t = 0,503	0,31
#5	4,34	4,524	4,317	4,448	t = 0,079	0,469
#6	4,365	4,73	3,641	3,838	Z = -1,674	0,046
#7	1,857	1,781	1,905	1,781	t = -0,193	0,424
#8	3,813	4,01	3,673	4,105	t = 0,312	0,379
#9	1,924	1,724	2,187	2,01	t = -0,651	0,261
#10	4,416	4,581	4,365	4,486	t = 0,2	0,422
#11	3,721	4,352	4,403	4,543	t = -1,582	<i>0,064</i>
#12	4,025	4,086	3,822	4,2	t = 0,622	0,27
#13	4,044	4,086	3,594	3,571	t = 1,301	0,103
#14	1,768	1,419	1,07	1,038	t = 3,298	0,002
#15	3,937	4,162	3,902	3,724	t = 0,082	0,468
#16	4,483	4,676	4,606	4,829	Z = -0,346	0,363
#17	4,438	4,695	4,597	4,752	t = -0,714	0,241
#18	3,987	4,086	3,924	4,314	t = 0,142	0,444
#19	4,102	4,295	4,063	4,486	t = 0,096	0,462
#20	4,124	4,219	3,994	4,352	t = 0,351	0,364
#21	1,727	1,19	1,352	1,171	Z = -0,693	0,242
#22	1,552	1,362	1,79	1,419	t = -0,735	0,235
#23	4,232	4,257	4,232	4,6	Z = -1,386	0,082
#24	4,641	4,81	4,644	4,79	t = -0,019	0,492
#25	4,771	5	4,867	4,924	Z = -1,155	0,123
#26	4,14	4,238	4,46	4,619	Z = -1,848	0,033
#27	4,708	4,905	4,838	4,943	Z = -1,559	<i>0,06</i>
#28	1,105	1	1,019	1	Z = -1,848	0,033
#29	4,556	4,714				

Tabelle 5.3. Auswertung der Aussagen der Evaluation.

Baseline Charakter haben vier Aussagen, bei denen kein Unterschied zu erwarten ist. Dies sind die Aussagen #17 „Die Suche nach Ressourcen in der Ergebnisliste ist intuitiv.“, #25 „Ich war mir bei den Wartezeiten immer sicher, dass die Anwendung weiterarbeitet.“, #27 „Vor der Ausführung möglicherweise problematischer Aktionen gibt die Anwendung eine Warnung aus.“ und #28 „Bei meiner Arbeit mit der Anwendung treten häufig Systemfehler, z.B. Absturz, auf.“. Bei den ersten beiden Aussagen ist kein Unterschied zu erkennen. Aber bei Aussage #27 tritt bei $P = 0,06$ fast ein signifikanter Unterschied pro MobileFacets auf und hätte eine mittlere Effektstärke bei

$r^2 = 0,101$. Aussage #28 hat mit $P = 0,033$ einen signifikanten Unterschied für MobileFacets bei einer mittleren Effektstärke von $r^2 = 0,142$.

Insgesamt gibt es noch bei drei weiteren Aussagen einen signifikanten und bei einer Aussage fast einen signifikanten Unterschied. Aussage #6 „Die Anwendung enthält alle für die Aufgaben benötigten Funktionen.“ zeigt mit $P = 0,046$ den ersten signifikanten Unterschied zu Gunsten von MFacets auf, wobei eine mittlere Effektstärke ($r^2 = 0,117$) festgestellt wird. Die Aussage #11 „Das Abwählen/Löschen von markierten Facetten ist intuitiv.“ besitzt bei $P = 0,064$ fast einen signifikanten Unterschied zum Vorteil von MobileFacets und hätte eine mittlere Effektstärke von $r^2 = 0,102$. In Aussage #14 „Ich weiß nie, welche Facetten ich bereits ausgewählt habe.“ sieht man bei $P = 0,002$ wieder einen signifikanten Unterschied pro MobileFacets mit einer großen Effektstärke von $r^2 = 0,331$. Mit $P = 0,033$ lässt sich bei Aussage #26 „Es ist für mich einfach, zwischen unterschiedlichen Bearbeitungsbildschirmen zu wechseln.“ der letzte signifikante Unterschied für MobileFacets feststellen. Hier liegt eine mittlere Effektstärke bei $r^2 = 0,142$ vor.

Auswertung der freien Kommentare

Zuletzt werden noch die wichtigsten Kommentare nach Gruppen getrennt zusammengefasst wiedergegeben. Diese lassen sich wie folgt auflisten:

- **MFacets**
 - 4 x More Facets soll verbessert werden (1 x Auswahl in More Facets soll zurückspringen).
 - 4 x Alphabetische Sortierung ermöglichen (2 x nur in More Facets).
 - 4 x Klick auf Ressource soll Ressource öffnen, nicht nur über den Button.
 - 3 x Öffnen der Ergebnisliste ist nicht intuitiv → Button hinzufügen.
 - 2 x Leiste vergrößern.
 - 2 x Suche nach Facetten ermöglichen → Suchfeld.
 - 2 x Direktes Abwählen der Facetten auf dem Bildschirm ermöglichen, nicht nur über die Leiste.
 - 2 x Breadcrumbs hinzufügen.
 - 2 x Löschen-Button vergrößern.
 - 2 x Datenqualität verbessern.
 - 1 x Auswahl der Verknüpfung der Facetten hinzufügen.
- **MobileFacets**
 - 8 x Suche nach Facetten ermöglichen → Suchfeld.
 - 7 x Klick auf die Zurück-Taste im Ergebnislisten-Tab soll nicht die Anwendung schließen, sondern auf den Facetten-Tab springen.
 - 6 x Ressourcen im Facetten-Tab sollen dieselben Informationen wie im Ergebnislisten-Tab haben.
 - 4 x Alphabetische Sortierung ermöglichen.
 - 4 x Datenqualität verbessern.

- 2 x Klick auf die Zurück-Taste in der Liste soll an die letzte und nicht an die erste Stelle springen.
- 2 x Breadcrumbs anwählbar machen.
- 1 x Auswahl der Verknüpfung der Facetten hinzufügen.
- 1 x Ergebnisliste soll initial voll sein.
- 1 x Ergebnisse sollen nach Facettenauswahl direkt angezeigt werden
→ gleicher Bildschirm.

Nach der Auswertung der Effektivität, Effizienz und Zufriedenheit sollen im nächsten Unterabschnitt die Ergebnisse interpretiert werden, so dass Schlussfolgerungen zur Benutzbarkeit der Anwendungen gezogen werden können.

5.3.5 Schlussfolgerungen

Im Folgenden werden nun die Auswertungen der einzelnen Aufgaben mit Bezug auf die in 5.1.1 gestellten Hypothesen näher betrachtet. Danach werden bestimmte Bewertungen der Aussagen entsprechend interpretiert, um zusammen mit den freien Kommentaren einen Überblick über die Zufriedenheit der Probanden zu erhalten. Anschließend wird die Benutzbarkeit beider Anwendungen bewertet.

