

Universität
Rostock



Traditio et Innovatio

Bachelorarbeit

Patternbasiertes Redesign von
Benutzungsoberflächen für mobile
Anwendungen – Eine Fallstudie

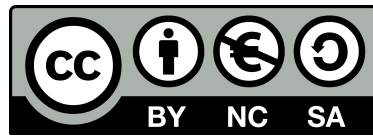
Zusammenfassung

Um den heutigen Anforderungen zu genügen, ist es häufig wünschenswert, bestehende Applikationen auch auf mobilen Geräten verfügbar zu machen. Die unterschiedlichen technischen Möglichkeiten der Geräte (zum Beispiel Beschränkungen in der Displaygröße und Batteriebetrieb) sowie Veränderungen des Anwendungskontextes lassen jedoch selten eine „1:1-Übertragung“ der Applikationen zu. Anpassungen sind notwendig.

In der Bachelorarbeit geht es insbesondere um das patternbasierte Redesign von Benutzungsoberflächen (user interfaces - UIs). Anhand einer Fallstudie sollen Adaptionmöglichkeiten für UIs erprobt werden. Ausgangspunkt in der Fallstudie ist die webbasierte Anwendung Fleetmon, die z.B. aktuelle Positionen und andere Informationen über Schiffe auf einer Karte anzeigt (www.digital-seas.com). Basierend auf einer Analyse dieser Anwendung sind in der Bachelorarbeit alternative UIs für eine mobile Anwendung zu erarbeiten und prototypisch umzusetzen. Die Lösungsfindungen sollen dabei durch Designpatterns geleitet sein. Sie sind in der Arbeit zu illustrieren und gegenüberzustellen.

Die vorliegende Arbeit wird auf Grundlage der Literaturarbeit „Patterns für die Adaption von Desktop-UIs zu UIs für mobile Geräte“ [1] geschrieben.

Erstellt von	Marcus Sümnick Matrikelnummer 7200172
Eingereicht am	4. Juli 2011
Betreuer von	Dr.-Ing. Anke Dittmar Lehrstuhl für Softwaretechnik, Universität Rostock
Geprüft von	Prof. Dr.-Ing. habil. Peter Forbrig Lehrstuhl für Softwaretechnik, Universität Rostock



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

This work is licensed under a Creative Commons Attribution 4.0 International License.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Inhaltsverzeichnis

1	EINLEITUNG.....	1
2	FLEETMON.....	2
3	BESTANDSANALYSE VON FLEETMON.....	4
3.1	VISUELLE-UND INTERAKTIONSANALYSE	4
3.2	INTERFACEELEMENTE.....	8
3.3	ERLÄUTERUNG DER PATTERNZUORDNUNG.....	10
3.4	INTERAKTIONSMODELL UND DIALOGGRAPH.....	13
4	BETRACHTEN DER ZIELUMGEBUNG.....	15
4.1	ZIELGRUPPE, ANWENDUNGS- UND AUFGABENKONTEXT.....	15
4.2	TECHNISCHE RAHMENBEDINGUNGEN.....	16
4.3	REDESIGNENTSCHEIDUNGEN.....	17
4.4	INTERAKTIONSMODELL UND DIALOGGRAPH.....	18
5	PATTERN MAPPING	20
5.1	DETAILBETRACHTUNG	21
5.2	AUSWERTUNG	31
6	ZUSAMMENFASSUNG	33
7	LITERATURVERZEICHNIS	34
8	ABBILDUNGSVERZEICHNIS.....	35
9	ANHANG.....	35

1 Einleitung

Speziell die Computertechnik für den Endverbraucher entwickelt sich in den letzten Jahren immer mehr dahin, dass sie unabhängiger von kabelgebundener Strom- und Netzwerkversorgung verwendet werden kann. Gleichzeitig wächst der Markt der Smartphones [2]. Die Nutzer dieser Geräte verlangen nach Softwarelösungen, die sie im Alltag unterstützen und ihnen gleichzeitig den Zugang zu den neuen Nutzungsmöglichkeiten der Geräte eröffnen. Beide Veränderungen haben zum Ergebnis, dass selten auf bereits bestehende Software zurückgegriffen werden kann, um sie im mobilen Einsatz zu verwenden. Funktionieren herkömmliche Desktopanwendungen auf Laptops an einem fixierten Standort noch, ohne dass der Nutzer große Einschränkungen oder Veränderungen hinnehmen muss. So werden die veränderten Rahmenbedingungen jedoch spätestens bei Convertibles, reinen Tablet-Computern oder Smartphones auffällig. Diese Geräteklassen, die darauf ausgelegt sind direkter als alle anderen bisher bekannten Computer vom Nutzer verwendet zu werden, stellen andere und neue Bedingungen an die Software als herkömmliche Desktopcomputer. Gründe hierfür sind vor allem veränderte und neue Eingabemethoden, neue sich ergebende Aufgaben sowie neue, sich durch die Mobilität der Hardware erschließende, Nutzungsorte. Direktere Verwendung bezieht sich in diesem Zusammenhang sowohl auf die Methode der Dateneingabe und Steuerung – primär mit den Fingern – als auch auf die gesamten indirekten Einflüsse wie Dauer, Situation und Intensität der Nutzung. Aus all diesen Veränderungen resultieren neue Anforderungen an die Software. Um dem Nutzer auch im mobilen Umfeld die ihm vom Desktop bekannte Software bereitzustellen bedarf auch der Entwicklung mobiler Versionen der Anwendungen, die er schon auf dem Desktop nutzt. Mit dem *Pattern Mapping* wurde dabei von Javahery et al. [3] eine Möglichkeit vorgestellt, wie Teile des Redesigns dieser Anwendungen strukturiert werden können und gleichzeitig einen Mehrwert für weitere Anpassungen zu schaffen.

Gegenstand dieser Arbeit ist es, das Verfahren im Rahmen einer Fallstudie an einem realen Beispiel zu erproben. Im ersten Abschnitt wird dabei ein Überblick über die zu redesignende Software sowie dessen Merkmale gegeben. Im zweiten Abschnitt gibt eine detaillierte Untersuchung Aufschluss über den Aufbau des Interface von Fleetmon. Im dritten Teil wird die Zielumgebung betrachtet. Als Ergebnis davon werden unter anderem die Rahmenbedingungen auf Hardwareseite, an die sich das Redesign halten muss, festgelegt. Im vierten Abschnitt wird das *Pattern Mapping* für bestimmte Elemente im Detail betrachtet und diskutiert, aus welchen Gründen bestimmte Entscheidungen getroffen wurden. Im letzten Teil der Arbeit wird der Verlauf des *Pattern Mappings* ausgewertet.

2 Fleetmon

Um zunächst einen Überblick zu erhalten, welches Konzept hinter Fleetmon steht und welche Aufgaben durch die Nutzer erledigt werden, folgt sowohl eine kurze Beschreibung und Einordnung der bestehenden Anwendung selbst als auch der Zielgruppe. Ferner wird gezeigt, warum eine mobile Version von Fleetmon notwendig ist.

Fleetmon ist ein seit 2006 durch die Firma Cruise Systems GmbH entwickeltes Geoinformationssystem (GIS). Unter einem Geoinformationssystem versteht man dabei „ein rechnergestütztes System, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst und redigiert, gespeichert und reorganisiert, modelliert und analysiert sowie alphanumerisch und graphisch präsentiert werden.“ [4] Es wird überwiegend im maritimen Umfeld eingesetzt. In einem Client-Server-Szenario ermöglicht es die Darstellung der Positionen von Schiffen und anderen Objekten, wie beispielsweise Borinseln, Signalanlagen oder an Land aufgestellte AIS-Sendeeinrichtungen, zu denen eine Geoposition übermittelt wird. Neben Namen, Positions- und Kursdaten können weitere Metainformationen wie beispielsweise Herkunftsland, Schiffstyp, Ladung, Zielort und Ankunftszeit angezeigt werden. Diese Daten werden zentral auf einem Serversystem gesammelt, aufbereitet und an die Fleetmon-Clients verteilt. Fleetmon kann sowohl als alleinstehende Desktopanwendung auf einem Computer als auch eingebunden in eine Website eingesetzt werden. Dies wird durch die Entwicklung mit Adobe ActionScript und dem für dessen Ausführung notwendigen Flash Player möglich. Die Verwendung von Adobe Flash stellt gleichzeitig eine Plattformunabhängigkeit im Desktopbereich sicher, da für aktuelle Desktopbetriebssysteme und dort verwendete Browser eine Adobe Flash Player Version verfügbar ist.

Die Positionsdaten der dargestellten Schiffe und Objekte werden überwiegend aus dem frei zu empfangenen *Automatic Identification System* (AIS) erhoben. Dabei handelt es sich um ein System, welches automatisch und permanent die Position eines Schiffs über das UKW-Seefunksignal an seine Umwelt mitteilt. Andere Schiffe können dieses Signal empfangen und auswerten, um über die aktuelle Situation bezüglich des Schiffsverkehrs in ihrem Gebiet informiert zu sein. An dieser Stelle kommt das Empfängernetz von Cruise Systems zum Tragen. Ein weltweites Netz aus Antennen stellt den Empfang der AIS-Signale sicher und leitet die Daten über eine Internetverbindung an einen zentralen Server weiter. Dieser nimmt die Datenströme der einzelnen Empfangsantennen entgegen, speichert diese und leitet sie auf Anfrage an die Fleetmon-Clients weiter. Neben den Daten, welche via AIS empfangen werden, können auch direkte Datenquellen, wie etwa dedizierte Satellitenortungseinheiten, als Informationsquelle dienen.

Die Zielgruppe der Anwendung lässt sich in zwei Gruppen unterteilen. Die erste und größte Gruppe umfasst dabei Nutzer mit einem professionellen Nutzungshintergrund. Hierzu zählen jegliche Tätigkeitsfelder im maritimen Sektor wie beispielsweise Lotsen, Reeder und Schiffseigner oder administrative Organe wie Hafenverwaltungsbehörden. Schiffslotsen nutzen Fleetmon unter anderem um sich darüber zu informieren, welche

Schiffe sich in ihrem Arbeitsbereich aufhalten und wann diese in ihren Arbeitsbereich einfahren. Reeder könnten mit Fleetmon ihre gesamte Schiffsflotte weltweit und in Echtzeit verfolgen. Sie sind so über deren Kurs und Aufenthaltsort unabhängig von einem direkten Kontakt informiert.

Die zweite Zielgruppe sind Privatpersonen ohne direktes professionelles Nutzungsinteresse und mit maritimen Freizeitaktivitäten. Hierzu zählen beispielsweise neben Sportbootbesitzern auch Shipspotter. Als Shipspotter bezeichnet man Mitglieder einer Community, welche es sich zum Hobby gemacht haben, Fotos zu einzelnen Schiffen sowie vielfältige Informationen zu diesen zu sammeln und im Internet bereitzustellen.

Durch die immer bessere Verfügbarkeit von mobilen Datenverbindungen und der gleichzeitig fortschreitenden Entwicklung von Smartphone zu bezahlbaren Preisen liegt der Wunsch der zuvor genannten Zielgruppe nahe, Fleetmon auch unterwegs nutzen zu können. Dies ist mit der beschriebenen Anwendung jedoch aus mehreren Gründen so nicht möglich. Der wichtigste ist dabei, dass es sich von Grund auf um eine Desktopanwendung handelt, welche nicht für die Nutzung auf mobilen Geräten konzipiert ist. Der nicht für die mobile Nutzung ausgelegte Aufgabenfokus und das daraus resultierende Interface sind nur zwei von vielen weiteren Punkten, weswegen ein wie in der Literaturarbeit [1] beschriebenes Redesign der Anwendung notwendig ist.

Fleetmon besteht aus einer Hauptansicht, welche über die gesamte Zeit der Anwendungsnutzung nicht verlassen wird. In dieser Ansicht befinden sich neben frei beweglichen auch in der Position nicht weitere veränderbare Informationsbereiche.

Eine Übersicht über alle Informationsbereich findet sich in der nachfolgenden Tabelle. Die neben den Bereichsbezeichnungen aufgeführten Nummern referenzieren auf die besonderen Elemente in diesem Informationsbereich (vgl. Abbildung 1).

Bereichsbezeichnung	Elementnummern
Login	
Hauptansicht	1, 2, 3, 10, 11
Suche	6
Eigene Schiffsliste	
Steuerung	7, 12
Informationselemente	4, 8, 13, 14
Einstellungen	9

3.1.1 Login

Wird Fleetmon gestartet und wurden bei einem vorherigen Aufruf keine Zugangsdaten für den Login gespeichert, so befindet sich ein halbtransparentes Overlay auf dem gesamten Interface. Auf diesem ermöglicht ein Formular die Eingabe der Zugangsdaten. Nach einer erfolgreichen Authentifizierung verschwindet der Loginbereich mit dem Overlay und gibt Fleetmon frei für die eigentliche Nutzung durch den Anwender. Verfügt der Nutzer über noch keine Zugangsdaten, so kann er diese über die Website von digital-seas.com erhalten.

