Hochschule Neubrandenburg
University of Applied Sciences

Deutsches Zentrum
DLR für Luft- und Raumfahrt e.V.

HOCHSCHULE NEUBRANDENBURG

Studiengang Geoinformatik

# Masterarbeit

## Entwicklung einer mobilen Lösung zur Visualisierung von relevanten Navigations-Informationen an Bord von Schiffen

Autor: Karsten Prehn
URN: urn:nbn:de:gbv:519-thesis 2014-0129-0

1. Betreuer: Prof. Dr. Andreas Wehrenpfennig
2. Betreuer: Dr. Ralf Ziebold

Datum: 30.10.2014

## Danksagung

Ich möchte mich bei allen bedanken, die mich während der Anfertigung dieser Arbeit unterstützt haben; insbesondere bei meiner Familie und bei meinen Mitarbeitern und Kollegen vom DLR.

**Eidesstattliche Erklärung**

Ich versichere, diese Arbeit selbstständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Neubrandenburg, _____

_____

Karsten Prehn

**Abstract**

Diese Arbeit beschreibt die Umsetzung einer Visualisierung von Navigationsdaten auf einem mobilen System anhand der Analyse von Anforderungen, einem darauf aufbauendem Entwurf und der Implementierung auf Basis der Android-Plattform. Die Anforderungsanalyse bezieht sich, aufbauend auf zuvor getroffenen Nutzungsanforderungen, auf vorhandene Normen und Standards, die für Navigationsgeräte an Bord von Schiffen verbindlich sind. Die Daten bezieht die Visualisierung von einem neuen, sich in Entwicklung befindenden, Navigationssystem, das hochgenaue Navigations- und Integritätsinformationen liefert. Die fertige Anwendung wird unter Labor- und realen Bedingungen an Bord eines Schiffes getestet und die dabei gewonnenen Erkenntnisse werden beschrieben.

# Inhalt

# ABBILDUNGSVERZEICHNIS

# TABELLENVERZEICHNIS

# ABKÜRZUNGEN

| | |
|---|---|
| AIS | Automatic Identification System |
| AJAX | Asynchronous JavaScript and XML |
| BRZ | Bruttoraumzahl |
| CCRP | Consistent Common Reference Point |
| CDGNSS | codebasiertes DGNSS |
| COG | Course over Ground |
| CPU | Central Processing Unit |
| DGNSS | Differential GNSS |
| DLR | Deutsches Zentrum für Luft- und Raumfahrt |
| ECDIS | Electronic Chart Display Information System |
| ECS | Electronic Chart Display |
| EKF | Extended Kalman Filter |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| GUI | Graphical User Interface |
| HPL | Horizontal Protection Level |
| HTTP | Hypertext Transfer Protocol |
| IALA | International Association of Lighthouse Authorities |
| IBS | Integrated Bridge System |
| IDL | Interface Definition Language |
| IEC | International Electronic Commission |
| IMO | International Maritime Organization |
| IMU | Inertial Measurement Unit |
| INS | Integrated Navigation System |
| IP | Internet Protocol oder International Protection |
| JSON | JavaScript Object Notation |
| LED | Light emitting diode |
| MGBAS | Maritime Ground Based Augmentation System |
| MSC | Maritime Safety Committee |
| MVC | Model View Controller |
| MVT | Maritime Verkehrstechnik |
| PDGNSS | phasenbasiertes DGNSS |
| PNT | Position, Navigation, and Timing |
| PVT | Position, Velocity, and Timing |
| RADAR | Radio Detection and Ranging |
| RAIM | Receiver Autonomous Integrity Monitoring |
| ROT | Rate of Turn |
| RTK | Real Time Kinematic |
| SOG | Speed over Ground |
| SOLAS | International Convention for the Savety of Life at Sea |
| SPP | Single Point Positioning |
| STW | Speed through Water |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| UI | User Interface |
| UML | Unified Modelling Language |
| URL | Uniform Resource Locator |
| UTC | Universal Time Coordinated |
| WGS | World Geodetic System |
| XML | Extensible Markup Language |

# 1 EINLEITUNG

Nach dem Untergang der Titanic am 14. April 1912 wurde als Reaktion das Übereinkommen zum Schutz des menschlichen Lebens auf See (SOLAS: International Convention for the Savety of Life at Sea) ins Leben gerufen mit dem Ziel, einen international gültigen Mindeststandard für die Sicherheit auf Schiffen zu schaffen. [1] Aktuell ist die fünfte Version aus dem Jahr 1974 (SOLAS-74) in Kraft. SOLAS regelt in 12 Kapiteln unter anderem Schiffskonstruktion, -stabilität und Elektroinstallationen, Radiokommunikation, Feuerschutz und Navigationsausrüstung. Im Folgenden sollen die Begriffe Navigation, Information und Navigationsinformationen unter dem Gesichtspunkt der Sicherheit auf See sowie der Begriff Mobilität erläutert werden.

Unter Navigation versteht man nach [2] die Technologie, zu Wasser, zu Land und in der Luft ein Gefährt sicher und zügig von einem Startpunkt an ein Ziel zu steuern. Für die Ermittlung des dafür besten Weges müssen die momentane Position und die Geschwindigkeit bestimmt werden. Für die Umsetzung müssen nach [2] folgende Fragen beantwortet werden: „wo befinde ich mich?“ (die Frage nach dem Standort), „mit welcher Geschwindigkeit bewege ich mich?“ (Geschwindigkeit), „in welche Richtung bewege ich mich?“ (Kurs), „wann bin ich am Ziel?“ (Zeit). Dies definiert die Navigationsinformationen Position, Kurs, Geschwindigkeit und Zeit.