Schlussfolgerungen für die Effizienz

Aufgabe 1 und 2 sind als Baseline Aufgaben bestimmt worden mit der Hypothese, dass kein Unterschied zwischen MFacets und MobileFacets festgestellt wird. Für die Zeit trifft dies zu, allerdings nicht für die Klickzahlen in beiden Aufgaben, wo jeweils ein signifikanter Unterschied erkannt worden ist. Wenn man sich die Zahlen genauer anschaut, sieht man, dass die Probanden in MFacets ca. einen Klick pro Aufgabe mehr benötigt haben. Da für die Bearbeitung beider Aufgaben nur wenige Klicks notwendig sind, erhält man so einen relativen Unterschied von bis zu 19% mehr Klicks. Die Messungen aus beiden Aufgaben suggerieren also, dass MFacets im Allgemeinen, also auch für die darauffolgenden Aufgaben, mehr Klicks als MobileFacets benötigen wird, was die Auswertung weiterer Aufgaben auch bestätigt. Nun stellt sich die Frage, ob dieser Unterschied mit den zufällig der MFacets Gruppe zugeordneten Probanden oder einer Eigenheit der MFacets Anwendung zu erklären ist. Die genaue Untersuchung der Logeinträge zu diesen Aufgaben deutet auf Letzteres als Grund für den Unterschied, speziell das Löschen einer Facette zwischen beiden Aufgaben. Für Aufgabe 1 ist kein Löschen einer Facette nötig, so dass der Unterschied an Hand eines Ausreißers in der MFacets Gruppe erklärt werden kann. In diesem Fall sind von diesem Probanden doppelt so viele Klicks getätigt worden als bei den restlichen Probanden. Daher ist in Aufgabe 1 nur im Mittel und nicht im Median ein Unterschied zu verzeichnen. Die meisten Probanden (75% bei MFacets, 67% bei MobileFacets) haben sich

entschlossen, zwischen den beiden Aufgaben bewusst die Facette *Cafes* zu löschen. In MFacets lässt sich eine Facette entweder durch direktes Abwählen oder über die Facetteninformationsliste, in MobileFacets entweder implizit über die Zurück-Taste (nicht unter den 67%) oder durch den Löschen-Button entfernen. In zusätzlichen Klicks bedeutet dies für das direkte Abwählen in MFacets und die Verwendung des Löschen-Buttons in MobileFacets jeweils einen Extraklick. Ein Löschen über die Facetteninformationsliste hingegen verbraucht drei gesonderte Klicks, nämlich jeweils ein Klick zum Herunterziehen der Leiste, zum Löschen der Facette und zum Hochziehen der Leiste. Dieser Umstand sorgt dafür, dass ein signifikanter Unterschied in Aufgabe 2 zu verzeichnen ist.

Die Hypothese zu Aufgabe 3, dass die volle Liste von MFacets schneller zum Ziel als die leere Liste von MobileFacets führt, wird durch die Zeit direkt und durch die Klickzahl fast bestätigt. In Aufgabe 3 findet man auch die einzige Aufgabe, in der MFacets eine geringere Klickzahl als MobileFacets benötigt, was allerdings direkt mit der Hypothese zu erklären ist. Dass die Probanden der MobileFacets Gruppe dennoch im Mittel 24% und im Median 33% mehr Klicks zu verbuchen haben, liegt daran, dass 92% dieser Probanden im Ergebnislisten-Tab die leere Ergebnisliste überprüft haben und/oder bei der Wahl einer passenden Facette mehrere Facetten untersucht haben.

Die Aufgaben 4 und 7 besitzen dieselbe Hypothese, dass das Browsen im Grid von MFacets schneller als in der Liste von MobileFacets ist. Bezogen auf die Zeit gilt dies nur für Aufgabe 4 und betrachtet man die Klickzahlen, dann widerlegt dies besonders Aufgabe 7, was aber unter Berücksichtigung der Baseline Tests zu erwarten ist. Dass der Zeitaufwand für Aufgabe 7 bei beiden Anwendungen fast gleich ausfällt, steht wohl direkt im Zusammenhang mit den gesuchten Facetten. Im Gegensatz zu Aufgabe 4 sind hier die Facetten so ausgewählt worden, dass sie schwieriger zu finden sind. Folglich kann man daraus schließen, dass für die Suche nach einer Facette unerheblich ist, ob die Facetten in einem Grid oder in einer Liste angeordnet sind, wenn man nicht weiß, wo genau zu suchen ist. Dies kann wiederum auch mit der vorhandenen Datenqualität zusammenhängen, wenn Facetten über unerwartete Pfade zu erreichen sind. Aus Aufgabe 4 lässt sich jedoch ableiten, dass MFacets einen Zeitvorteil mit sich bringt, wenn dem Benutzer klar ersichtlich ist, auf welchem Weg eine gesuchte Facette zu finden ist.

Aufgabe 5 und 6 haben ähnliche Hypothesen indem sie aussagen, dass das Finden von Facetten unter vielen bzw. sehr vielen Facetten bei MFacets schneller als bei MobileFacets geschieht. Bezüglich der Zeit gilt dies noch nicht für Aufgabe 5, aber in Aufgabe 6 kommt der Vorteil von MFacets durch die Darstellung von mehr Facetten auf dem Bildschirm deutlich zum Vorschein. Dass bei Aufgabe 5 die Probanden der MobileFacets Gruppe zwar durchschnittlich, aber nicht signifikant weniger Zeit benötigen, ist damit zu erläutern, dass das notwendige Scrollen die Zeitdauer weniger beeinflusst als in Aufgabe 6. Das Öffnen der More-Facets-Ansicht in MFacets und das erstmalige Zurechtfinden in dieser Ansicht, vergrößert sogar die benötigte Zeit.

Die More-Facets-Ansicht ist auch ein Grund dafür, dass in MFacets doppelt so viele Klicks als in MobileFacets notwendig sind. Während in MobileFacets zum Auswählen einer Facette nur das Scrollen und ein Klick benötigt werden, sind in MFacets für das Öffnen der More-Facets-Ansicht, die Auswahl der Facette und das Schließen der More-Facets-Ansicht insgesamt drei Klicks für den selben Vorgang erforderlich. Durch diese Aufgaben lässt sich somit aussagen, dass MFacets bei der Suche nach einer Facette unter vielen Facetten weniger Zeit, aber dafür deutlich mehr Klicks verbraucht.

An Hand dieser Aufgaben kann man bezüglich der Effizienz schlussfolgern, dass bei MFacets für die Aufgabenbearbeitung maximal genau so viel und unter bestimmten Voraussetzungen weniger Zeit vonnöten ist als bei MobileFacets. Allerdings ist dies nur möglich, wenn für dieselbe Bearbeitung mehr Klicks getätigt werden. Wie sehr man nun den Aspekt der Zeit oder der Klickzahl für die Effizienz gewichtet, ist schwer zu definieren. Im Folgenden werden diese und weitere Fragestellungen bei der Interpretation der Bewertungen der Aussagen aufgegriffen.

Schlussfolgerungen für die Zufriedenheit

Um die Frage zu vertiefen, ob die Zeit oder die Klickzahl für die Effizienz wichtiger sei, kann man zumindest die Bewertung der Aussagen zu diesem Aspekt näher begutachten. Dies sind die Aussagen #2 *„Die Anwendung erfordert keine überflüssigen Eingaben.“* und #8 *„Ich empfinde die Anzahl an erforderlichen Arbeitsschritte als angemessen.“*. Wie man in Tabelle 5.3 erkennt, gibt es bei diesen Aussagen keinen signifikanten Unterschied zwischen den Anwendungen und in diesem Zusammenhang sind die Probanden mit MFacets im Durchschnitt sogar ein wenig zufriedener. Daraus lässt sich schließen, dass die Probanden mit MFacets zwar mehr Klicks getätigt haben, dies sich aber nicht negativ auf ihre Zufriedenheit im Vergleich zu MobileFacets ausgewirkt hat.

Der signifikante Unterschied pro MFacets bei Aussage #6 *„Die Anwendung enthält alle für die Aufgaben benötigten Funktionen.“* und die positive Tendenz zu MFacets in #13 *„Es ist einfach, die zur Bewältigung der Aufgaben benötigte(n) Facette(n) zu finden.“* lassen sich gut mit den freien Kommentaren bezüglich des Wunsches nach einer Möglichkeit der Suche nach Facetten, z.B. über ein Texteingabefeld, kombinieren. Während diesen Wunsch nur 17% der Probanden bei MFacets geäußert haben, sind dies bei MobileFacets 67% gewesen. Dieser Wunsch tritt hauptsächlich bei MobileFacets auf, da die Scrollphasen in Aufgabe 5 und Aufgabe 6 länger andauern als in MFacets, so dass eine Abkürzung über ein entsprechendes Suchfeld eher erwünscht ist. Der Wunsch nach einer alphabetischen Sortierung ist in beiden Gruppen bei 33% der Probanden geäußert worden, so dass hier kein Unterschied zu verzeichnen ist.