3.1.2 Hauptansicht

Nach dem erfolgreichen Login ist die gesamte Hauptansicht von Fleetmon zu sehen. Im Hintergrund werden sofort nach der erfolgreichen Anmeldung die aktuellen Umgebungsdaten für die momentane Position auf der Karte (10) geladen und dargestellt. Die Karte, auf der die Schiffe dargestellt werden, ist das tragende Element der Hauptansicht. Es besteht die Möglichkeiten zwischen unterschiedlichem Kartenmaterial zu wechseln und dieses in Ausschnitten zu vergrößern. Die Schiffsdaten

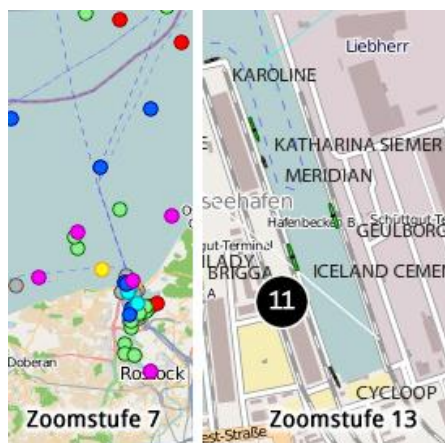


Abbildung 2 Gerendertes Overlay und Objektdarstellung der Schiffe

werden dabei in unterschiedlichen Zoomstufen unterschiedlich dargestellt. In niedrigen Zoomstufen, in der eine große Anzahl von Schiffe gleichzeitig auf der Karte sichtbar ist, werden die Schiffe auf einem vorgerenderten und statischen Layer angezeigt. Diese dient dazu, dem Nutzer eine Übersicht über die Verteilung der Schiffe in den unterschiedlichen Regionen zu geben. Wird in den Kartenausschnitt weiter hineingezoomt und sinkt damit gleichzeitig die Anzahl der parallel anzuzeigenden Schiffe, geschieht dies in einer anderen Darstellungsform. Fortan werden die Schiffe auf der Karte in Form von anklickbaren Symbolen (11) dargestellt, zu denen auch detaillierte Informationen abgerufen werden können. Visuell kann der Nutzer die Veränderung von reinen statisch gerenderten hin zu interaktiven und beweglichen

Symbolen auch dadurch wahrnehmen, dass diese unterschiedlich große schiffskörperähnliche Formen besitzen und eine Beschriftung. Die Darstellung der Schiffe ändert sich dabei je nach eigenem Zustand und Bewegungsrichtung. Das Schiff wird in einer niedrigen Zoomstufe als durch die Fahrtrichtung vorgegebener gerichteter Pfeil dargestellt. Hingegen werden auf Reede liegende Schiffe in den niedrigen Zoomstufen als Kreis abgebildet. In noch höheren Zoomstufen ändert sich die Darstellung in ein symbolisiertes Schiff. Dabei ist die Darstellung in Länge und Breite maßstabsgetreu und basiert auf den vom Schiff übermittelten Angaben über die Ausdehnung des selbigen. Hinzu kommt eine approximierte Richtungsanzeige. In dieser Zoomstufe werden die Positionsdaten für die Schiffe vom Server an Fleetmon in Echtzeit aktualisiert. Bei einer konstanten Übermittlung der Positionsdaten des Schiffs an den Server kann so auch die Bewegung auf dem Bildschirm in Echtzeit verfolgt werden. Bewegt der Nutzer den Mauszeiger über die einzelnen Schiffe auf der Karte, so werden einige Schlüsseldaten in einem dem Schiff visuell zugeordneten Fenster angezeigt (2). Wird das Schiff ausgewählt, so werden alle Detailinformationen in einem separaten Fenster dargestellt (3). In diesem Detailfenster ist ebenso die Funktion des Hinzufügens als auch des Entfernens aus der eigenen Schiffstabelle umgesetzt. Gleichzeitig mit der Anzeige der Objekte auf der Karte werden auch alle Objekte in einer am linken Bildschirmrand befindlichen Tabelle (1) gelistet. Dort erfolgt die Anzeige wie Namen, Herkunftsland und Typ. Diese Tabelle kann zusätzlich anhand aller Spalten sortiert werden. Für jedes Schiff in der Tabelle gibt es eine einfache Anzeige über den Status, ob es auf digital-seas.com Fotos zu diesem gibt und ob es in der eigenen Schiffstabelle gespeichert ist. Durch das Auswählen eines Eintrags in der Tabelle bewegt sich die Karte zu diesem und zentriert sie in der Mitte. Mit deinem doppelten Klick auf den Tabelleneintrag wird der Kartenausschnitt auf eine feste Stufe vergrößert.

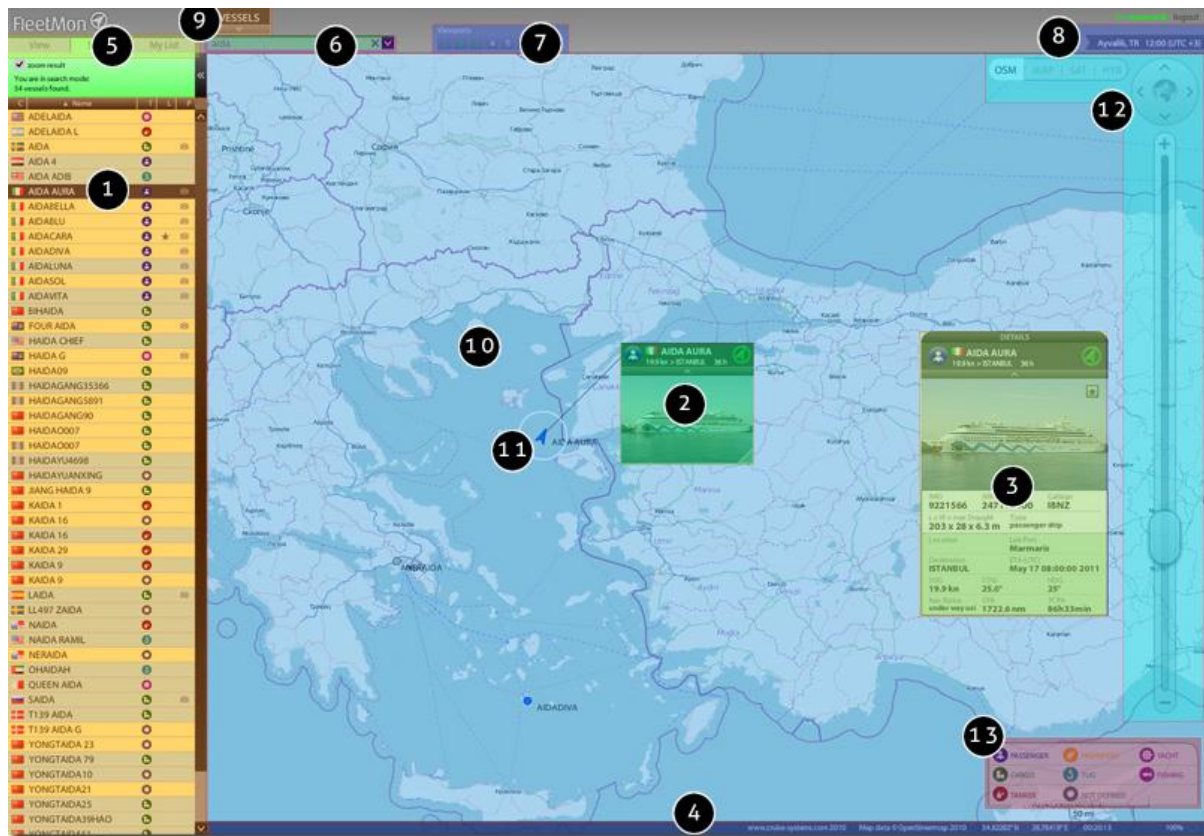


Abbildung 3 Ein gesuchtes Schiff ist ausgewählt und dessen Positionsverlauf ist geladen und dargestellt

3.1.3 Suche

Mit der Suchfunktion (6) ist es möglich, über den gesamten Datenbestand hinweg Schiffe zu suchen und die Ergebnisse sowohl auf der Karte als auch in der Tabelle anzeigen zu lassen (vgl. Abbildung 3). Gibt die Suche mehr Schiffe zurück als gleichzeitig sinnvoll angezeigt werden können wird der Nutzer aufgefordert seine Suche zu verfeinern. Durch die Notwendigkeit der relativ genauen Einschränkung der Suche und die daraus resultierende Anzahl an Ergebnissen erfolgt immer eine interaktive Darstellung der Schiffe auf der Karte. Ferner stehen in der Suchansicht die gleichen Funktionen (zum Beispiel das Sortieren der Tabelle) wie in allen anderen Ansichten zur Verfügung.

3.1.4 Eigene Schiffsliste

Der Menüpunkt „My List“ ermöglicht es dem Nutzer Schiffe, die er als seine Favoriten markiert hat, in einer eigenen Ansicht darzustellen. In dieser werden alle Objekte auf der Karte und in der am linken Rand befindlichen Tabelle (1) angezeigt, die der Nutzer vorher zu dieser hinzugefügt hat. Somit wird ermöglicht, eine kleine und vom Nutzer definierte Untermenge aller Schiffe einfach im Blick zu behalten.

3.1.5 Steuerung

Die Navigation auf der Kartenansicht erfolgt primär mit einem Zeigegerät, zum Beispiel einer Computermouse oder einem Touchpad. Mit diesen kann die Karte verschoben, der Kartenausschnitt vergrößert und ein Schiff ausgewählt werden. Im Zusammenspiel mit einem Teil der Steuerelemente auf der Karte kann neben der Änderung des Kartentyps auch die Navigation auf der Karte vorgenommen werden (12). Als Alternative zur

topografischen Karte von OpenStreetMap können Luftbildaufnahmen und Luftbildaufnahmen mit Wegenetzkarte von Bing.com sowie eine Seekarte ausgewählt werden. Ein Schieberegler (12) ermöglicht das Verändern der Zoomstufe, sollte die Computermouse nicht über ein Mausrad verfügen.

Um möglichst einfach und schnell von jedem beliebigen zu einem bestimmten Ort zu gelangen gibt es acht frei belegbare Speicherplätze (7). Mit einer Tastenkombination kann die aktuelle Ansicht inklusive der Zoomstufe gespeichert und mit einer weiteren Tastenkombination aufgerufen werden. Somit ist es möglich, oft besuchte Orte schnell und einfach anzusteuern.

3.1.6 Informationselemente

Zu den sekundären Elementen, die für die Nutzung der Anwendung wichtig sind, zählt die Legende, zu der die Icons der Schiffstypen (14), ein Entfernungsmaßstab (13), die Statusanzeige am unteren Bildschirmrand (4), der Gebietsname und die lokale Uhrzeit der auf der Karte angezeigten Region am oberen Bildschirmrand (8).

3.1.7 Einstellungen

Fleetmon verfügt über wenige direkte, durch den Nutzer vorzunehmende Einstellungen. Selten zu ändernde Punkte befinden sich in einem ausklappbaren Menü (9). Hierzu zählt beispielsweise die Möglichkeit, Fotos und Schiffsnamen nicht anzeigen zu lassen, um sich als Nutzer nicht von zu vielen gleichzeitigen Informationen ablenken zu lassen.

3.2 Interfacelemente

Nach einer ausführlichen Übersicht der sichtbaren Funktionen und deren Umsetzung in Form von Elementen im Interface von Fleetmon, der Erklärung der inhaltlichen Bedeutung sowie einer optischen Abtrennung (vgl. Abbildung 1 und Abbildung 3) folgt eine tabellarische Übersicht der im Interface benannten Elemente. Dabei werden die Elemente nacheinander gelistet und dahingehend betrachtet, welche HCI-Patterns sich in den einzelnen Elementen wiedererkennen lassen. Die Zuordnung bedarf dabei einer sorgfältigen Betrachtung der Pattern Library Kataloge^{1,2,3} in Abgleich mit dem vorliegenden Interface und dessen Elementen. Die in der Tabelle freigelassenen Felder spiegeln einerseits die Unvollständigkeit der Pattern Librarys und andererseits auch die Tatsache wieder, dass nicht jede gestalterische Umsetzung auf einem UI-Designpattern aufbaut. Ein automatisiertes Abgleichen ist ausgeschlossen und kann nicht erfolgen, weil die Beschreibung der Patterns in einer so uneinheitlichen und informellen Form vorliegt, dass eine sinnvolle automatische und computerbasierte Auswertung nicht möglich ist.

Ob die in der nachfolgenden Tabelle aufgelisteten Patterns bewusst bei der Entwicklung von Fleetmon verwendet worden bleibt bei der gesamten Betrachtung offen.

Element-nummer	Beschreibender Name	Pattern(s)
1	Schiffstabelle	Table Sorter ¹ Retractable Menu ¹ Color Coding ² Symbol Coding ² "Scrollbar"
2	MouseOver-Box	Carrousel ² Details on Demand ¹ Data-Tips ³
3	Detailinfobox	Carousel ² Drag and Drop ² Favorites ² Collapsible Panels ¹ Action-Button ¹ Color Coding ³ Symbol Coding ²
4	Statusleiste	Footer Bar ¹
5	Tabs	Module Tabs ² Views ¹
6	Suchfeld	Search Box ¹
7	Viewportnavigator	Shortcut Box ¹
8	Locationinfobox	
9	Einstellung	
10	Karte	Symbol Coding ³ Color Coding ³ Layering ³ Map Navigator ³ "On-Screen Navigation" "Maus-Steuerung"
11	Objekticons	Active Objects ³
12	OnScreen-Navigation	Simple Zoom ³
13	Typenlegende	Color Coding ³ Symbol Coding ³
14	Maßstab	Scala ³

3.3 Erläuterung der Patternzuordnung

3.3.1 Schiffstabelle



C	Name	T	L	P
	ADELAIDA			
	ADELAIDA L			
	AIDA			
	AIDA 4			
	AIDA ADIB			
	AIDA AURA			
	AIDABELLA			

Abbildung 4 Ausschnitt der Schiffstabelle

In der Schiffstabelle werden immer alle auf der Karte befindlichen interaktiven Objekte dargestellt. Alle Spalten der Tabelle sind sortierbar. Dies passt auf die Beschreibung des Patterns *Table Sorter*. Damit die Tabelle auf kleineren Bildschirmen nicht immer einen festen Platz der Kartenansicht belegt, kann diese, wie im *Retractable Menu* Pattern beschrieben, eingeklappt werden. In diesem Zusammenhang

kann die Tabelle als Menu betrachtet werden, bei dem die Zeilen als Navigationspunkte für die Karte dienen. Die einzelnen Zeilen sind je nach Status (in Fahrt grün, auf Reede gelb und unbekannt grau) farblich kodiert. Das beschriebene Verhalten und Aussehen findet sich in der Beschreibung des *Color Coding* Patterns wieder. Dies lässt auf dessen Anwendung des Pattern schließen. Das *Symbol Coding* Pattern kommt gleich an drei Stellen in der Schiffsliste zum Tragen. Sowohl der Schiffstyp als auch das Land und die Verfügbarkeit eines Fotos auf digital-seas.com werden in Form eines Icons symbolisiert. Die *Scrollbar* am rechten Rand der Tabelle ermöglicht die Navigation in der Tabelle, sollte diese nicht vollständig auf dem Bildschirm darstellbar sein.