Die Aussagen #11 *„Das Abwählen/Löschen von markierten Facetten ist intuitiv.“*, #14 *„Ich weiß nie, welche Facetten ich bereits ausgewählt habe.“*

und #26 „*Es ist für mich einfach, zwischen unterschiedlichen Bearbeitungsbildschirmen zu wechseln.*“ fallen alle zu Gunsten von MobileFacets aus, was hauptsächlich durch die Facetteninformationsliste zu erklären ist. Für einige Probanden, speziell die ohne Android Erfahrung, ist das Prinzip einer herunterziehbaren Leiste ungewohnt, was sich auch in den Wünschen nach einer größeren und deutlicheren Leiste oder nach einer direkten Abwahl der Facette auf dem Bildschirm wie bei MobileFacets widerspiegelt. Da jedoch die MFacets spezifische Aussage #29 „*Die Verwendung von Facetten und Ergebnisliste auf demselben Bildschirm ist intuitiv.*“ äußerst positiv bewertet worden ist, lässt sich wegen der limitierten Größe des Bildschirms eines Smartphones schwer ein Kompromiss finden. Eventuell kann hier eine ausgiebigere Einführung in die Anwendung Abhilfe schaffen. Zu Aussage #26 lässt sich noch aussagen, dass die Bewertung auch mit dem Wunsch von 25% der Probanden nach einem Button zum Öffnen der Ergebnisliste übereinstimmt.

Die Aussagen #27 „*Vor der Ausführung möglicherweise problematischer Aktionen gibt die Anwendung eine Warnung aus.*“ und #28 „*Bei meiner Arbeit mit der Anwendung treten häufig Systemfehler, z.B. Absturz, auf.*“ weisen zwar einen (fast) signifikanten Unterschied für MobileFacets auf, allerdings sind die Unterschiede auf der Likert-Skala so gering, dass sie ignoriert werden können.

An dieser Stelle soll auch noch die Aussage #15 „*Die Anordnung der Facetten als Grid/Liste ist angemessen.*“ erwähnt werden. In Tabelle 5.3 sieht man, dass es zwar keinen signifikanten Unterschied gibt, aber im Mittel sind beide Anwendungen gleich und im Median MFacets etwas besser bewertet worden, so dass die vielleicht ungewohnte Anordnung der Facetten in einem Grid eher positiv als negativ angesehen werden kann.

Schlussfolgerungen für die Benutzbarkeit

Abschließend ist festzuhalten, dass kein eindeutiges Urteil zum Vergleich der Benutzbarkeit beider mobilen Anwendungen gefällt werden kann. Bezüglich der Effektivität ist kein Unterschied zwischen den Anwendungen erkennbar, so dass sowohl MFacets als auch MobileFacets zur facettierten Suche geeignet sind. Die Effizienz hingegen ist schwerer zu bewerten. Mit MFacets lassen sich Aufgaben zeitlich schneller bearbeiten als mit MobileFacets, dennoch sind für diesen Vorteil mehr Klicks notwendig. Allerdings hat die Auswertung der Zufriedenheit gezeigt, dass die höheren Klickzahlen nicht negativ auffallen, so dass man unter Berücksichtigung dieses Aspekts MFacets als effizienter bezeichnen kann. Allgemein ist die Zufriedenheit bei beiden Anwendungen als eher positiv zu bewerten. Dennoch fällt auf, dass die ungewohnte Leiste mit Zugang zur Facetteninformationsliste in MFacets weniger positiv angesehen werden kann als die direkte Darstellung der ausgewählten Facetten auf dem Bildschirm in MobileFacets. Dafür sind die Probanden bei der Suche nach einer Facette mit MFacets zufriedener als mit MobileFacets.

Somit kann MFacets bezüglich der Benutzbarkeit etwas besser als MobileFacets angesehen werden, da mit MFacets die Aufgaben zeitlich schneller bearbeitet worden sind und die Probanden Facetten einfacher finden können als in MobileFacets.

Fazit, Ausblick und Danksagung

In dieser Diplomarbeit ist die facettierte Suche zunächst an Hand von verwandten Arbeiten beschrieben worden. Zum einen ist gezeigt worden, wie man Daten in Form von unterschiedlichen Taxonomien klassifizieren kann. Zum anderen ist die generelle Umsetzung von Prinzipien und Funktionalitäten einer facettierten Suche diskutiert und unter Zuhilfenahme von verschiedenen Beispielanwendungen verdeutlicht worden.

Danach sind die Anwendungen FaThumb und MobileFacets detailliert analysiert worden, um die Anforderungen für MFacets zu definieren. Diese Anforderungen beziehen sich einerseits auf die Benutzeroberfläche, eher aus FaThumb stammend, und andererseits auf Funktionalitäten und die genaue Implementation, eher aus MobileFacets kommend. Außerdem sind die Unterschiede zwischen den einzelnen Anwendungen aufgezeigt worden.

Anschließend ist das gesamte MFacets-System im Detail vorgestellt worden. Dazu gehört sowohl die Betrachtung der Architektur, als auch die Darstellung des Clients und des Servers. Die Anwendung MFacets ist nach dem Model-View-Controller-Prinzip aufgebaut und dementsprechend dargestellt worden. Hierzu ist zuerst das Datenmodell, auf dem die facettierte Suche beruht, erklärt worden, worauf die Entwicklung der Benutzeroberfläche unter Verwendung von Mockups in zwei Versionen und finalen Screenshots illustriert worden ist. Um die Erläuterung des Clients abzuschließen, ist zuletzt die Programmsteuerung erklärt worden, welche die Benutzerinteraktionen regelt und die Kommunikation zwischen Client und Server abwickelt. Danach ist der MFacets-Server näher betrachtet worden, welcher die Clients mit den richtigen Informationen aus den Datenrepositories versorgt. Hier sind die jeweiligen Datenrepositories kurz vorgestellt worden bevor die beiden Webservices zur Facettenstruktur und zur Bereitstellung der Ressourcen beschrieben worden sind. Nach der Vorstellung des MFacets-Systems ist ein Abschnitt zur Vorbereitung der Anwendungen auf die Evaluation gefolgt, so dass die Anwendungen eine erfolgreiche Durchführung der Evaluation gewährleisten können.

Im letzten Kapitel dieser Diplomarbeit ist die summative Evaluation zwischen MFacets und MobileFacets bezüglich der Benutzbarkeit erläutert worden. Hierzu ist zunächst die Konzeption der Evaluation mit dem Aufstellen von Hypothesen im Aufgabenteil und der Beschreibung der Aussagen im Kommentarteil vorgestellt worden. Diese Konzeption beruht darauf, dass die Benutzbarkeit an Hand von Effektivität, Effizienz und Zufriedenheit zu messen ist. Daraufhin ist die Durchführung der Evaluation dargelegt worden, um zu zeigen, dass alle Probanden dieselben Bedingungen und Voraussetzungen vorgefunden haben. Zum Schluss sind die Messungen dieser Evaluation ausgewertet und sinngemäß interpretiert worden, so dass man zur Schlussfolgerung gelangt ist, dass MFacets bezüglich der Benutzbarkeit von den Probanden besser als MobileFacets bewertet worden ist.

Um einen Ausblick auf eventuelle zukünftige Arbeiten zu geben, ist es möglich, weitere Datenrepositories anzubinden, um unter anderem auch weitere Taxonomien für die facettierte Suche anzubieten. Je nach Datensatz lässt sich auch die Verknüpfung von Facetten vom ODER-Prinzip auf das UND-Prinzip abändern, was allerdings auch eine Änderung der Anwendung in der Facettennavigation benötigt, um die Anzeige einer leeren Ergebnisliste zu vermeiden. Außerdem können weitere Bestandteile einer facettierten Suche für mobile Anwendungen umgesetzt werden, wie z.B. die Verwendung von Drop-Down Listen, um deren Benutzbarkeit feststellen zu können. Um die Benutzbarkeit von MFacets oder MobileFacets zu erhöhen, können die Kommentare der Probanden berücksichtigt werden. Weiter kann die Konzeption der Evaluation als Grundlage für weitere Evaluationen zur Benutzbarkeit dienen, insbesondere die Verwendung der kontinuierlichen Likert-Skala um spezielle statistische Analysen durchführen zu können.

Zum Schluss möchte der Autor dieser Diplomarbeit noch seinen Dank an bestimmte Personen zum Ausdruck bringen. Ein Dank geht an Jun.-Prof. Dr. Ansgar Scherp für die hervorragende Betreuung während der Diplomarbeit. Danke auch an Kurt und Nilda Schneider für die kontinuierliche Unterstützung in jeglichen Lebensabschnitten und an Ralf Schneider für die stetige Hilfe und als erste Testperson in allen Phasen der Diplomarbeit. Außerdem sei noch Connie Alipio für ihre unendliche Geduld und Unterstützung in den letzten sechs Jahre gedankt.

Literaturverzeichnis

- BIT11a. BITKOM. Das mobile Internet boomt, 07.02.2011. http://www.bitkom.org/de/presse/8477_66799.aspx, Letzter Zugriff: 14.11.2011.
- BIT11b. BITKOM. Zahl der App-Downloads explodiert, 14.02.2011. http://www.bitkom.org/de/presse/8477_66877.aspx, Letzter Zugriff: 14.11.2011.
- BLCC⁺06. Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop, SWUI 2006, Athens, GA, USA, November 6, 2006*, 2006.
- DFW08. Raimund Dachzelt, Mathias Frisch, and Markus Weiland. FacetZoom: A Continuous Multi-Scale Widget for Navigating Hierarchical Metadata. In *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, Florence, Italy, April 5-10, 2008*, pages 1353–1356. ACM, 2008.
- EHTA11. Timofey Ermilov, Norman Heino, Sebastian Tramp, and Sören Auer. OntoWiki Mobile - Knowledge Management in Your Pocket. In *Proceedings of the 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011*, volume 6643 of *Lecture Notes in Computer Science*, pages 185–199. Springer, 2011.
- Fer08. Sébastien Ferré. Agile Browsing of a Document Collection with Dynamic Taxonomies. In *Proceedings of the 19th International Workshop on Database and Expert Systems Applications (DEXA 2008), 1-5 September 2008, Turin, Italy*, pages 377–381. IEEE Computer Society, 2008.
- FZEB07. Ismail Fahmi, Junte Zhang, Henk Ellermann, and Gosse Bouma. SWHi System Description: A Case Study in Information Retrieval, Inference, and Visualization in the Semantic Web. In *Proceedings of the 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007*, volume 4519 of *Lecture Notes in Computer Science*, pages 769–778. Springer, 2007.
- Gar11. Gartner. Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010, 09.02.2011. <http://www.gartner.com/it/page.jsp?id=1543014>, Letzter Zugriff: 14.11.2011.

- GH99. Günther Gediga and Kai-Christoph Hamborg. IsoMetrics: An usability inventory supporting summative and formative evaluation of software systems. In *Proceedings of the 8th International Conference on Human-Computer Interaction: Ergonomics and User Interfaces, HCI International '99, Munich, Germany, August 22-26, 1999*, volume 1, pages 1018–1022. Lawrence Erlbaum, 1999.
- GM08. Karen Grace-Martin. Can Likert Scale Data ever be Continuous?, 21.10.2008. <http://karengracemartin.articlealley.com/can-likert-scale-data-ever-be-continuous-670606.html>, Letzter Zugriff: 14.11.2011.
- GMR07. Fredrik Gundelsweiler, Thomas Memmel, and Harald Reiterer. ZEUS - Zoomable Explorative User Interface for Searching and Object Presentation. In *Proceedings of Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design, Symposium on Human Interface 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, 2007*, volume 4557 of *Lecture Notes in Computer Science*, pages 288–297. Springer, 2007.
- Hea06. Marti A. Hearst. Design recommendations for hierarchical faceted search interfaces. *ACM SIGIR Workshop on Faceted Search*, 2006.
- Hea08. Marti A. Hearst. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. In *Proceedings of the Second Workshop on Human-Computer Interaction and Information Retrieval, HCIR08, Redmond, WA, USA, October 23, 2008*, pages 13–17, 2008.
- HEE⁺02. Marti A. Hearst, Ame Elliott, Jennifer English, Rashmi R. Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the flow in web site search. *Communications of the ACM*, 45(9):42–49, 09.2002.
- HEZ10. Philipp Heim, Thomas Ertl, and Jürgen Ziegler. Facet Graphs: Complex Semantic Querying Made Easy. In *Proceedings of the 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010*, volume 6088 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2010.
- HHMT09. Peter Haase, Daniel Herzig, Mark A. Musen, and Thanh Tran. Semantic Wiki Search. In *Proceedings of the 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009*, volume 5554 of *Lecture Notes in Computer Science*, pages 445–460. Springer, 2009.
- HvOH06. Michiel Hildebrand, Jacco van Ossenbruggen, and Lynda Hardman. /facet: A Browser for Heterogeneous Semantic Web Repositories. In *Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 272–285. Springer, 2006.
- KG06. Amy K. Karlson and George G. Robertson and Daniel C. Robbins and Mary Czerwinski and Greg Smith. FaThumb: a facet-based interface for mobile search. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, pages 711–720. ACM, 2006.
- Kle10. Alexander Kleinen. Facettiertes Browsen von semantischen Daten auf einem mobilen Endgerät. Diplomarbeit, University of Koblenz-Landau, 2010.

- Kur11. Jürgen Kuri. Ericsson: rasanter Anstieg des mobilen Datenverkehrs, 07.11.2011. <http://www.heise.de/newsticker/meldung/Ericsson-rasanter-Anstieg-des-mobilen-Datenverkehrs-1374494.html>, Letzter Zugriff: 14.11.2011.
- LFH10. Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction*. Wiley, 2010.
- LSR⁺09. Bongshin Lee, Greg Smith, George G. Robertson, Mary Czerwinski, and Desney S. Tan. FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 1293–1302. ACM, 2009.
- MDD09. Yevgeniy Eugene Medynskiy, Mira Dontcheva, and Steven M. Drucker. Exploring Websites through Contextual Facets. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 2013–2022. ACM, 2009.
- OR06. Eyal Oren and Renaud Delbru and Stefan Decker. Extending Faceted Navigation for RDF Data. In *Proceedings of the 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 559–572. Springer, 2006.
- Pol09. Jan Polowinski. Widgets for faceted browsing. In *Proceedings of the Human Interface and the Management of Information. Designing Information Environments, Symposium on Human Interface 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009*, volume 5617 of *Lecture Notes in Computer Science*, pages 601–610. Springer, 2009.
- Ran65. Shiyali Ramamrita Ranganathan. *The Colon Classification*. Rutgers series on systems for the intellectual organization of information. Graduate School of Library Service, Rutgers, the State University, 1965.
- Rob11. Judy Robertson. Stats: We're Doing It Wrong, 04.04.2011. <http://cacm.acm.org/blogs/blog-cacm/107125-stats-were-doing-it-wrong/>, Letzter Zugriff: 14.11.2011.
- SNE06. Stefan Steiniger, Moritz Neun, and Alistair Edwardes. Foundations of Location Based Services Lesson 1 CartouChe - Lecture Notes on LBS, V. 1.0, 2006.
- Ss08. D. A. Smith and m. schraefel. Interactively using Semantic Web knowledge: Creating scalable abstractions with FacetOntology. Technical report, School of Electronics and Computer Science, University of Southampton, 2008. (Unpublished).
- ST09. Giovanni Maria Sacco and Yannis Tzitzikas. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*. The Information Retrieval Series. Springer, 2009.
- SVH07. Osmo Suominen, Kim Viljanen, and Eero Hyvönen. User-Centric Faceted Search for Semantic Portals. In *Proceedings of the 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007*, volume 4519 of *Lecture Notes in Computer Science*, pages 356–370. Springer, 2007.