3.3.2 MouseOver-Box

Bewegt der Nutzer den Mauszeiger über ein interaktives Objekte auf der Karte, so wird ein zweiteiliges Fenster angezeigt (*Details on Demand* Pattern, *Data-Tips* Pattern). Im oberen Teil befinden sich Hauptinformationen wie Name, Herkunftsland, Typ sowie ausgewählte Detailinformationen, welche unter anderem eine Einschätzung der Aktualität der Daten zulassen. Im unteren Teil kann, wenn vorhanden und vom Nutzer gewünscht, ein Foto vom Schiff angezeigt werden. Sind mehrere Fotos für das Schiff in der Datenbank hinterlegt, so kann der Nutzer sich mit Hilfe von zwei Steuerpfeilen durch die Bilder navigieren. Die Klickstrecke durch die Fotos ist in der Anzahl nicht begrenzt und beginnt wieder beim ersten Foto, wenn man beim letzten angelangt ist (*Carousel* Pattern). Zusätzlich ist es möglich, die Größe der Fotos frei zu skalieren. Die maximale Größe ist dabei nur durch den Bildschirm begrenzt.

3.3.3 Detailbox

Die Detailbox bietet Raum zum Anzeigen aller wichtigen Kenndaten des Schiffs. Neben den aus der MouseOver-Box bekannten Daten und deren Darstellungsforum, welche die Detailsbox wiederverwendet, werden im zweiten Teil der Box weitere Details zum ausgewählten Schiff dargestellt. Die freie Positionierung des Fensters innerhalb der Anwendung ist durch die Umsetzung des *Drag-and-Drop* Patterns möglich. Ebenso hat der Nutzer die Möglichkeit, das Fenster auf ein Minimum zu reduzieren (*Collapsible Panels* Pattern). Die zusätzlichen Informationen werden strukturiert im unteren Teil des Fensters angezeigt. So befindet sich alle wichtigen Identifikationsmerkmale wie IMO, MMSI und Callsign in einer Reihe und neben der Anzeige des Zielorts die erwartete



Abbildung 5 Detailbox mit Informationen zum aktuell ausgewählten Objekt

Ankunftszeit an diesem. Zusätzlich werden die regelmäßig aktualisierten Daten farblich Hinterlegt, um sowohl den aktuellen Zustand abzubilden als auch diese Daten in einen besonderen Fokus des Nutzers zu rücken (*Color Coding Pattern*). Zusätzlich zum Anzeigen der Informationen zu einem Schiff bietet Fleetmon die Möglichkeit, dieses zu einer eigenen Liste hinzuzufügen, in der man sie einfach im Überblick behalten kann. Mit dem Favoriten-Button, der aus *Action-Button* Pattern implementiert, ist es möglich, das Schiff zur eigenen Liste hinzuzufügen oder zu entfernen. Diese Umsetzung basiert dabei auf dem Pattern *Favorites*.

3.3.4 Tabs

Die im linken Bereich der Anwendung dargestellten Tabellen sind jeweils auf einen eigenen Tab ausgelagert und ermöglichen so die platzsparende Darstellung der selbigen (*Module Tabs Pattern*). Auch bietet es so die Möglichkeit zu einem späteren Zeitpunkt noch weitere Listen im gleichen Interface darzustellen. Gleichzeitig dienen die Tabs als Navigator zwischen den einzelnen Ansichten (*Views Pattern*), welche sich aus der unterschiedlichen Art der Datenbeschaffung der gelisteten Schiffe ergeben. Neben der Ansicht der aktuell im Kartenausschnitt befindlichen Schiffe gibt es weiterhin die Suchansicht sowie die der Schiffe in „My List“. Durch das Klicken auf die einzelnen Tabs werden für die jeweilige Ansicht die entsprechenden Daten geladen und in der Tabelle und der Karte dargestellt.

3.3.5 Suchfeld

Das Suchfeld (*Search Box Pattern*) ermöglicht das Durchsuchen der gesamten Datenbank anhand von Freitexteingaben. Beim Aktivieren des Eingabefelds wird automatisch in die Suchansicht geschaltet. Dort werden die gefunden Schiffe auf der Karte und in der Tabelle angezeigt. Wählt der Nutzer die Option der vollständigen Darstellung aller Suchergebnisse auf der Karte aus, so wird die Karte automatisch soweit herausgezoomt, bis alle Schiffe im Kartenausschnitt angezeigt werden können.

3.3.6 Viewportnavigator

Das Menü aus acht Buttons ermöglicht die schnelle Navigation zwischen vom Nutzer festgelegten Positionen auf der Karte. Der Nutzer kann mit Tastenkombinationen aktuelle Einstellungen der Karte wie Zoomstufe, Kartentyp und Position speichern. Mit dem Drücken der Taste 1 bis 8 können die zuvor gespeicherten Einstellungen abgerufen

werden. So wird dem Nutzer die Möglichkeit gegeben schnell zwischen weit auseinander befindlichen Orten zu navigieren.

Der Viewportnavigator ist somit eine Teilumsetzung des *Shortcut Box* Patterns. Dabei werden die Verknüpfungen erst durch den Nutzer selbst definiert und können jeder Zeit geändert werden.

3.3.7 Karte und Objekticons

Die Karte ist das Hauptelement der Anwendung. Auf ihr und um sie herum werden alle anderen Elemente angeordnet. Die Schiffsdarstellung erfolgt in drei unterschiedlichen Darstellungsebenen. Die erste Form ist die vorgerenderte Darstellung der Schiffe auf der Karte. Diese werden auf einem transparenten Layer über die bestehende Karte gelegt und ermöglichen so einen Überblick über die Verteilung der Schiffe (*Layering* Pattern). Zusätzlich werden alle Repräsentationen sowohl farblich (*Color Coding* Pattern) und als auch mit Hilfe von Symbolen (*Symbol Coding* Pattern) dargestellt. Dies ermöglicht eine schnelle Wahrnehmung und einfache Einordnung der Schiffe in vordefinierte Klassen. In der zweiten und dritten Darstellungsebene werden zusätzlich die unterschiedlichen Bewegungszustände des Schiffs dargestellt. Zudem sind die Objekte einzeln auswählbar und es kann eine Interaktion mit ihnen erfolgen (*Active Objects* Pattern).

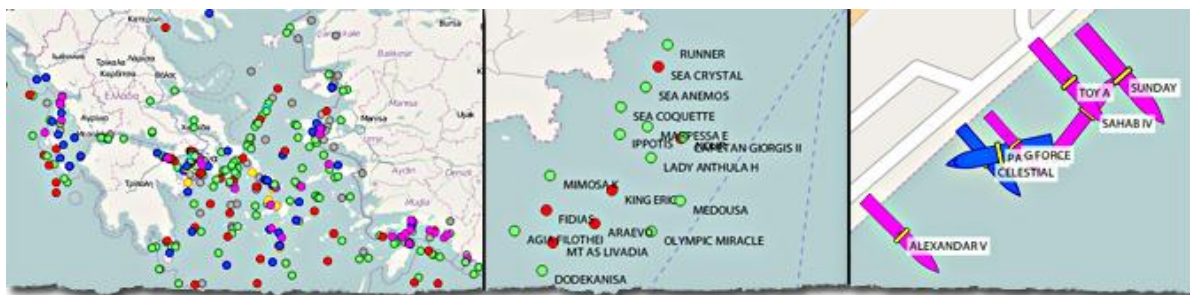


Abbildung 6 Unterschiedliche Darstellungsformen der Schiffe

3.3.8 On-Screen Navigation



Abbildung 7 OnScreen-Elemente zum Steuern der Karte

Die Elemente der On-Screen Navigation ermöglichen das Bewegen der Karte ohne über einen großen Aktionsradius zu verfügen. Zudem bietet sie exklusive Funktionen an. Die Sammlung von Steuerelementen auf der rechten Seite der Karte besteht aus einer Steuerung für die Bewegung über die Karte (*Map Navigator* Pattern), einen Schieberegler zum Zoomen (*Simple Zoom* Pattern) und mehrere Auswahlwähler zum Wechseln des Kartenmaterials anbietet. Zusätzlich gibt es noch die Möglichkeit mit einem Klick auf den Globus, der sich zwischen den Steuerelementen für die Richtung befindet, eine Gesamtansicht der Karte zu erhalten.

3.3.9 Typenlegende und Skala

In der Typenlegende der Schiffe sind Ergebnisse zweier Grundideen sichtbar. Einerseits das Verwenden von deutlichen Farben für das einfache Zuordnen von Schiffen zu den



Abbildung 8 Übersicht der Objekttypen, für die es sowohl ein Icon als auch eine eigene Farbe gibt

festgelegten Klassen und Typen (*Color Coding Pattern*), auch bei kleineren Darstellungsgrößen. Andererseits das zusätzliche Anbringen von Piktogrammen bei einer größeren Darstellung (*Symbol Coding Pattern*), um diese möglichst selbsterklärend zu gestalten. Das hat den

Vorteil, dass es bei aussagestarken Piktogrammen keiner weiteren textuellen Erklärung bedarf. Die Farben kommen generell bei jeder Darstellung von Schiffen zum Einsatz, die Piktogramme werden hingegen nur in der Tabelle, Objektdarstellung sowie den Detail- und MouseOver-Boxen angezeigt.

Die Maßstabsskala ist die einzige Möglichkeit zum ungefähren Abschätzen der Entfernung auf der Karte, da diese ansonsten über kein Gradnetz oder ähnliches verfügt. Sie passt sich automatisch der Zoomstufe an und gibt die Entfernung in Kilometern und Meilen an.

3.4 Interaktionsmodell und Dialoggraph

Um einen vom konkreten UI losgelösten Blick auf die in Fleetmon umgesetzten Aufgaben zu erhalten wird in dieser Arbeit der Weg der Abstraktion über die Erstellung eines *Concur Tast Trees* gegangen. Die Abbildung des Ist-Zustands basiert dabei einerseits auf dem Wissen des Autors über Fleetmon sowie auf der Interpolation und der Extraktion von Aufgaben aus den Elementen des UIs.

Das erstellte CTT-Modell resultiert aus der Analyse des Interfaces und der aktiven Nutzung der Anwendung. Der sich daraus ergebende Baum ist in Abbildung A1 zu sehen. Aus dem vorliegenden CTT ergeben fünf primäre Aufgabebereiche, welche in Fleetmon durch Funktionen und Interfaceelemente abgedeckt wurden. Zu diesen Aufgaben zählt das

- Navigieren auf der Karte,
- das Entdecken von Schiffen,
- das direkte Suchen von Schiffen,
- das Verwalten der eigenen Schiffsliste,
- das Anpassen der benutzerdefinierten Einstellungen innerhalb der Anwendung.

Bis auf das Verändern der Anwendungseinstellung sind alle Primäraufgaben noch in weitere Teilaufgaben untergliedert.

Für die mobile Version gilt es nun zu untersuchen, welche dieser Aufgaben für den Nutzer notwendig und somit umzusetzen sind. Dabei soll der Abschnitt 4.1 behilflich sein.

3.4.1 Ergebnisse aus der Betrachtung des Interaktionsmodells

Bei der Betrachtung des CTTs fallen dem Betrachter wiederkehrende Aufgabenmuster auf. Ein Beispiel hierfür ist der Teilbaum *Anzeigen*. Dieser tritt, wie in Abbildung 9 illustriert, in gleicher oder sehr ähnlicher Form in allen drei Hauptaufgabenzweigen auf. Gleiches gilt für andere Teilaufgaben, wie beispielsweise das Anzeigen der Daten oder die Navigation. Bei einer noch detaillierteren Modellierung der Aufgabe im CTT würden diese Muster vermutlich noch sichtbarer werden. Die Gründe für das wiederkehren gleicher Unteraufgaben in diesem konkreten Fall lassen sich unter anderem durch eine Beobachtungen erklären. Es handelt es bei den Hauptaufgaben um immer die gleichen Abläufe, welche sich überwiegend nur dadurch die Datenquelle oder deren Filterungsmechanismen unterscheiden.