- TAP08. Yannis Tzitzikas, Nikos Armenatzoglou, and Panagiotis Papadakis. FleXplorer: A Framework for Providing Faceted and Dynamic Taxonomy-Based Information Exploration. In *Proceedings of the 19th International Workshop on Database and Expert Systems Applications (DEXA 2008), 1-5 September 2008, Turin, Italy*, pages 392–396. IEEE Computer Society, 2008.
- TLH11. Jouni Tuominen, Nina Laurenne, and Eero Hyvönen. Biological Names and Taxonomies on the Semantic Web - Managing the Change in Scientific Conception. In *Proceedings of the 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011*, volume 6644 of *Lecture Notes in Computer Science*, pages 255–269. Springer, 2011.
- Tun09. Daniel Tunkelang. *Faceted Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2009.
- vAWvH10. Chris J. van Aart, Bob J. Wielinga, and Willem Robert van Hage. Mobile Cultural Heritage Guide: Location-Aware Semantic Search. In *Proceedings of the Knowledge Engineering and Management by the Masses - 17th International Conference, EKAW 2010, Lisbon, Portugal, October 11-15, 2010*, volume 6317 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2010.
- WRS⁺05. Max L. Wilson, Alistair Russell, Daniel A. Smith, Alisdair Owens, and m. c. schraefel. mSpace Mobile: A Mobile Application for the Semantic Web. In *Proceedings of the End User Semantic Web Workshop at the 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 7, 2005*, 2005.
- ZM04. Junliang Zhang and Gary Marchionini. Coupling Browse and Search in Highly Interactive User Interfaces: A Study of the Relation Browser++. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, JCDL 2004, Tucson, AZ, USA, June 7-11, 2004*, page 384. ACM, 2004.

A

Anhang - Evaluation

A.1 Einführung MFacets

Beschreibung der Funktionalitäten der Anwendung

ausgewählte Facetten Anzahl aktueller Ressourcen Leiste kann herunter gezogen werden

Landmark, Living People selected (28)

- Aydo Abay no distance
- Basilica of St. Castor 1.66kms
- Bodo Illgner no distance
- DB Museum, Koblenz 2.02kms
- Egon Alfred Klepsch no distance
- Electoral Palace (Koblenz) 1.6kms

Ergebnisliste

normale Facette

Facette mit Subfacetten

ausgewählte Facette Navigationsansicht

Suchfilter Sortierung Facetten öffnen

Search Alphabetical Open Facets

Datum 25-8-2011 Anzahl dargestellter POIs

Point of Interest (POI) kann ausgewählt werden

Zeitleiste

Zoom-Buttons

earlier later

Google

Einführung

Um dich mit der Anwendung etwas vertrauter zu machen, führe bitte folgende Anweisungen aus:

1. Falls noch nicht geschehen, starte MFacets.
2. Klicke auf die Facette **Place** → **Shopping** → **Food & Drink Shops** → **Supermarkets**.
3. Klicke auf die Facette **Bakeries** zum Auswählen, klicke noch einmal zum Abwählen.
4. Klicke auf den Zurück-Button.
5. Klicke lange auf die Facette **Fashion** und bestätige den Dialog.

6. Klicke so oft auf den Zurück-Button bis du die Ursprungsansicht erreichst.
7. Klicke auf die Ergebnisliste.
8. Klicke auf die Sortierung und sortiere die Ergebnisliste nach Distanz.
9. Klicke auf das Suchfeld und gebe ein „R“ ein.
10. Klicke auf den Pfeil neben der Ressource **Roots**.

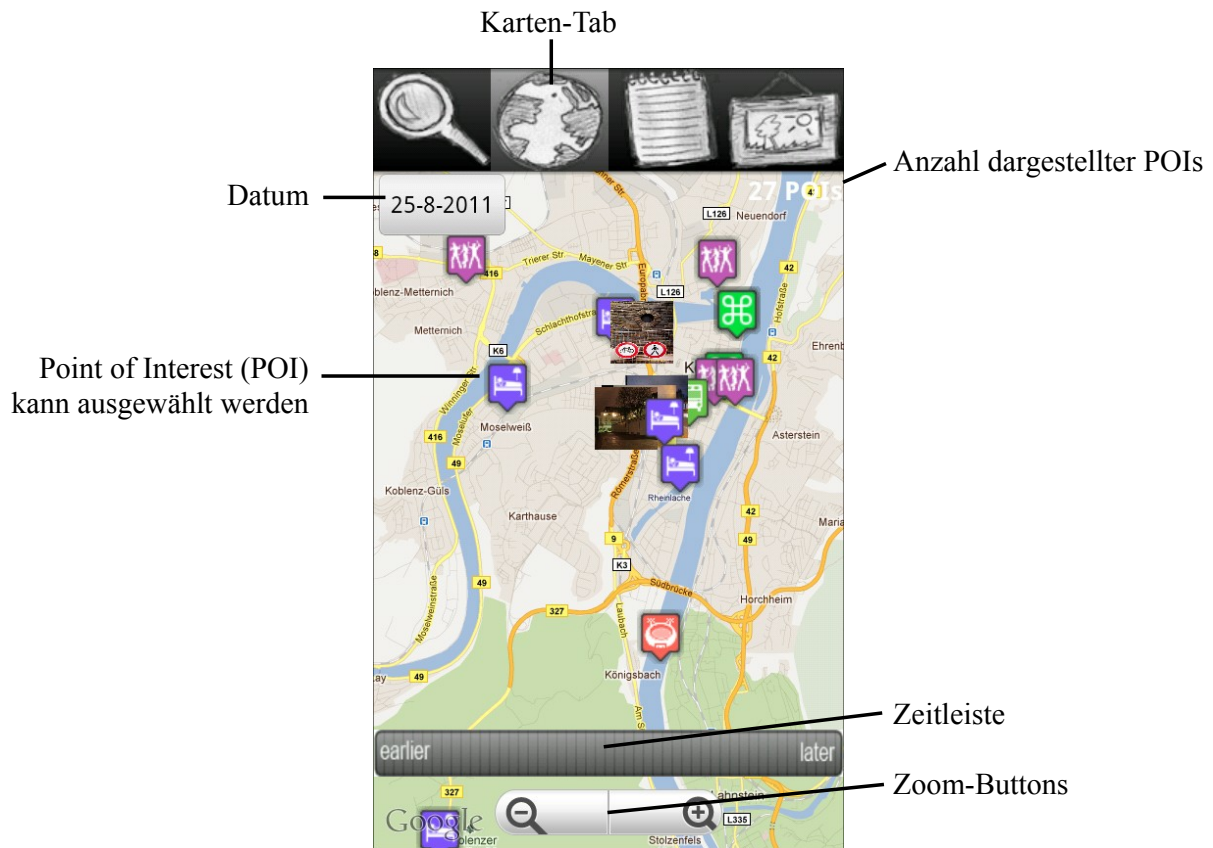
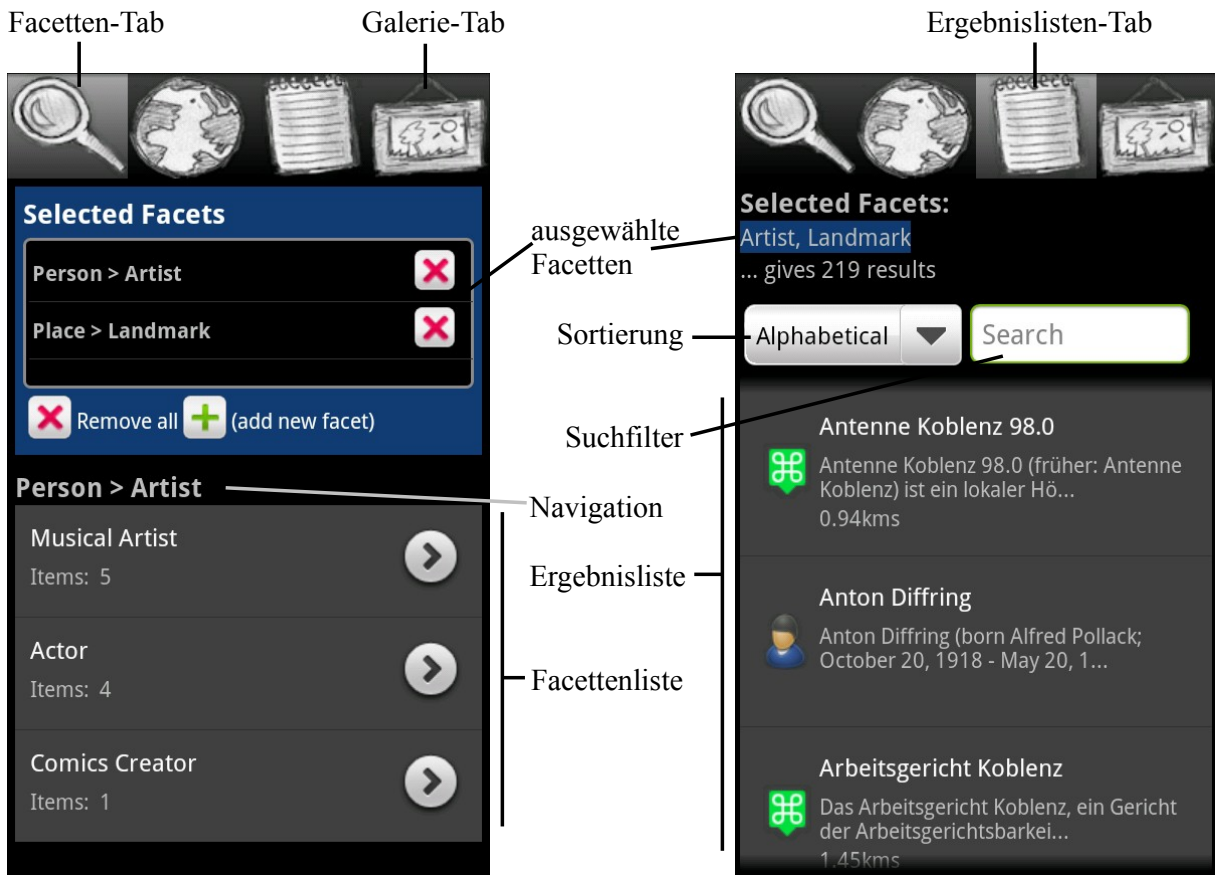
11. Klicke auf den Zurück-Button um das Fenster zu schließen.
12. Klicke auf das Suchfeld und lösche das „R“.
13. Klicke auf den Zurück-Button um die Tastatur zu schließen.
14. Klicke auf die Sortierung und sortiere die Ergebnisliste nach dem Alphabet.
15. Klicke auf den Button **Open Facets**.