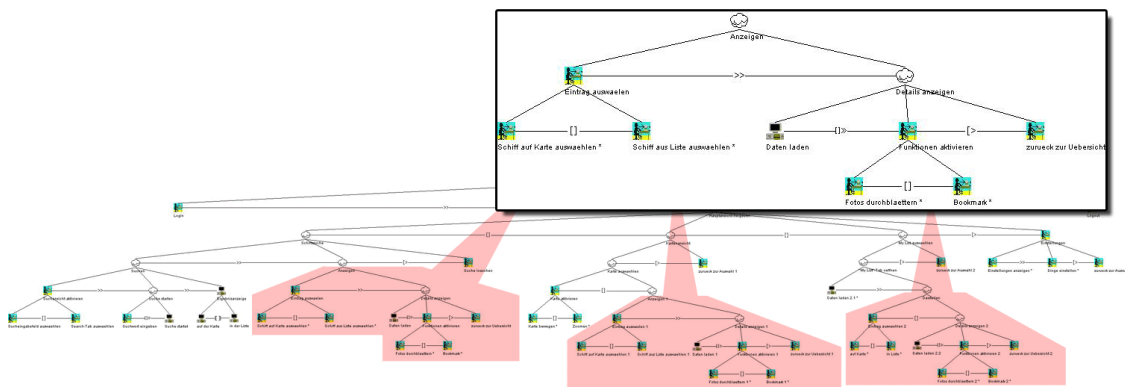


Abbildung 9 Teilbaum der Aufgabe "Anzeigen", welcher in dieser oder ähnlicher Form mehrfach im Aufgabenbaum vorkommt. Die rot hervorgehobenen Bereiche verdeutlichen, wie häufig diese Aufgaben ausgeführt werden.

Weitere und dann auch durch den Vergleich zur Mobilanwendung entstehende Ergebnisse folgen in Abschnitt 4.4.

3.4.2 Dialoggraph

Der auf Basis des CTT erstellte Dialoggraph (siehe Abbildung A2) entstand aus zweierlei Gründen. Zum Einem um unabhängig von der konkreten Anwendung nochmals überprüfen zu können, ob und in wieweit der erstellte CTT richtig erstellt wurde und so das Verhalten widerspiegelt, wie es in der Anwendung implementiert wurde. Zum Anderen, dem wichtigeren Aspekt, um eine Vergleichsmöglichkeit zwischen der bestehenden und der zu entwickelnden Anwendung zu haben. Dies wird erst durch die gleiche Modellierungsform möglich und kann dann auf einer gleichen Ebene geschehen.

4 Betrachten der Zielumgebung

Nach der Betrachtung des Ist-Zustands im 3. Kapitel wird in diesem der Soll-Zustand genauer untersucht. Im letzten Teil des Kapitels wird mittels CTT und Dialoggraph der Umfang und der Aufbau der zu erstellenden Fleetmonversion definiert, beschrieben und veranschaulicht.

Für die Überführung der Aufgaben von Fleetmon auf eine mobile Anwendungsumgebung bedarf es einer genauen Betrachtung der Zielumgebung. Unter diese Betrachtung fällt die Beschreibung der Zielgruppe und deren Anwendungs- und Aufgabenkontext für die redesiante Version von Fleetmon. Diese beiden Untersuchungen können überwiegend ohne Festlegung der Zielumgebung bezüglich der technischen Plattform erfolgen. Eine Ausnahme stellt hier nur die Tatsache dar, dass die Umsetzung auf einer für die mobile Verwendung geeigneten technischen Plattform erfolgen soll. Das an dieser Stelle weitere Vermeiden von Einschränkungen hat den Vorteil, dass die Ergebnisse bis zu diesem Schritt als Grundlage für ein Redesign in eine andere technische Umsetzungsrichtung verwendet werden können. Im darauffolgenden Abschnitt werden die technischen Rahmenbedingungen abgesteckt, anhand welcher die Ableitung einer konkreten Plattform für das Redesign erfolgt. Die Festlegungen in diesem Schritt sind notwendig, damit die Abhängigkeiten, wie beispielsweise die Verfügbarkeit von UI-Konzepten und -elementen, als konkrete Umsetzung von UI-Designpattern, beim *Pattern Mapping* der Ausgangspattern und -elemente auf die der neuen Plattform beachtet werden können.

4.1 Zielgruppe, Anwendungs- und Aufgabenkontext

Als Zielgruppe der mobilen Version von Fleetmon kann mindestens eine Teilmenge der in Kapitel 2 aufgezählten Personenkreise benannt werden. Hierzu zählen Entscheidungsträger und Führungspersonal in Reedereien, welche auch unterwegs über die aktuelle Position der eigenen Schiffe informiert sein wollen. Hinzu kommen Lotsen, denen die mobile Version von Fleetmon dazu dienlich sein soll, zu erfahren, welche Schiffe sich in ihrem küstennahen Arbeitsgebiet befinden oder wann mit dem Einlaufen zu rechnen ist. Dem professionellen Nutzerkreis stehen die Amateure gegen über, bei denen die Anwendung unter anderem im Bereich des Shipspottings zum Einsatz kommt. Sie können mit der mobilen Version von Fleetmon verfolgen, welche Schiffe als nächstes an der eigenen Position zu erwarten sind und wo sich andere Schiffe gerade befinden. Neben diesen sehr aktiven Nutzergruppen gibt es auch eine deutlich kleinere und eher passive Gruppe, welche die Anwendung selten und nur sporadisch zu sehr unterschiedlichen Aufgaben aus beiden zuvor beschriebenen Gebieten nutzt.

Bei allen hier aufgeführten Nutzergruppen, die nur als anschauliches Beispiel dienen sollen und keinem Anspruch auf Vollständigkeit erfüllen, wird die Anwendung primär im mobilen Kontext eingesetzt werden. Dies impliziert, dass für eine Übertragung von Echtzeitdaten mobile Datenfunkverbindungen genutzt werden müssen und zudem, zumindest nicht permanent, keine stationäre Energieversorgung zur Verfügung steht. Die Einschätzung über die Zielgruppe und deren Nutzungswünsche basieren dabei einerseits auf den Rechercheergebnissen für ein Projekt, welches 2009 im Rahmen der

Vorlesung „Softwaretechnik“ durchgeführt wurde [6], sowie auf Erfahrungen der bereits für das iPhone verfügbaren mobilen Version von Fleetmon [7]. Im Folgenden eine priorisierte Übersicht der Kriterien, die für die mobile Anwendung aus Nutzersicht wichtig sind:

Kriterium	Bemerkung
Mobilität	Die neue Anwendung soll als Gegenstück zur Desktopanwendung dienen und mobilen Zugriff auf die Daten von Fleetmon bieten
Strukturiertheit	Dem Nutzer sollen mögliche Entscheidung bei der Navigation durch die Anwendung abgenommen werden, so dass sich dieser auf die Informationsgewinnung konzentrieren kann und nicht den Fokus auf die Bedienung der Anwendung legen muss.
intuitive Bedienung	Nutzung von Patternumsetzungen, mit denen der Nutzer schon aus anderen Anwendungen des Telefon vertraut ist oder sein könnte

4.2 Technische Rahmenbedingungen

Damit die Anwendung, wie zuvor beschrieben, unterwegs genutzt werden kann, bedarf es eines Endgeräts, welches über die Möglichkeit einer mobilen Datenfunkverbindung und einer ausdauernden Energieversorgung verfügt. Zudem sollte eine einfache und sichere Bedienung ohne zusätzliche Eingabegeräte wie Stifte oder andere externe Zeiger auch in heterogenen Bewegungssituationen wie dem Gehen möglich sein. Eine ausschließliche Bedienung mit den Fingern ist in diesem Fall die umzusetzende Steuerung. Die in der Entwicklung erreichten Fortschritte der vergangenen Jahre haben dabei gezeigt, dass hierfür praxistaugliche Lösungen möglich sind und bereits existieren. Ein genügend großes, gut ablesbares und für die potenzielle Interaktion mit der Anwendung nutzbares Display ist ebenso eine grundlegende Voraussetzung. Da bei der mobilen Version von Fleetmon mit einer geringen Menge an Texteingaben zu rechnen ist, kann auf die Forderung nach einer eigenständigen Tastatur verzichtet und ein Softkeyboard als ausreichende Lösung angesehen werden. Ferner ist eine Möglichkeit zur eigenen Positionsbestimmung notwendig, um mit der Anwendung einfach im geografischen Kontext arbeiten zu können. Im Bereich der permanenten professionellen Nutzung wäre es ferner denkbar, eine eigenständige Hardware zu nutzen, welche den besonderen Ansprüchen im alltäglichen Dauereinsatz in Bezug auf Verarbeitung und Haltbarkeit zugutekommt. Hingegen ist es für die zweite, dem professionellen Bereich zuzuschreibende, Zielgruppe der Reeder sowohl aus Kosten- wie auch aus Praktikabilitätsgründen nicht realistisch, eine eigene Hardwarelösung zu verwenden. Sowohl für diese als auch für die Zielgruppe der Amateure und Anwender, die das Programm nur selten nutzen, ist eine kostengünstige und möglichst einfach nutzbare Lösung anzustreben. So sollte darauf geachtet werden, dass die Anwendung auf dem Gerät betrieben werden kann, über das der zukünftige Anwender bereits verfügt beziehungsweise für das er bereit ist, die erforderlichen Anschaffungs- und Betriebskosten zu zahlen.

Neben den Aspekten der Hardware nimmt die Software, und dabei im Besonderen das Betriebssystem als Basis für die Anwendung, bei der Planung des Redesigns eine sehr wichtige Rolle ein. Sowohl die Akzeptanz und die Verbreitung bei der Zielgruppe als auch die technischen Möglichkeiten aus Sicht des Entwicklers sind für diese Arbeit ein wichtiges Kriterium. Ein Ziel ist dabei, wie bereits ausführlich diskutiert [1], sich auf die Verwendung von vorhandenen UI-Konzepten und -elementen für das *Pattern Mapping* zu konzentrieren und möglichst darauf zu verzichten neue und plattformfremde Konzepte einzuführen.

Konkret wird das Redesign unter der Smartphoneplattform Android 2.3 durchgeführt. Diese Entscheidung wurde aus mehreren Gründen getroffen: Einerseits passen die zuvor beschriebenen Anforderungen auf die Merkmale der Plattform und der von ihr unterstützten Geräte. Andererseits sind es weiche Faktoren, die die Entscheidung untermauert haben. Hierzu zählt sowohl die Tatsache, dass das Software Developer Kit (SDK) kostenlos, frei zugängliches und gut dokumentiert ist. Zum anderen, dass Android eine sehr verbreitete Betriebsplattform für den Bereich der Smartphones [8] ist, weswegen die Anwendung nicht in einer in der Praxis noch wenig erprobten Umgebung entwickelt werden sollte. Gleichzeitig wird eine stetige Weiterentwicklung des Systems in Bezug auf die Unterstützung aktueller Techniken betrieben. Hierzu zählen beispielsweise Multitouchsteuerung, aktuelle Verbindungsstandards oder Ortungstechnologien.

4.3 Redesignentscheidungen

Anhand der in Abschnitt 4.1 beschriebenen Zielgruppe sowie ihres umrissenen Aufgabenprofils und den in Abschnitt 4.2 genannten technischen Kriterien können ein Teil der grundlegenden Entscheidungen für das Redesign getroffen werden. Hierzu zählt einerseits die primäre Abgrenzung nach unten respektive die Festlegung des Mindestmaßes an Aufgaben, welche, abgeleitet von den Anforderungen der Zielgruppen an die Anwendung, umgesetzt werden. Andererseits beschreiben die durch die technischen Rahmenbedingungen vorgegebenen Eigenschaften die Grenzen für die Umsetzung nach oben. Damit stecken sie den groben Rahmen ab, in dem sich das Redesign bewegen wird.

Alle grundlegenden Aufgaben wurden von der Desktopanwendung auf die mobile Anwendung übernommen. Dies ist möglich, weil es sich bei der Anwendung primär um eine Anwendung zum Betrachten von Daten handelt. Diese bedürfen auf dem mobilen Gerät zwar unter Umständen einer anderen Darstellung. Grundsätzlich ist deren Anzeige jedoch auch auf einem kleinen Bildschirm möglich; keine Grundlegenden Aufgaben mussten weggelassen werden. Die Veränderungen entstehen vor allem durch die andere Umsetzung der Funktionen. So kann die Schiffsliste (vormals Schiffstabelle) nicht gleichzeitig mit der Karte angezeigt werden. Hier ist ein Wechsel zwischen den Ansichten notwendig. Auf eine parallele Darstellung oder Verarbeitung musste bei allen anderen Elementen, wie beispielsweise der Detailansicht, ebenso verzichtet werden. Ferner sind neue Aufgaben hinzugekommen. Diese beschränkt sich in dieser Arbeit jedoch auf das offensichtlich Notwendige. Der Fokus der Arbeit, welcher auf dem grundlegenden

Redesign der bestehenden Anwendung für den mobilen Kontext liegt, soll nicht verschoben werden. Die neuen Aufgaben resultieren primär aus dem neuen Kontext der Nutzung und der Ausführungsumgebung der Anwendung. Hierzu zählt die Möglichkeit der eigenen und voll automatischen Positionsbestimmung, der Darstellung auf der Karte und dem damit verbundenen Laden von Umgebungsdaten. Durch den mobilen Kontext spielt die Bestimmung der eigenen Position bei der Nutzung der Anwendung eine besonders wichtige Rolle. Mit der allgegenwertigen Verfügbarkeit von Lokalisationstechniken ist diese Aufgabe ein offensichtliches Resultat aus dem neuen Kontext.

4.4 Interaktionsmodell und Dialoggraph

Das CTT-Modell des Ist-Zustands, der Desktopversion von Fleetmon, wurde für die Erstellung der mobilen Version nicht direkt verwendet. Vielmehr diente das Interaktionsmodell der Desktopanwendung als Basis für die Erstellung der mobilen Version von Fleetmon. Die Verwendung zweier voneinander getrennter CTTs entspricht dabei eigentlich nicht der Vorstellung aus der modellbasierten Entwicklung „one model – many interfaces“ (vgl. Paterno et al. [9]). Jedoch traten bei dem Versuch der direkten Verwendung durch Anpassung der Beziehungen zwischen den Teilaufgaben zu viele Unwägbarkeiten auf. Im Besonderen betraf das die temporalen Beziehungen der Aufgaben, welche sich zu denen der Desktopanwendung zu sehr unterschieden. So passieren in der Desktopversion gewisse Dinge parallel, welche in der mobilen Version in unterschiedlichen Ansichten dargestellt werden. Hierzu zählen beispielsweise das Anzeigen der Daten auf der Karte und in der Schifftabelle. Ferner mussten gewisse Aufgaben auch neu Erstellt oder weiter Aufgeteilt werden. Sowohl eingegrenzt als auch

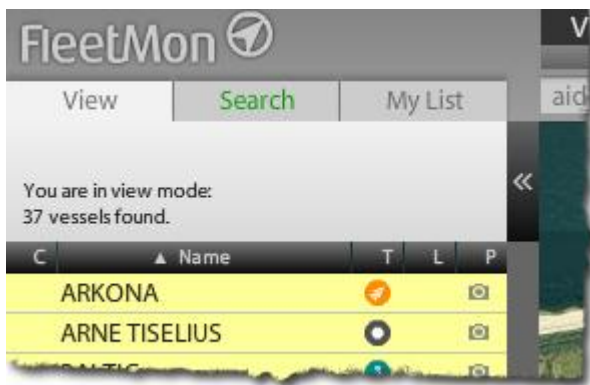


Abbildung 10 Filter auf einer Datenquelle

erweitert wurde das Aufgabenmodell durch die im Abschnitt 4.1 aufgeführten Details zur Zielgruppe und Ausführungsumgebung.

Verglichen mit dem ursprünglichen CTT der Desktopanwendung können mehrere Veränderungen festgestellt werden. Allgemein finden in der mobilen Version wenige Prozesse automatisch und durch das System ausgeführt statt, die mit dem Laden und Anzeigen von Daten in Verbindung stehen. Dies resultiert daraus,

dass diese Operationen auf der mobilen Plattform besonders aufwendig und mit hohem Ressourcenverbrauch in Bezug auf Energieverbrauch und Netzwerkverkehr verbunden sind. Diese Vorgänge wurden primär in den Aufgabenbereich des Nutzers verlagert und werden nur auf dessen Wunsch ausgeführt. Das bedeutet in der konkreten Anwendung, dass die aktuell verfügbaren Schiffsdaten zu einer Region nicht parallel oder leicht versetzt automatisch zur Navigation auf der Karte geladen werden. Vielmehr wird die Aufgabe „lade für diese Region aktuelle Daten“ auf ein UI-Element gelegt, welches den Ladevorgang auslöst. Diese Entscheidung basiert auf der Erkenntnis aus dem im Projekt

zur Softwaretechniklehrveranstaltung erstellten ersten Prototypen [6] von Fleetmon. Ein weiterer Punkt ist die Darstellung der Daten. Diese Schiffsinformationen werden nicht, wie in der Desktopversion, in beiden – der Tabelle und der Karte – Darstellungsformen gleichzeitig sondern immer nur in einer, die vom Nutzer ausgewählt wird. Diese Entscheidung ist der Tatsache des geringen Platzes auf dem Display und des anderen Aufmerksamkeitsfokus geschuldet. Die Ursache liegt, wie in der Vorarbeit [1] zusammenfassend beschrieben, vor allem auf der Interaktion mit dem Umfeld und der eigentlichen Tätigkeit, bei der die Anwendung in diesem Fall unterstützende Arbeit leisten soll.

An Aufgaben hinzugekommen sind die eigene Positionsbestimmung durch den Nutzer sowie das Auswählen der Ansicht, in welcher der Nutzer die Schiffe betrachten möchte. Genauso wie in der Desktop- gleichen sich auch in der Mobilanwendung bestimmte Abläufe und bilden ein gewisses Muster. Dies ist in denselben Tatsachen wie in der Desktopanwendung (beschrieben in Abschnitt 3.4.1) begründet. Aus Sicht der Gestaltung und der konkreten Programmierumsetzung kann die Überschneidung durchaus von Vorteile sein. So könnten die gemeinsamen und sich überlagernden Aufgaben vereint dargestellt und umgesetzt werden. Als Beispiel kann hier die Schiffstabelle in Fleetmon angeführt werden. Diese mit dem *Tab* Pattern umgesetzten Filter nutzen jeweils dieselbe Tabellendarstellung und laden jeweils nur andere Daten für die Darstellung. Der CTT findet sich in Abbildung A3.

Wie Eingangs zu diesem Abschnitt geschildert entstand für die mobile Anwendung ein eigener CTT. Auf Basis dieses CTTs wurde ein Dialoggraph A4 für die mobile Anwendung erstellt, welcher das Verhalten bei der Bedienung der Anwendung erfahrbar macht. Im Vergleich zum Dialoggraphen der Desktopanwendung, welcher auf Basis des für diese Anwendung erstellten CTTs entstanden ist, wird deutlich, dass sich das Verhalten der Desktopanwendung zur mobilen Anwendung klar unterscheidet. Deutlichster Unterschied ist dabei die Tatsache, dass alle Informationen immer nur in einer Ansicht angezeigt werden können und somit sinnvoll und zielgerichtet mit dem vorhandenen Platz umgegangen werden muss.

5 Pattern Mapping

Nach der vorbereitenden Analyse und den Grundüberlegungen für die Rahmenbedingungen der mobilen Fleetmonversion folgt nun die Überführung der Desktopanwendung in die Mobilanwendung. Dies geschieht nun auf der Ebene der UI-Designpatterns und deren *Pattern Mappings* von der Desktopplattform zur Mobilplattform.

Die folgende Tabelle zeigt dabei die ursprünglichen Elemente, deren in Abschnitt 3.2 zugeordneten Pattern sowie die Bezeichnung in der mobilen Fleetmonversion und der Patterns, mit dessen Hilfe ein Überführung auf das mobile Gerät möglich ist. Elemente, die es im mobilen Interface nicht mehr geben wird, sind mit einem Strich versehen: Deren Aufgabe wird durch ein anderes Element übernommen oder fällt in der Form weg.

urspr. Ele.nr	beschreibender Name	Pattern(s)	beschreibender Name	Pattern(s)
1	Schiffstabelle	Table Sorter ¹ Retractable Menu ¹ Color Coding ³ Symbol Coding ³ "Scrollbar"	Schiffsliste	List navigation ⁴ Static list ⁴ Color Coding ³ Symbol Coding ³ Change view ⁴ Fling to scroll ⁴ Retractable Menu ¹
2	MouseOver-Box	Carrousel ² Details on Demand ¹ Data-Tips ³	OnTap-Box	Symbol Coding ³
3	Detailinfobox	Carousel ² Drag and Drop ² Favorites ² Collapsible Panels ¹ Action-Button ¹ Color Coding ³ Symbol Coding ³	Detailinfos	Carousel ² Favorites ² Color Coding ³ Action Buttons ⁴ Symbol Coding ⁴ List navigation ⁴ Thumbnail ⁵
4	Statusleiste	Footer Bar ¹	Information Bar	Progress bar
5	Tabs	Module Tabs ² Views ¹	-	
6	Suchfeld	Search Box ¹	Suchfeld	Search bar ⁴
7	Viewport-navigator	Shortcut Box ¹	-	
8	Locationinfobox		-	
9	Einstellung		Einstellungen	
10	Karte	Symbol Coding ³ Color Coding ³ Layering ³ Map Navigator ³ "On-Screen Navigation"	Karte	Symbol Coding ³ Color Coding ³ Layering ³ Pinch & Spread ⁴

"Maus-Steuerung"				
11	Objekticons	Active Objects ³	Objekticons	Active Objects ³
12	OnScreen-Navigation	Panning ³ Simple Zoom ³	-	
13	Typenlegende	Color Coding ³ Symbol Coding ³	-	
14	Maßstab	Scala ³	-	
16			Login	Progress wheel dialog ⁴

5.1 Detailbetrachtung

Um die Veränderung der Anwendung und des Interfaces nachvollziehen zu können und den Einfluss des *Pattern Mappings* zu verstehen, folgt eine detaillierte Betrachtung der Umsetzung des *Pattern Mappings*.

Konkret werden sowohl die Schiffsliste wie auch die Karte und die Detailinfo beziehungsweise der Detailinfos-Screen diskutiert. Diese drei Hauptelemente finden sich sowohl in der ursprünglichen als auch in der neuen Fleetmonversion wieder. Zudem verfügen sie über die größte Anzahl an Pattern aller Elemente, welche für ein Redesign zur Verfügung stehen. Die Vielzahl von *Pattern Mappings* ermöglicht den Prozess des selbigen hinsichtlich der unterschiedlichen Abwägungsgründe zu verdeutlichen. Um das Ergebnis des *Pattern Mappings* visuelle zu verdeutlichen, dem Leser ein mögliches resultierendes Ergebnis zu zeigen und auch beim *Pattern Mapping* selbst bestimmte Möglichkeiten auszuloten sind Mockups⁶ als Ergebnis des fertigen *Pattern Mappings* entstanden. Anstelle von Scribbles wurden direkt Mockups erstellt, um diese für einen späteren Prototypen animieren zu können.

5.1.1 Schiffsliste

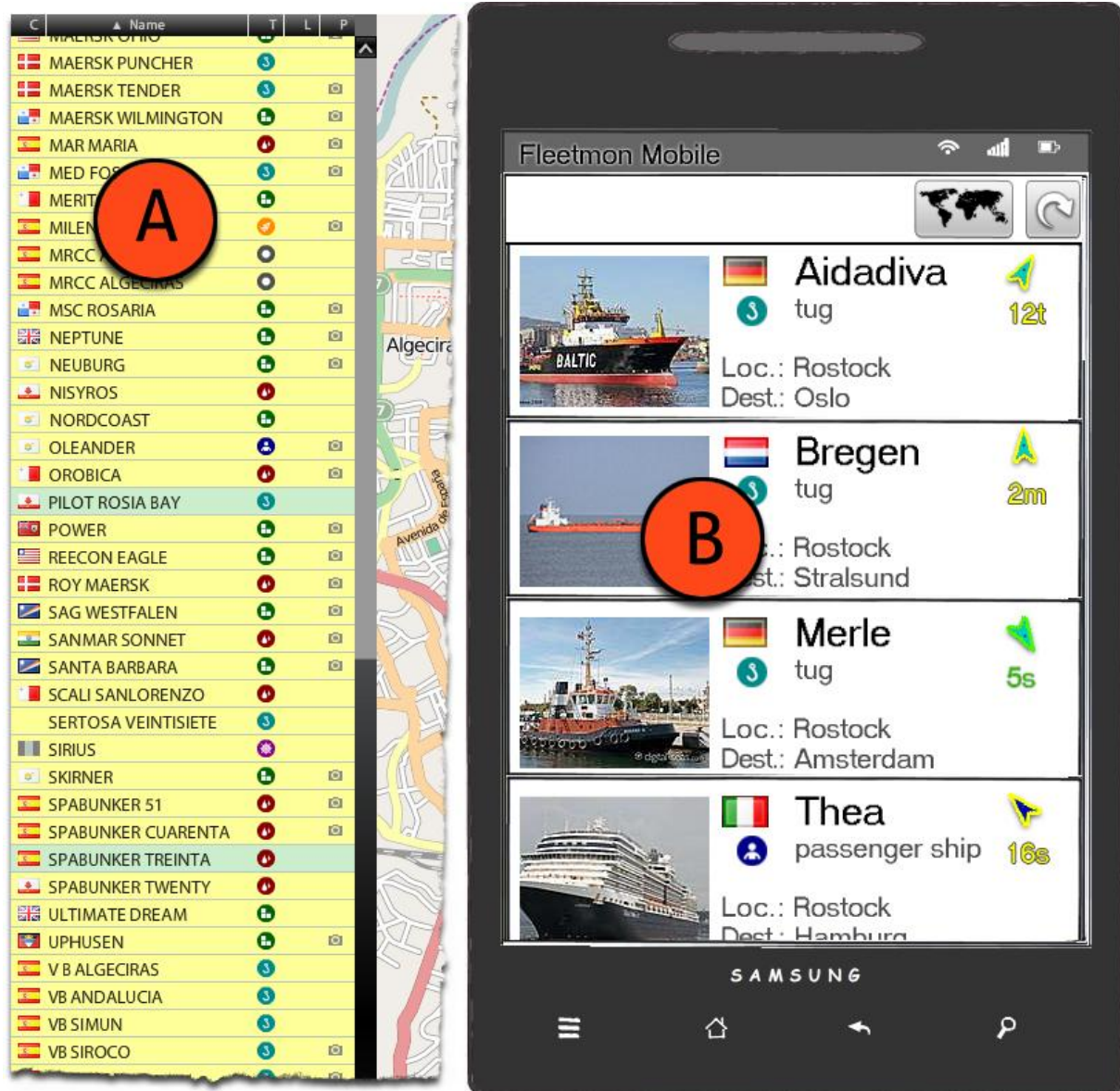


Abbildung 11 Vergleich der Schiffsliste in der Fleetmonversion für den Desktop und der Schiffsliste in der Mobil-Version von Fleetmon (Mockup)