16. Klicke auf den Menü-Button und wähle **Map** aus.
17. Klicke auf das obere Einkaufswagen-Symbol und danach auf den erscheinenden Text.
18. Klicke auf den Button neben dem Foto.
19. Klicke auf den Zurück-Button um zur Facettenansicht zu gelangen.
20. Ziehe die Leiste am oberen Bildschirmrand herunter.

21. Klicke auf den Löschen-Button zu **Fashion** und bestätige den Dialog.
22. Klicke auf den Löschen-Button zu **Delete all selected Facets** und bestätige den Dialog.
23. Ziehe die Leiste am unteren Bildschirmrand hinauf.

A.2 Einführung MobileFacets

Beschreibung der Funktionalitäten der Anwendung



Einführung

Um dich mit der Anwendung etwas vertrauter zu machen, führe bitte folgende Anweisungen aus:

1. Falls noch nicht geschehen, starte MobileFacets.
2. Klicke auf den Karten-Tab, dann auf den Ergebnislisten-Tab, dann auf den Facetten-Tab.
3. Klicke auf die Facette **Place** → **Shopping** → **Food & Drink Shops** → **Supermarkets**.
4. Klicke auf den Button (**add new facet**).
5. Klicke auf die Facette **Place** → **Shopping** → **Fashion** → **Leisure Wear**.

6. Klicke auf den Zurück-Button.
7. Klicke auf den Ergebnislisten-Tab.
8. Klicke auf die Sortierung und sortiere die Ergebnisliste nach Distanz.
9. Klicke auf das Suchfeld und gebe ein „R“ ein.
10. Klicke auf die Ressource **Roots**.

11. Klicke auf den Zurück-Button um das Fenster zu schließen.
12. Klicke auf das Suchfeld und lösche das „R“.
13. Klicke auf den Zurück-Button um die Tastatur zu schließen.
14. Klicke auf die Sortierung und sortiere die Ergebnisliste nach dem Alphabet.
15. Klicke auf den Karten-Tab.

16. Klicke auf das obere Einkaufswagen-Symbol und danach auf den erscheinenden Text.
17. Klicke auf den Button neben dem Foto.
18. Klicke auf den Facetten-Tab.
19. Klicke auf den Löschen-Button zu **Fashion** und bestätige den Dialog.
20. Klicke auf den Button **Remove all** und bestätige den Dialog.

A.3 Aufgaben

Szenario

Du bist Mitarbeiter der Münchener Software AG und geschäftlich in Koblenz unterwegs. Nach einem Messebesuch hast du noch viel Zeit, um Koblenz zu erkunden und Interessen nachzugehen. Dabei wird dir die Anwendung MFacets behilflich sein.

1. Zuerst möchtest du das nächst gelegene Cafe aufsuchen. Wähle dazu die Facette **Place** → **Eating And Drinking** → **Cafes & Coffee Shops** → **Cafes** aus. Wie lautet das am nächsten gelegene Cafe und wie weit ist es entfernt?

Antwort: _____

2. Nachdem du im Cafe angekommen bist und dir etwas zu trinken bestellt hast, machst du dir Gedanken, wo du zu Abend essen möchtest. Du erinnerst dich an eine Empfehlung für das mexikanische Restaurant **Casa Mexicana**. Um es zu finden, wähle geeignete Facetten aus. Wie lautet die Adresse zum Restaurant?

Antwort: _____

3. Dir fällt gerade ein, dass es in Koblenz eine Basilika gibt, aber der genaue Name ist dir jetzt entfallen. Um den genauen Namen heraus zu finden, entferne zunächst alle ausgewählten Facetten. Danach stelle sicher, dass die Ergebnisliste Einträge enthält und verwende danach den Suchfilter neben der Ergebnisliste und suche nach **Basilika**. Wie heißt der gesamte Name?

Antwort: _____

4. Du interessierst dich für weitere Örtlichkeiten in Koblenz. Finde durch Browsen durch die Facetten jeweils eine Örtlichkeit der folgenden Kategorien: Buchgeschäft, 3-Sterne Hotel, Fast Food Geschäft und Einkaufszentrum. Wie heißen die Örtlichkeiten?