An der Schiffsliste, dem ersten der drei Beispiele, wird ersichtlich, wie die konkrete Umsetzung von Designpattern variieren kann. A zeigt dabei die Schiffsliste auf dem Desktopsystem, B die gleichen Daten in einer Listung in Form eines Mockups für das Mobilsystem. Gehen wir davon aus, dass sowohl Umsetzung A als auch Umsetzung B die Form einer Listendarstellung – A ist dabei eine mehrdimensionale Liste und kann auch als Tabelle bezeichnet werden – haben, so erkennen wir sowohl Gemeinsamkeit als auch Unterschiede. Durch die physikalische Größe des Displays auf der einen und die präzisere Bedienmöglichkeit bei kleinen Interfaceelementen in A auf der anderen Seite sind die Einträge in A deutlich kleiner als in B dargestellt. Hinzu kommt die Tatsache, dass die Listung in A durch Spalten unterteilt ist, welche sich sortieren lassen. Auf diese Möglichkeit wurde in der einspaltigen Liste unter B verzichtet. Die hier vorgenommene alphabetische Sortierung anhand des Namens ist fest und kann nicht geändert werden. Die Entscheidung, dem Nutzer in B keine Sortierung zu ermöglichen und auf eine feste

alphabetische Reihenfolge anhand des Namens zu setzen hat zwei konkrete Vorteile. Zum einen wird, wie in 4.1 gefordert, eine Klarheit in der Steuerung der Anwendung erreicht und Komplexität aus dem Interface genommen. Im Falle einer Umsetzung von Sortiermöglichkeiten wäre eine Darstellung der Sortierreihenfolge im Interface notwendig, damit es zu keiner Irritation bei der Navigation mit und innerhalb der Liste kommt. Eine Möglichkeit, dies mit einer platzsparenden, zugleich intuitiven und von anderen Anwendungen bekannten Patternumsetzung zu realisieren, ergab sich nicht. Auf eine selbst entwickelte Lösung sollte an dieser Stelle verzichtet werden. Zum anderen kann so auf eine Suche innerhalb der Listung nach dem Namen verzichtet werden. Durch ein gezieltes Scrollen in B kann der Nutzer schnell zum gewünschten Eintrag beziehungsweise in dessen Region gelangen. Als Unterstützung hierfür wurde das Pattern *Fling to scroll* anstelle des in A verwendeten *Scrollbalkens* umgesetzt. *Fling to scroll* baut dabei auf dem *List navigator* Pattern auf und erweitert diese. Die Umsetzung gibt dem Nutzer auch die Möglichkeit, je nach Wunsch langsam oder schnell in der Listung und über viele Einträge zu navigieren, indem er mit dem Finger entweder schnell oder langsam über das Display wischt. Gleichzeitig spart es Platz, da ein mit den Fingern bedienbarer Scrollbalken eine bestimmte Größe aufweisen müsste, um sicher bedienbar zu sein. Alternativ beziehungsweise zusätzlich zum *Fling to scroll* Pattern könnte noch darüber nachgedacht werden, ob das Pattern *Scroll Thumb* umgesetzt wird. Dabei wird ein für das schnelle Navigieren über viele Einträge notwendiges Element nur dann eingeblendet, wenn tatsächlich eine Navigation innerhalb der Listet erfolgt. Ein solches Element kann zum Beispiel ein Button sein, der sich beim Scrollen der Lister in die jeweilige Richtung bewegt. Zusätzlich könnte noch eingeblendet werden, an welcher Position



Abbildung 12 Symbolische Darstellung des *Fling to scroll* Patterns

respektive Buchstaben in der Liste man sich befindet um eine genauere Navigation zu ermöglichen. Ein weiteres Pattern, dass in der Liste umgesetzt wird ist das *Static list* Pattern. Dessen Umsetzung verzichtet auf ein dynamisches Nachladen der Listenelemente. Stattdessen lädt es alle Daten beim Aufruf der Liste und stellt diese dann da. Dies ist unter anderem auch eine Voraussetzung für eine sinnvolle Umsetzung des *Fling to scroll* Pattern. Bei einer sich dynamisch erweiternden Liste würde das beschleunigte Scrollen immer dann stoppen, wenn der Nutzer am Ende der Liste angelangt ist.

Neben dem direkten Austausch eines Patterns durch ein anderes kann auch die



Abbildung 13
Umsetzung des
"Color Coding"-
und des „Symbol
Coding“-Patterns

Kombination mehrerer Pattern zu einer gemeinsamen Umsetzung eine funktionierende Lösung hervorbringen. So wird zum Beispiel der Status eines Schiffes mit der Umsetzung des *Color-Coding* Patterns in A realisiert. In B passiert dies auf dem gleichen Weg. Hinzu kommen, dass gleichzeitig der Typ des Schiffes farblich in den Pfeil codiert wird und so auf der Karte die Zugehörigkeit eines Schiffs zu einer Kategorie leicht erkennbar ist. In derselben Darstellung wird die Himmelsrichtung, in die der Rumpf zeigt, dargestellt. Hierfür wird das *Symbol-Coding* Pattern verwendet. Als Ergebnis zeigt ein zweifarbiges Pfeil (je nach Bewegungsstatus des Schiffs gelb, grün oder grau umrandet) die Himmelsrichtung, in die der Schiffsrumpf zeigt, in acht Unterteilungen an. Alle Informationen spielen zusammen mit der Aktualität des Signals im mobilen Einsatz eine gewichtige Rolle für die Bewertung der Datenrelevanz in Bezug auf den aktuellen Standort.

Die Darstellung eines Fotos in der Listenansicht anstelle der symbolischen Information hat sowohl Vor- als auch Nachteile. Einerseits ermöglicht sie, falls ein Foto vorhanden ist, möglicherweise eine einfache Identifikation des Schiffes beziehungsweise die Bestätigung der Vermutung, dass es sich um das richtige Schiff handelt. Andererseits bedeutet das Laden von Fotos hohe Datenkosten, weswegen die symbolische Darstellung wie in A an dieser Stelle genügen könnte. Letztendlich fällt die Entscheidung jedoch auf die Umsetzung des *Thumbnail* Pattern und damit die Einbindung eines kleinen Fotos in die Listenansicht, da dies zu einer besseren Nutzbarkeit führt. Durch die gleichzeitige Fortentwicklung im Bereich der drahtlosen Funknetze und deren Zuwachs an Leistungsfähigkeit ist zudem davon auszugehen, dass es sich bei dem negativen Argument um ein zeitlich begrenztes Problem handelt.

In der Desktopanwendung wird eine parallele Darstellung der Schiffsdaten sowohl in einer tabellarischen Form als auch in einer geografisch referenzierten mittels einer Karte umgesetzt. Da dies aus Platzmangel auf dem mobilen Gerät nicht möglich ist wird auf die Umsetzung des *Change View* Patterns zurückgegriffen. Dies ermöglicht, anders als bei der parallelen Darstellung in der Liste und auf der Karte den Wechsel zwischen den Darstellungsformen und somit auch zwischen dem Informationsfokus. Die Listendarstellung erlaubt die schnelle Suche des Schiffes, welches sich in dem Kartenabschnitt vermutet wird. Die Kartenansicht hingegen gestattet das Bestimmen der in der Umgebung befindlichen Schiffe sowie deren Verfolgung. Umgesetzt wird das *Change View* Pattern durch die Implementierung des *Action Button* Patterns. Dabei werden sowohl in der Karten- als auch in der Listenansicht ein Button zum Umschalten in die jeweils andere Ansicht implementiert.

Das Pattern *Retractable Menu* findet, anders als in A, keine explizite Umsetzung in der Schiffsliste. Vielmehr wird es allgemein in der Anwendung verwendet. Es ermöglicht das Einblenden des Menüs für die grundlegende Navigation zwischen den Hauptansichten der mobilen Version von Fleetmon.

Zum Abschluss dieses Abschnitts noch einmal eine Übersicht über die Patterns und ihr Mapping zwischen den beiden unterschiedlichen Anwendungen.

Pattern(s) in Desktopanwendung	Pattern(s) in Mobilanwendung
Table Sorter	Static list
Retractable Menu	Retractable Menu
Color Coding	Color Coding
Symbol Coding	Symbol Coding
Scrollbar	List navigation Fling to scroll
Symbol Coding	Thumbnail

5.1.2 Detailinfo

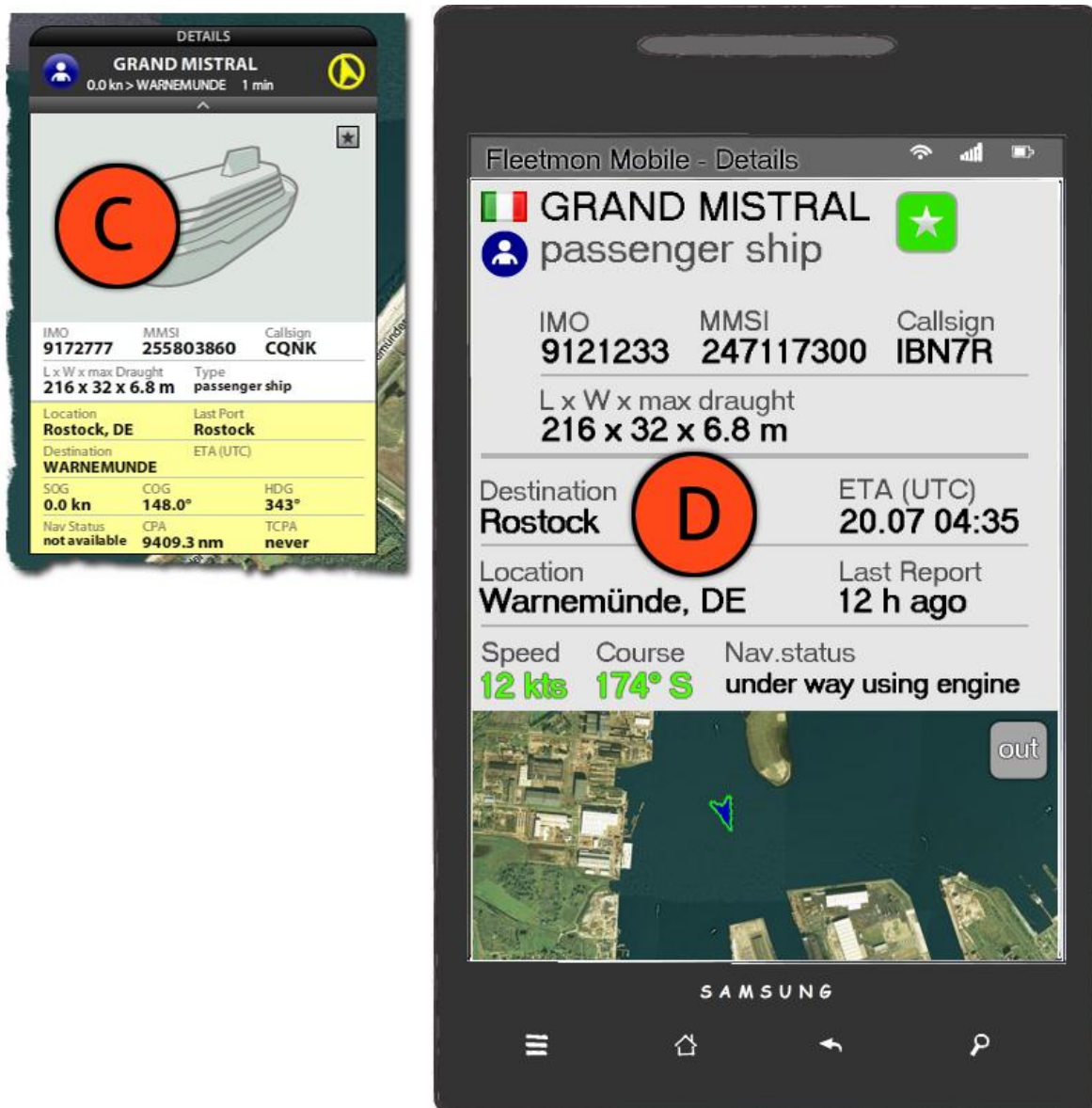


Abbildung 14 Die Detailsinfobox der Desktopanwendung im Vergleich zur Darstellung in den Mockups der mobilen Fleetmonversion

Die Ansicht „Detailinfos“ ermöglicht die Darstellung weitergehender Informationen zu einem Schiff sowie das Hinzufügen und Entfernen des selbigen aus der eigenen Favoritenliste. Dabei handelt es sich ebenso um eine Listendarstellung. Die „Detailinfos“-Ansicht unterscheidet sich jedoch dadurch, dass sie nur über sehr wenige und dafür große Listeneinträge verfügt. Gleichzeitig sind in den jeweiligen Listeneinträgen anderen Pattern umgesetzt um wieder andere Elemente zu realisieren. Für die Navigation in der Liste wurde das *List navigator* Pattern verwendet. Bei der inhaltlichen Gestaltung wurde sich an den Vorgaben der Desktopanwendung orientiert. Die Ausgabe weiterer Informationen wäre hier technisch möglich, unter der Betrachtung der Prämissen aus 4.1 jedoch nicht sinnvoll. Es sollen – sowohl für den Anwender mit professionellem Hintergrund als auch dem Gelegenheitsnutzer – alle wichtigen und aktuellen Informationen zu einem Schiff geliefert werden. Weitere Daten, etwas aus dem Schiffswiki, das auf Digital-Seas.com gepflegt wird, sind auf der Website abrufbar. Neben