Buchgeschäft: _____

3-Sterne Hotel: _____

Fast Food Geschäft: _____

Einkaufszentrum: _____

5. Du erinnerst dich an den bekannten Koblenzer **Max von Laue**, aber du hast vergessen welchen Beruf er ausgeübt hat. Du entscheidest dich über ein Ausschlussverfahren den Beruf zu erfahren. Deshalb überprüfst du, ob Max von Laue im **Militär (Military Person)**, als **Fußballer (German Footballers)** oder als **Physiker (German Physicists)** tätig war. Welchen dieser drei Berufe hat Max von Laue ausgeübt?

Antwort: _____

6. An dieser Stelle interessierst du dich für weitere bekannte Persönlichkeiten. Finde jeweils eine Person zu folgenden Berufen: **Architekt (Architect)**, **Fußballmanager (German Football Managers)** und **Sportschütze (Olympic Shooters Of Germany)**. Wie lauten die Namen der Personen?

Architekt: _____

Fußballmanager: _____

Sportschütze: _____

7. Als letztes möchtest du noch die nähere Umgebung erkunden. Suche nach folgenden, am nächsten gelegene Örtlichkeiten: Bar, Diskothek, Videoverleihgeschäft und Vegetarisches Restaurant. Wie heißen die Örtlichkeiten und wie weit sind sie entfernt?

Bar: _____

Diskothek: _____

Videoverleihgeschäft: _____

Vegetarisches Restaurant: _____

A.4 Aussagen

Bewertung von Aussagen

1. Es hat lange gedauert bis ich die Bedienung der Anwendung erlernt habe.
trifft nicht zu | _____ | trifft voll zu
2. Die Anwendung erfordert keine überflüssigen Eingaben.
trifft nicht zu | _____ | trifft voll zu
3. Die Anwendung unterstützt mich angemessen in der Durchführung der Aufgabe.
trifft nicht zu | _____ | trifft voll zu
4. Die Interaktion mit der Anwendung verhält sich entsprechend meiner Erwartungen.
trifft nicht zu | _____ | trifft voll zu
5. Die von der Anwendung gelieferten Suchergebnisse entsprachen meinen Erwartungen.
trifft nicht zu | _____ | trifft voll zu
6. Die Anwendung enthält alle für die Aufgaben benötigten Funktionen.
trifft nicht zu | _____ | trifft voll zu
7. Die Durchführung der einzelnen Arbeitsschritte ist **nicht** intuitiv.
trifft nicht zu | _____ | trifft voll zu
8. Ich empfinde die Anzahl an erforderlichen Arbeitsschritten als angemessen.
trifft nicht zu | _____ | trifft voll zu
9. Das Auswählen einer Facette ist **nicht** intuitiv.
trifft nicht zu | _____ | trifft voll zu
10. Das Auswählen mehrerer Facetten ist einfach.
trifft nicht zu | _____ | trifft voll zu
11. Das Abwählen/Löschen von markierten Facetten ist intuitiv.
trifft nicht zu | _____ | trifft voll zu
12. Die Navigation durch die Facetten ist intuitiv.
trifft nicht zu | _____ | trifft voll zu
13. Es ist einfach, die zur Bewältigung der Aufgaben benötigte(n) Facette(n) zu finden.
trifft nicht zu | _____ | trifft voll zu
14. Ich weiß **nie**, welche Facetten ich bereits ausgewählt habe.
trifft nicht zu | _____ | trifft voll zu
15. Die Anordnung der Facetten als Grid ist angemessen.
trifft nicht zu | _____ | trifft voll zu

16. Das Verfeinern der Suche über die Facetten ist für mich nachvollziehbar.
trifft nicht zu | _____ | trifft voll zu
17. Die Suche nach Ressourcen in der Ergebnisliste ist intuitiv.
trifft nicht zu | _____ | trifft voll zu
18. Auf dem Bildschirm finde ich alle Informationen, die ich gerade benötige.
trifft nicht zu | _____ | trifft voll zu
19. Die Darstellung der Informationen auf dem Bildschirm unterstützt mich bei der Bearbeitung meiner Aufgaben.
trifft nicht zu | _____ | trifft voll zu
20. Die Funktionen, die zur Bewältigung der Aufgaben dienen, sind einfach zu finden.
trifft nicht zu | _____ | trifft voll zu
21. Die Anwendung erschwert meine Aufgabenbearbeitung durch eine uneinheitliche Gestaltung.
trifft nicht zu | _____ | trifft voll zu
22. Die von der Anwendung verwendeten Begriffe sind für mich **nicht** sofort verständlich.
trifft nicht zu | _____ | trifft voll zu
23. Es ist für mich unmittelbar ersichtlich, was die Interaktionselemente der Anwendung bewirken.
trifft nicht zu | _____ | trifft voll zu
24. Ich kann die Rückmeldungen, die ich von der Anwendung erhalte, eindeutig dem auslösenden Vorgang zuordnen.
trifft nicht zu | _____ | trifft voll zu
25. Ich war mir bei den Wartezeiten immer sicher, dass die Anwendung weiterarbeitet.
trifft nicht zu | _____ | trifft voll zu
26. Es ist für mich einfach, zwischen unterschiedlichen Bearbeitungsbildschirmen zu wechseln.
trifft nicht zu | _____ | trifft voll zu
27. Vor der Ausführung möglicherweise problematischer Aktionen gibt die Anwendung eine Warnung aus.
trifft nicht zu | _____ | trifft voll zu
28. Bei meiner Arbeit mit der Anwendung treten häufig Systemfehler, z.B. Absturz, auf.
trifft nicht zu | _____ | trifft voll zu
29. Die Verwendung von Facetten und Ergebnisliste auf demselben Bildschirm ist intuitiv.
trifft nicht zu | _____ | trifft voll zu

A.5 Logstruktur für MFacets

```

1 Boolean := "true" | "false";
2 String := StringExpression;
3 Digits := ( NonZeroDigit { Digit } ) | Digit;
4 NonZeroDigit := "1" | ... | "9";
5 Digit := "0" | ... | "9";
6
7
8 MFacetsLogEntry := Timestamp ":" Action;
9
10 Timestamp := Year "/" Month "/" Day "-" Hour ":" Minute ":"
    Second;
11 Year := Digit Digit Digit Digit;
12 Month := ( "0" | "1" ) Digit;
13 Day := ( "0" | "1" | "2" | "3" ) Digit;
14 Hour := ( "0" | "1" | "2" ) Digit;
15 Minute := ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit;
16 Second := ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit;
17
18 Action := Search | Location | Resources | Facets | Click |
    Filter;
19
20 Search := LocationSearch | KeywordSearch | SearchParameter;
21 LocationSearch := "Location search for following location:";
22 KeywordSearch := "Keyword search with: " Keyword;
23 SearchParameter := "DBpedia search: " Boolean ", Events
    search: " Boolean ", Qype search: " Boolean;
24 Keyword := String;
25
26 Location := TopLeftLocation | CenterLocation |
    BottomRightLocation | DistanceEntry;
27 TopLeftLocation := "Coordinates top left corner: "
    Coordinate;
28 CenterLocation := "Coordinates center: " Coordinate;
29 BottomRightLocation := "Coordinates bottom right corner: "
    Coordinate;
30 DistanceEntry := "Distance top left to bottom right corner: "
    Distance;
31 Coordinate := Latitude " - " Longitude;
32 Latitude := Digits "." Digits;
33 Longitude := Digits "." Digits;
34 Distance := Digits "." Digits " kms";
35
36 Resources := TotalResourcesRetrieved |
    DBpediaResourcesRetrieved | EventResourcesRetrieved |
    QypeResourcesRetrieved;
37

```

```

38 TotalResourcesRetrieved := ResourceCount " resources
   retrieved";
39 DBpediaResourcesRetrieved := DBpediaResourceCount "
   resources from DBpedia retrieved";
40 EventResourcesRetrieved := EventResourceCount " resources
   from Events retrieved";
41 QypeResourcesRetrieved := QypeResourceCount " resources from
   Qype retrieved";
42 ResourceCount := Digits;
43 DBpediaResourceCount := Digits;
44 EventResourceCount := Digits;
45 QypeResourceCount := Digits;
46
47 Facets := TotalFacetsCount | FacetsResourceCount;
48
49 TotalsFacetCount := FacetCount " facets retrieved, "
   DBpediaFacetCount " from DBpedia, " EventFacetCount "
   from Events, " QypeFacetCount " from Qype";
50 FacetsResourceCount := JSONFacet;
51
52 FacetCount := Digits;
53 DBpediaFacetCount := Digits;
54 EventFacetCount := Digits;
55 QypeFacetCount := Digits;
56
57 JSONFacet := "{" JSONName ":" JSONString ","
   JSONResourceCount ":" JSONDigits [ "," JSONChildren ":[ "
   JSONFacet { "," JSONFacet } "]" ] }";
58 JSONName = "\" facetname \";
59 JSONResourceCount = "\" count \";
60 JSONChildren = "\" children \";
61 JSONString = "\" String \";
62 JSONDigits = "\" Digits \";
63
64 Click := FacetInformationListClick | ResourceListClick |
   ResourceClick | FacetTableClick | ButtonClick |
   MenuClick | BackKeyClick | SearchKeyClick;
65
66 FacetInformationListClick := OpenFacetInformationList |
   FacetInformationListElementClick |
   CloseFacetInformationList;
67
68 OpenFacetInformationList := "Open FacetInformationList, "
   SelectedFacetCount " facets selected, "
   DisplayedFacetCount " facets displayed";
69 FacetInformationListElementClick := ( Facet | AllFacets ) "
   deleted";

```

```

70 CloseFacetInformationList := "Close FacetInformationList, "
    SelectedFacetCount " facets selected, "
    DisplayedFacetCount " facets displayed";
71 SelectedFacetCount := Digits;
72 DisplayedFacetCount := Digits;
73 Facet := String;
74 AllFacets := "All facets";
75
76 ResourceListClick := "Open ResourceList";
77
78 ResourceClick := "Click on Resource " Resource;
79 Resource := String;
80
81 FacetTableClick := FacetClick | MoreFacetsClick;
82
83 FacetClick := "Click on Facet " Facet [ Selection ];
84 MoreFacetsClick := "Click on MoreFacets " Facet [ Selection
    ];
85 Selection := ", selected" | ", unselected";
86
87 ButtonClick := "Click on Button " Button;
88 Button := String;
89
90 MenuClick := "Click on Menu " Menu;
91 Menu := String;
92
93 BackKeyClick := "Click Back key to " State [ ", " Facet ];
94 State := String;
95
96 SearchKeyClick := "Click Search key";
97
98 Filter := ResourceListFilter | ResourceListSort |
    MoreFacetsFilter;
99 ResourceListFilter := "ResourceList filtered by: " Filter "
    -> " ResourceCount " resources";
100 ResourceListSort := "ResourceList sorted by: " Sort;
101 MoreFacetsFilter := "MoreFacets filtered by: " Filter " -> "
    FacetCount " facets";
102 Filter := String;
103 Sort := "Alphabet" | "Distance";

```

A.6 Logstruktur für MobileFacets

```

1 Boolean := "true" | "false";
2 String := StringExpression;
3 Digits := ( NonZeroDigit { Digit } ) | Digit;
4 NonZeroDigit := "1" | ... | "9";
5 Digit := "0" | ... | "9";
6
7
8 MobileFacetsLogEntry := Timestamp ":" Action;
9
10 Timestamp := Year "/" Month "/" Day "-" Hour ":" Minute ":"
    Second;
11 Year := Digit Digit Digit Digit;
12 Month := ( "0" | "1" ) Digit;
13 Day := ( "0" | "1" | "2" | "3" ) Digit;
14 Hour := ( "0" | "1" | "2" ) Digit;
15 Minute := ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit;
16 Second := ( "0" | "1" | "2" | "3" | "4" | "5" ) Digit;
17
18 Action := Search | Location | Resources | Facets | Click |
    Filter;
19
20 Search := LocationSearch | KeywordSearch | SearchParameter;
21 LocationSearch := "Location search for following location:";
22 KeywordSearch := "Keyword search with: " Keyword;
23 SearchParameter := "DBpedia search: " Boolean ", Events
    search: " Boolean ", Qype search: " Boolean;
24 Keyword := String;
25
26 Location := TopLeftLocation | CenterLocation |
    BottomRightLocation | DistanceEntry;
27 TopLeftLocation := "Coordinates top left corner: "
    Coordinate;
28 CenterLocation := "Coordinates center: " Coordinate;
29 BottomRightLocation := "Coordinates bottom right corner: "
    Coordinate;
30 DistanceEntry := "Distance top left to bottom right corner: "
    Distance;
31 Coordinate := Latitude " - " Longitude;
32 Latitude := Digits "." Digits;
33 Longitude := Digits "." Digits;
34 Distance := Digits "." Digits " kms";
35
36 Resources := TotalResourcesRetrieved |
    DBpediaResourcesRetrieved | EventResourcesRetrieved |
    QypeResourcesRetrieved;
37

```

```

38 TotalResourcesRetrieved := ResourceCount " resources
   retrieved";
39 DBpediaResourcesRetrieved := DBpediaResourceCount "
   resources from DBpedia retrieved";
40 EventResourcesRetrieved := EventResourceCount " resources
   from Events retrieved";
41 QypeResourcesRetrieved := QypeResourceCount " resources from
   Qype retrieved";
42 ResourceCount := Digits;
43 DBpediaResourceCount := Digits;
44 EventResourceCount := Digits;
45 QypeResourceCount := Digits;
46
47 Facets := TotalFacetsCount | FacetsResourceCount;
48
49 TotalsFacetCount := FacetCount " facets retrieved, "
   DBpediaFacetCount " from DBpedia, " EventFacetCount "
   from Events, " QypeFacetCount " from Qype";
50 FacetsResourceCount := JSONFacet;
51
52 FacetCount := Digits;
53 DBpediaFacetCount := Digits;
54 EventFacetCount := Digits;
55 QypeFacetCount := Digits;
56
57 JSONFacet := "{" JSONName ":" JSONString ","
   JSONResourceCount ":" JSONDigits [ "," JSONChildren ":[ "
   JSONFacet { "," JSONFacet } "]" ] }";
58 JSONName = "\" facetname \";
59 JSONResourceCount = "\" count \";
60 JSONChildren = "\" children \";
61 JSONString = "\" String \";
62 JSONDigits = "\" Digits \";
63
64 Click := TabClick | FacetClick | ResourceClick | ButtonClick
   | MenuClick | BackKeyClick | SearchKeyClick;
65
66 TabClick := "Open Tab " ( "BrowseFacets" | "Map" | "
   ResultList" | "Gallery" );
67
68 FacetClick := "Click on Facet " Facet;
69 Facet := String;
70
71 ResourceClick := "Click on Resource " Resource;
72 Resource := String;
73
74 ButtonClick := NormalButtonClick | SpecialButtonClick;
75
76 NormalButtonClick := "Click on Button " Button;

```

```
77 Button := String;
78
79 SpecialButtonClick := DeleteButtonClick | AddButtonClick;
80
81 DeleteButtonClick := "Click on DeleteButton for " Facet;
82 AddButtonClick := "Click on AddButton for " Facet;
83
84 MenuClick := "Click on Menu " Menu;
85 Menu := String
86
87 BackKeyClick := "Click Back key to " State [ ", " Facet ];
88 State := String;
89
90 SearchKeyClick := "Click Search key";
91
92 Filter := ResultListFilter | ResultListSort;
93 ResultListFilter := "ResultList filtered by: " Filter " -> "
    ResourceCount " resources";
94 ResultListSort := "ResultList sorted by: " Sort;
95 Filter := String;
96 Sort := "Alphabet" | "Distance";
```