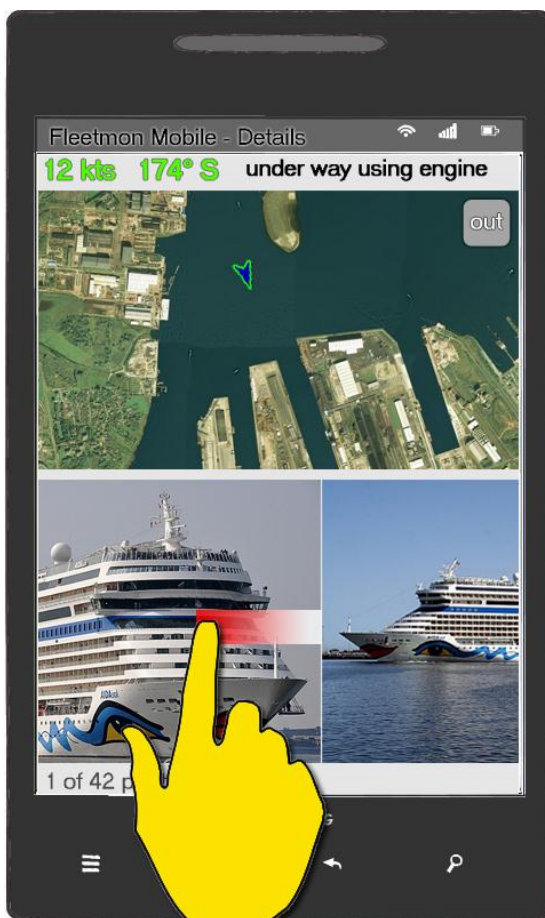


Abbildung 15 Umsetzung des Carousel-Patterns für das Betrachten der Fotos eines Schiffs

den statischen Daten zum Schiff, die wie in C auch in D im oberen Bereich des Fensters angezeigt werden, gibt es sich dynamisch ändernde Daten, welche im zweiten Drittel des Ausschnitts zu sehen sind. Dabei werden Daten wie Kurs und Geschwindigkeit durch die Umsetzung des *Color Coding* Patterns farblich hervorgehoben. Je nach Status des Schiffs – grün steht dabei für „in Fahrt“, gelb für „auf Rede“ und grau für „unbekannt“ – ändert sich die Farbe und gibt so schnell Auskunft über den momentanen Zustand des Schiffs. Wie schon in der Schiffsliste erfolgt auch in der Detailansicht der Daten die Umsetzung des *Symbol Coding* Patterns beim Typ als auch beim Herkunftsland des Schiffs. Dabei nimmt das Icon des Schiffstyps eine doppelte Rolle ein: es dient mit dem Piktogramm nicht nur zur Erkennung des Schiffstyps sondern ordnet diesem auch gleichzeitig eine bestimmte Farbe zu. Die farbliche Codierung (*Color Coding* Pattern) ist auch noch einmal im Abschnitt 3.3.9 genauer abgebildet. Der Button, mit dem ein Schiff zur Favoritenliste hinzugefügt und

wieder entfernt werden kann, ist das Ergebnis der Umsetzung dreier Pattern. Neben dem *Action Button* Pattern, das eine Hauptfunktionalität in der aktuellen Ansicht auf einen hervorgehobenen Button legt, wird ebenso das *Color Coding* Pattern und das *Favorites* Pattern umgesetzt. Dabei wird mit der Umsetzung des *Color Coding* Pattern signalisiert, ob ein sich ein Schiff bereits in der eigenen Schiffsliste (grün) befindet oder nicht (grau).

Wie auch schon in C wird das *Carrousel* Pattern auch in D umgesetzt und für die Anzeige

der Fotos zu einem Schiff verwendet. Dabei wird auf Buttons für die Navigation verzichtet und es kommen Wischgesten zum Einsatz. Trotz der großen Verbreitung dieser Interaktionstechnik unter der in Abschnitt 4.2 beschriebenen Umgebung muss beachtet werden, dass mit der verwenden dieser Technik eine klare Einschränkung in Bezug auf verwendbare Hardware gelegt wird. Für die Karte, auf der das Schiff noch isoliert und ohne weitere Schiffe angezeigt wird, kommt ebenfalls ein besonderes Pattern zum Einsatz, welches sowohl die On-Screen-Navigation als auch die Steuerung mit Maus und Tastatur aus C ersetzt. Dabei handelt es sich nicht, wie bei einer Karte auf einem Telefon mit Touchscreen zu erwarten wäre um das *Pinch & Spread* Pattern. Vielmehr wird das „Action Button“-Pattern umgesetzt, welches den Zoom zwischen zwei Stufen ermöglicht. Die Umsetzung des Action Button Pattern wird dabei dem Pinch & Spread Pattern vorgezogen, da dies die Navigation auf der Karte der Detailseite vereinfacht. Der Button ermöglicht das Umschalten zwischen zwei Zoomstufen mit einer Hand. Einmal auf die Zoomstufe, auf der Staatengrenzen sichtbar werden und somit eingeschätzt werden kann, an welchem Ort sich das Schiff im globalen geografischen Kontext befindet. Beim zweiten Mal tapen gelangt man wieder zurück auf ein so hohes Zoomlevel, so dass erkennbar wird wo sich das Schiff Vorort befindet – gegebenenfalls an welchem Liegeplatz. Ein weiteres navigieren über die Karte soll in dieser Ansicht nicht möglich sein, da es um die reine Darstellung der Daten sowie die Anreicherung dieser mit weiteren Kontextinformationen geht. Diese Funktion des schnellen Hin- und Herzoomens war so explizit in C nicht notwendig, da es mit dem Mausrad eine sehr gut zu bedienende Steuerung für den Zoom gibt. Befindet sich der Mauszeiger auf der Karte, kann überall einfach und schnell der Kartenausschnitt vergrößert und verkleinert werden.

Zum Abschluss dieses Abschnitts noch einmal eine Übersicht über die Patterns und ihr Mapping zwischen den beiden unterschiedlichen Anwendungen.

Pattern(s) in Desktopanwendung	Pattern(s) in Mobilanwedung
Carousel	Carousel
Drag and Drop	-
Favorites	Favorites
Action-Button	Action-Button
Collapsible Panels	-
Color Coding	Color Coding
Symbol Coding	Symbol Coding
-	List navigation

5.1.3 Karte



Abbildung 16 Darstellung der Karte in der Desktop- und der Mobilversion von Fleetmon

Eine Konsequenz aus den Erfahrungen des im Projekt zur Softwaretechniklehreveranstaltung [6] entwickelten Prototypen von Fleetmon ist die Nutzung unterschiedlicher Repräsentationsstufen der Schiffsinformationen auf der Karte, abhängig von der Zoomstufe. Für die grundlegende Realisierung wird für diesen Schritt das *Layering* Pattern umgesetzt. Dabei wird in niedrigen Zoomstufen eine andere Repräsentation der Schiffsdaten gewählt als in hohen und diese über das Kartenmaterial geblendet. Ist das aktuelle Ziel des Nutzers, das Erlangen einer Übersicht über den Schiffsverkehr in einer Region beziehungsweise die Navigation auf der Karte, so ist es nicht notwendig alle für den entsprechenden Bereich vorhandenen Schiffe als interaktives Objekt darzustellen. Neben der fehlenden Möglichkeit, eine sinnvolle Auswahl in der Masse an gleichzeitig angezeigten Schiffen auf so dichtem Raum zu treffen, ist auch für die interaktive Darstellung nicht genügend Systemleistung auf

mobilen Geräten verfügbar. Eine andere als die objektbasierte Repräsentation dient in diesem Fall auch als visuelle Abstraktion und unterstützt dabei gleichzeitig das Mobilitäts- und das Strukturierungskriterium (vgl. Abschnitt 4.1). Objekte, mit denen der Nutzer interagieren kann, werden erst angezeigt, wenn eine gewisse Zoomstufe erreicht ist. Mit der Auswahl eines Schiffs gelangt der Nutzer zu einer Detailansicht und damit zu weiteren Informationen über selbiges.

Neben dieser Betrachtung, die den primären Fokus auf die Interessen des Nutzers legt, sprechen auch Argumente aus Sicht des Entwicklers dafür, die Darstellung der Schiffsdaten im beschriebenen Szenario auf Layering zu setzen. Die Darstellung von Objekten auf einer Karte ist nicht nur in einer mobilen Entwicklungsumgebung eine Herausforderung. Geht es zudem noch um das Darstellen einer großen Anzahl von Objekten verschlimmert sich das Problem zusehends. Das Ergebnis ist eine gar nicht oder nur sehr schlecht bedienbare Karte, da viel Rechenzeit und Speicher für das Handling der Kartenobjekte benötigt wird, die bei der Berechnung der Gesamtdarstellung fehlt. Hingegen ist das Laden von vorgerenderten und teiltransparenten Kacheln, auf welchen die Schiffspositionen abgebildet werden, eine vergleichsweise einfache und sparsame Operation. Die entstehenden Kosten für den Datentransfer der Layerinformationen – diese werden nicht lokal sondern auf dem Server erstellt, welche auch die Schiffsinformationen liefert – können noch einmal weiter verringert werden, indem ein Caching der Daten umgesetzt wird. Die Realisierung des beschriebenen Verfahrens stellt dabei keine gesonderten Vorbereitungen auf Seiten der Datenbereitstellung da. Da die hier beschriebenen Erfahrungen auch in der Desktopversion von Fleetmon gemacht wurden steht bereits eine Engine zum Rendern der Kacheln bereit.

Sowohl auf den Layern als auch in den Objekten selbst wird das *Symbol Coding* Pattern umgesetzt, welches bereits im vorherigen Abschnitt Schiffsliste ausführlich beschrieben wurde. In einem Detail unterscheidet sich die Umsetzung in der Kartenansicht jedoch. Wird der vorgerenderte Layer eingeblendet, so wird in den Symbolen keine Richtungsanzeige kodiert. Die Schiffe werden lediglich durch einen kleinen farbigen Kreis dargestellt. Neben einer platzsparenden Darstellung ist in dieser Zoomstufe die Richtung, in die das Schiff zeigt, auch von geringer Relevanz.

Die Umsetzung des *Color Coding* Pattern wird in der Kartenansicht genauso fortgesetzt, wie es im zuvor erwähnten Abschnitt Schiffsliste vorgestellt wurde. Neben der Darstellung, für die grundsätzlich die gleichen Patterns wie für die Desktopanwendung genutzt wurden, unterscheidet sich die Navigation in der Karte deutlicher von der Desktopanwendung. So wird für das Zoomen in der Karte das *Pinch & Spread* Pattern umgesetzt (vgl. Abbildung 17). Diese ermöglicht die Steuerung ohne separate UI-Elemente und bietet so mehr Raum für die Darstellung der Schiffsdaten.



Abbildung 17 Verdeutlichung der Gestensteuerung für das Zoomen auf der Karte. Links das Hineinzoomen, rechts das Herauszoomen.

Trotz der Gestensteuerung konnte nicht auf UI-Elemente auf der Karte verzichtet werden. Bei den vier Elementen handelte es sich um (rechts nach links) einen Button zum Aktualisieren der auf der Karte dargestellten Schiffe, um einen Wechsler zwischen der Karten und der Listenansicht. Der dritte Button ermöglicht das Bestimmen und anzeigen der eigenen Position auf der Karte. Mit dem vierten Button kann zwischen einer Satelliten- und einer Kartendarstellung gewechselt werden. Alle vier Buttons sind Umsetzungen des *Action-Button* Pattern. Das manuelle Aktualisieren der auf der Karte dargestellten Daten ist der nicht zu unterschätzenden Datenmenge zu schulden, die beim regelmäßigen Laden anfallen würden. Gleichzeitig wird somit vermieden, dass schon bei einer kleinen Bewegung der Karte durch den Nutzer ein erneutes Laden und Anzeigen der Schiffe ausgelöst wird. Dies kann gerade bei rechenleistungsärmeren Geräten zu unerwünschtem Stocken in der Navigation führen kann. Bei der Wahl der Position der Buttons wurde drauf geachtet, dass die voraussichtlich häufiger genutzten Buttons mit der üblichen Einhandbedienung eines Smartphones durch einen Rechtshändler gut zu erreichen sind. Hingegen ist das Umschalten des Kartenmaterials und das Lokalisieren der eigenen Position eher eine seltenere Aktion und ist deswegen dementsprechend weniger direkt zu erreichen. Auf das nebeneinander anordnen der

Buttons wurde verzichtet, um somit die Gefahr des versehentlichen Drückens der jeweiligen Buttons zu vermeiden.

Zum Abschluss dieses Abschnitts noch einmal eine Übersicht über die Patterns und ihr Mapping zwischen den beiden unterschiedlichen Anwendungen.

Pattern(s) in Desktopanwendung	Pattern(s) in Mobilanwendung
Symbol Coding	Symbol Coding
Color Coding	Color Coding
Layering	Layering
“On-Screen Navigation”	Pinch & Spread
“Maus-Steuerung”	Action-Button
Map Navigator	

5.2 Auswertung

Im vorangegangenen Abschnitt haben wir das *Pattern Mapping* anhand dreier Elemente aus dem UI von Fleetmon betrachtet. Im nun folgenden Abschnitt wird das *Pattern Mapping* auf Basis der Erfahrungen aus dem Redesign der drei oberen Beispiele zusammenfassend betrachtet.

Am ersten der drei in dieser Arbeit redesignten Elemente lässt sich das prinzipielle Vorgehen beim *Pattern Mapping* gut erkennen. Die in der ursprünglichen Version umgesetzte Tabelle wurde in der mobilen Variante in eine Liste überführt. Die notwendigen Veränderungen ließen sich dabei einfach mit dem Mapping der Tabelle auf die Liste abbilden. Ein Grund hierfür ist die hohe Ähnlichkeit der Tabelle und der Liste. Das reduzieren der Tabelle auf die Umsetzung des *Static list* Pattern, welches die visuelle Komplexität reduziert, funktioniert gut. Die in den einzelnen Zeilen und Zellen abgebildeten Daten werden auch in der Liste ähnlich, gleich oder sogar genauer dargestellt (vgl. Abbildung 11). Die Patterns konnten weitestgehend überführt beziehungsweise übernommen werden.

Neben der direkten Überführung gibt es auch die Aufspaltung eines Pattern oder Elements in mehrere anderen Patterns beziehungsweise deren Umsetzung. Ein Beispiel hierfür ist die Scrollbar, welche auf das *List navigator* Pattern und das *Fling to scroll* Pattern überführt wurde. Das *List navigator* Pattern ersetzt hierbei die Funktion der Navigation in der Liste im Allgemeinen und das *Fling to scroll* Pattern die beschleunigte Navigation, wie sie die Scrollbar ermöglicht im Speziellen. Bei dieser Aufspaltung in zwei eigenständige Patterns wurde deutlich, dass nicht immer das eine und somit offensichtliche Mapping existiert. So hätte das *Fling to Scroll* Pattern auch durch andere Patterns ersetzt werden können. Gleichzeitig wäre es auch denkbar, weitere Patterns zusätzlich umzusetzen. Einen klaren und vorzuschreibenden Weg gibt es hierbei nicht. Vielmehr bedarf es der Erfahrung des Entwicklers, die richtige Entscheidung zu treffen, welches Mapping konkret vorgenommen werden kann und muss.

Gleiches gilt für das indirekte Mapping von Patterns. Dabei wurden Patterns oder Elemente nicht gemapt und übernommen beziehungsweise ganz neue eingeführt. Dies war bei den drei Beispielen immer dann erforderlich, wenn eine vollständig andere Darstellungsform gewählt wurde oder zusätzlich noch weitere Daten, als in der ursprünglichen Ansicht, dargestellt werden sollten. Hierfür ist die „Detailinfo“-Ansicht ein repräsentatives Beispiel. Wurden ursprünglich alle Daten in einem frei beweglichen Fenster angezeigt und war es zusätzlich möglich den geografischen Kontext über die unter dem Fenster befindliche Karte herzustellen, so muss dies in der redesignten Version alles in einer Ansicht umgesetzt werden. Aus der Perspektive des *Pattern Mappings* gehört dabei die Karte nicht zum Element des Detailfensters. Somit wurden in die „Detailinfo“-Ansicht neue Funktionalitäten umgesetzt, welche vorher nur indirekt zur Verfügung standen.

Zusammenfassend betrachtet bedarf es beim *Pattern Mapping* einer sehr gründlichen Vorbetrachtung. Hierzu zählt eine umfassende Analyse des Soll- und des Ist-Zustands sowie einer fundierten Erfahrung desjenigen, der das *Pattern Mapping* durchführt. Die notwendige Erfahrung lässt sich dabei nicht durch eine umfangreichere, detailliertere oder bessere Beschreibung des UI-Designpattern in der Pattern Library kompensieren. Ferner ist auch das Vorhaben des Erstellens eines formalen Gerüsts aus Pattern, welches dann auf unterschiedlichen Plattformen der gleichen Geräteklasse umgesetzt werden kann, ein unrealistisches. Dies wurde in dieser Fallstudie gerade im Zusammenhang mit Android deutlich. Dort existieren sehr viele unterschiedliche Geräte mit unterschiedlichen Spezifikationen bezüglich Displayformat und Auflösung. Gewisse Darstellungen ergeben dabei nur bis zu einer gewissen Größe Sinn und sollten darüber hinaus nicht mehr verwendet werden. Gleiches gilt für die Änderung es Anzeigeformt. Gegenstand des *Pattern Mappings* in dieser Arbeit war die ausschließliche Betrachtung des Hochformats. Bei einer weiterführenden Betrachtung müssen auch andere Formatformen, die durch die Lageänderung des Geräts im Raum entsteht, betrachtet werden. Zudem verwenden unterschiedliche Gerätehersteller auch unterschiedliche UI-Bibliotheken, weswegen die Umsetzung eines Patterns auf einer Plattform nicht immer gleich sein muss. Auch unterscheidet sich die Konnotation in der Darstellung und der Verwendung der Patterns auf den einzelnen Plattformen.

Die Rolle des Erstellens von Scribbles oder Mockups spielt bei der Entscheidung, welches Pattern auf welches gemapt wird, eine nicht zu unterschätzende Rolle. Häufig wird erst durch die visuelle Umsetzung erfahrbar, welches Mapping sinnvoll ist und welches nicht.

Unter Beachtung der hier aufgeführten Punkte kann das *Pattern Mapping* als Teil des Redesignprozesses einer Anwendung zur Strukturierung des selbigen beitragen und nachvollziehbarer gestalten.

6 Zusammenfassung

In der vorliegenden Fallstudie wurde die von Javahery et al. [3] vorgestellte Methode des Pattern Mapping als ein Hilfsmittel für ein Redesign der Desktopanwendung Fleetmon erprobt. Dabei war das Ziel ein UI für eine mobile Version von Fleetmon auf Basis des bestehenden der Desktopversion zu entwickeln.

Im ersten Schritt wurde der Ist-Stand analysiert. Hierzu gehörten die Untersuchung der Zielgruppe und die Überführung der bestehenden Anwendung in ein abstraktes Modell. Als Formalismus für die Darstellung der im UI umgesetzten Aufgaben wurde der *Concur Task Tree* verwendet. Anhand dessen konnte dann ein Dialoggraph erstellt werden, welcher Simulation der Softwarenutzung auf einer hohen Abstraktionsstufe ermöglichte. In diesem Schritt wurde auch das Interface untersucht und der Versuch unternommen, Elemente und Elementegruppen einem UI-Designpattern zuzuordnen. Dies funktionierte, in Abhängigkeit der Qualität der in den Pattern Librarys gesammelten Patternbeschreibungen, durchaus gut. Im nächsten Schritt wurde die Zielumgebung untersucht und der Soll-Zustand definiert. Ziel war es, im nächsten Schritt und auf Basis des aus der Desktopanwendung erstellten Aufgabenbaums eine Erweiterung anhand der Ergebnisse der Analyse der Zielumgebung vorzunehmen. Dies war jedoch nicht möglich, da zu große Änderungen in den temporalen Beziehungen der Aufgaben vorgenommen werden mussten. Ferner diente der Aufgabenbaum der Desktopanwendung als strukturelle Vorlage für den der Mobilanwendung. Nach der Fertigstellung des Aufgabenbaums wurde ein Dialoggraph für die Mobilanwendung erstellt, um für die weitere Umsetzung zu sehen, wie sich die Bedienung verhält. Im folgenden Schritt wurden dann das *Pattern Mapping* für die gesamte Anwendung vorgenommen. Für drei ausgewählte Ansichten entstand eine detailliertere Betrachtung des Vorgangs. Zudem entstanden Mockups, mit denen die reale Umsetzung erst visualisiert wurde. Im letzten Schritt erfolgte nochmals eine überblickende Zusammenfassung des *Pattern Mappings*.

Der in dieser Arbeit beschriebene und in Beispielen auch gegangene Weg des Redesigns ist ein grundsätzlich möglicher. Bei einem größeren Redesignvorhaben mit mehreren Zielumsetzungen (beispielsweise einer mobilen Anwendung für ein Telefon, einer Webanwendung sowie einer mobilen Web-Version) kann das hier beschriebene Vorgehen den Grad der Mehrarbeit senken und eine solide Ausgangsbasis für die Detailumsetzung schaffen. Ferner muss jedoch auch beachtet werden, dass der entstehende Aufwand für kleine und einmalige Projekt ohne fortlaufenden Charakter unverhältnismäßig hoch ist.

7 Literaturverzeichnis

1. SÜMNICK, M. **Patternbased Redesigns von Benutzungsoberflächen für mobile Anwendungen**. Rostock. 2011.
2. LABS, L. heise online - Smartphone-Markt wächst um 50 Prozent. **heise online**, 2011. Disponivel em: <<http://www.heise.de/newsticker/meldung/Smartphone-Markt-waechst-um-50-Prozent-1057411.html>>. Acesso em: 26 Juni 2011.
3. JAVAHERY, H. et al. Migrating User Interfaces Across Platforms Using HCI Patterns. In: SEFFAH, A.; JAVAHERY, H. **Multiple User Interfaces**. 1-. ed. West Sussex: John Wiley & Sons Ltd., 2004. Cap. 12, p. 241-259.
4. BILL, R.; NAUMANN, M. Geo-Informationssystem - Geoinformatik Lexikon. **Geoinformatik-Service**, 13 August 2008. Disponivel em: <<http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=793>>. Acesso em: 19 April 2011.
5. PATERNO, F.; MANCINI, C.; MENICONI, S. **ConcurTaskTrees**: A Diagrammatic Notation for Specifying Task Models. Proceedings of the IFIP TC13 Interantional Conference on HumanComputer Interaction. Uppsala: Chapman & Hall. 1997. p. 362-369.
6. SÜMNICK, M. et al. SWT-Projekt 0842, 2009. Disponivel em: <<http://emma.informatik.uni-rostock.de/Trac/swt0842>>. Acesso em: 23 Juni 2011.
7. JAKOTA DESIGN GROUP GMBH. Fleetmon :mobile, Schiffs-Tracking Vessel-Search Ship-Finder für iPhone, iPod touch und iPad im iTunes App Store. **iTunes App Store**, 26 Mai 2011. Disponivel em: <<http://itunes.apple.com/de/app/id326426737>>. Acesso em: 26 Mai 2011.
8. WIRTGEN, J. Smartphones: Android überholt Symbian, Apple verliert Marktanteile. **heise online**, 31 Januar 2011. Disponivel em: <<http://www.heise.de/newsticker/meldung/Smartphones-Android-ueberholt-Symbian-Apple-verliert-Marktanteile-1180547.html>>. Acesso em: 28 Mai 2011.
9. PATERNO, F.; SANTORO, C. **One Model, Many Interfaces**. Computer-Aided Design of User Interfaces III - Proceedings of the Fourth International Conference on Computer-AidedDesign of User Interfaces. Valenciennes: Kluwer. 2002. p. 143-154.

8 Abbildungsverzeichnis

ABBILDUNG 1 FLEETMON NACH DEM ERFOLGREICHEN LOGIN.....	4
ABBILDUNG 2 GERENDERTES OVERLAY UND OBJEKTDARSTELLUNG DER SCHIFFE.....	5
ABBILDUNG 3 EIN GESUCHTES SCHIFF IST AUSGEWÄHLT UND DESSEN POSITIONSVERLAUF IST GELADEN UND DARGESTELLT.....	7
ABBILDUNG 4 AUSSCHNITT DER SCHIFFSTABELLE.....	10
ABBILDUNG 5 DETAILBOX MIT INFORMATIONEN ZUM AKTUELL AUSGEWÄHLTEN OBJEKT.....	11
ABBILDUNG 6 UNTERSCHIEDLICHE DARSTELLUNGSFORMEN DER SCHIFFE.....	12
ABBILDUNG 7 ONSCREEN-ELEMENTE ZUM STEuern DER KARTE.....	12
ABBILDUNG 8 ÜBERSICHT DER OBJEKTYPEN, FÜR DIE ES SOWOHL EIN ICON ALS AUCH EINE EIGENE FARBE GIBT.....	13
ABBILDUNG 9 TEILBAUM DER AUFGABE "ANZEIGEN", WELCHER IN DIESER ODER ÄHNLICHER FORM MEHRFACH IM AUFGABENBAUM VORKOMMT. DIE ROT HERVORGEHOBENEN BEREICHE VERDEUTLICHEN, WIE HÄUFIG DIESE AUFGABEN AUSGEFÜHRT WERDEN.....	14
ABBILDUNG 10 FILTER AUF EINER DATENQUELLE.....	18
ABBILDUNG 11 VERGLEICH DER SCHIFFSTABELLE IN DER FLEETMONVERSION FÜR DEN DESKTOP UND DER SCHIFFSLISTE IN DER MOBIL-VERSION VON FLEETMON (MOCKUP).....	22
ABBILDUNG 12 SYMBOLISCHE DARSTELLUNG DES FLING TO SCROLL PATTERNS.....	23
ABBILDUNG 13 UMSETZUNG DES "COLOR CODING"- UND DES „SYMBOL CODING“-PATTERNS.....	24
ABBILDUNG 14 DIE DETAILSINFOBOX DER DESKTOPANWENDUNG IM VERGLEICH ZUR DARSTELLUNG IN DEN MOCKUPS DER MOBILEN FLEETMONVERSION.....	25
ABBILDUNG 15 UMSETZUNG DES CAROUSEL-PATTERNS FÜR DAS BETRACHTEN DER FOTOS EINES SCHIFFS.....	26
ABBILDUNG 16 DARSTELLUNG DER KARTE IN DER DESKTOP- UND DER MOBILVERSION VON FLEETMON.....	28
ABBILDUNG 17 VERDEUTLICHUNG DER GESTENSTEUERUNG FÜR DAS ZOOMEN AUF DER KARTE. LINKS DAS HINEINZOOMEN, RECHTS DAS HERAUSZOOMEN.....	30

9 Anhang

Der Anhang setzt sich zusammen aus:

A1 CTT des Ist-Zustands (der Desktopversion von Fleetmon)

A2 Dialoggraph des Ist-Zustands

A3 CTT des Soll-Zustands (der Mobilversion von Fleetmon)

A4 Dialoggraph des Soll-Zustands

Alle Dokumente sind in digitaler Form auch unter www.suemnick.com/docs/ba2011/ beziehbar.

¹ zu finden in der Interaction Design Pattern Library von van Welie <http://www.welie.com/patterns>

² zu finden in der Design Pattern Library von Yahoo! <http://developer.yahoo.com/ypatterns>

³ zu finden in dem patterbrowser - Interface Design Patterns - <http://www.patternbrowser.org>

⁴ zu finden bei Android Patterns – Shared by UNITiD and contributors - <http://www.androidpatterns.com>

⁵ zu finden bei UI-Patterns.com – User Interface Design Pattern Library - <http://ui-patterns.com>

⁶ Die Screenshots sowie die Mockups entstanden auf Basis von Grafiken aus der bestehenden Desktopversion von Fleetmon. Das Kartenmaterial wurde diesem Entnommen. Etwaige Rechte dritter bleiben unberührt.

Selbstständigkeitserklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

Marcus Sümnick
Rostock, den 4. Juli 2011