Nach [3] waren zum Zweck der Navigation, zur Vermeidung von Gefahren und damit auch zum Schutz von Schiff und Besatzung Papierseekarten an Bord von Schiffen lange Zeit vorgeschrieben. [4] definiert die Karte[1] als „‚[…] ein maßgebundenes und strukturiertes Modell räumlicher Bezüge. Sie ist im weiteren Sinne ein digitales, graphikbezogenes Modell, im engeren Sinne ein graphisches (analoges) Modell.‘ (Hake 1988)“ Diese Begriffserklärung berücksichtigt bereits die Möglichkeit, Karteninhalte durch ein digitales Datenmodell zu speichern (beispielsweise in Datenbanken) und durch Informationssysteme zu repräsentieren. Der Begriff „graphisches Modell“ impliziert nach [4] den für alle kartographischen Darstellungen typischen „Zeichenvorrat […] mit vereinbarten Bedeutungen“. Der Zeichenvorrat in Seekarten umfasst nach [3] unter anderem die Darstellung von Küsten-, Tief- und Flachwassergebiete, Tonnen, dem eigentlichen Fahrwasser, die 10m-Tiefenlinie, detaillierte Angaben zu Seezeichen, Einzellotungen, Gezeitenstromvektoren, Radarlinien (RADAR: Radio Detection and Ranging), Meldepunkten, Hoheitsgrenzen, Sicherheitsgrenzen, Ruffrequenzen, etc..

Kritische oder gefährliche Aspekte, die Schiff und Besatzung gefährden können und gegen die die eben benannten Zeichen dem Nautiker helfen sollen, sind nach [3]: schwieriges Fahrwasser, hohes Verkehrsaufkommen, Dunkelheit, schlechtes Wetter, Sichtbehinderungen durch Regen, starker Wind oder hoher Seegang.

[1] Der Begriff Karte stammt vom lateinischen Wort *Charta* (Brief, Urkunde) und ist erst ab dem 15. Jahrhundert gebräuchlich. Davor war der Begriff *mappa* üblich. Die englische Sprache kennt mit den Begriffen *map* (für Landkarten) und *chart* (für Seekarten) bzw. *navigational chart* noch diese Unterteilung. [4]

1999 wurde in dem gleichnamigen Buch mit der „[…] elektronische[n] Seekarte“ [3] „ein vollkommen neues interaktives nautisches Informationssystem“ als ein Gesamtsystem beschrieben, das Papierseekarten als vorgeschriebenes Hilfsmittel für die Navigation an Bord von Schiffen ablösen und Navigationsmittel und Prozessabläufe in einem integrierten System mit den Schiffssensordaten vereinen soll. [3]

Die elektronische Seekarte garantiert nach [3] gegenüber der Papierseekarte mehr Sicherheit für das Schiff und die Besatzung und bietet mehr Wirtschaftlichkeit, da sie Prozesse, die zuvor auf Papierseekarten stattfanden vereinfacht und beschleunigt und Informationen kompakt darstellt. Auch fällt das manuelle Wechseln der Karte weg. Sie stellt unter anderem Informationen zu gesetzten Wegpunkten und der befahrenen Route bereit, bietet eine kontinuierliche und automatische Positionsaktualisierung, eine schnelle und sichere Feststellung von Bahnabweichungen, einen Nachtmodus oder ermöglicht das Setzen von Kontrollmarken und Zeitvorgaben zur Überwachung der eigenen Fahrt.

Eine Information ist nach [5] dann gegeben, wenn auf eine spezifische Frage eine Antwort gegeben werden kann. Der Träger der Informationen sind die Daten. Daten sind weniger strukturiert als Informationen. Die Formalisierung ist der Prozess, mit dem Daten aus Informationen erzeugt (bzw. digitalisiert) und beispielsweise in einer Datenbank gespeichert werden. Im Kontext eines Transfers werden aus diesen Daten dann wieder Informationen rekonstruiert. Der Transfer der Daten ist ein Kommunikationsprozess zwischen den beteiligten Komponenten. Beteiligt können Hardware- und Softwarekomponenten sein aber auch der Mensch bzw. die Nutzerin oder der Nutzer.

Ein Informationssystem ist ein computergestütztes Frage-Antwort-System, das aus „Daten, die zu einer Datenbank zusammengefasst sind, und aus einer Reihe von Werkzeugen zur anwendungsgerechten Verarbeitung dieser Daten“ [5] besteht. Der entscheidende Unterschied zwischen einer digitalen Karte und einem (Geo-)Informationssystem ist, dass letzteres die Möglichkeit zur interaktiven Kommunikation zwischen System und Nutzer bietet. Als Informationssystem bietet die Elektronische Seekarte Antworten auf Fragen, die für die Navigation auf See essentiell sind. Bei [3] werden beispielsweise folgende Fragen benannt:

- „Wo sind Küste und Schifffahrthindernisse?“
- „Wo ist der eigene Standort?“
- „Wo sind die anderen Schiffe?“
- „Welche Einflüsse sind von Natur/Umwelt zu erwarten?“
- „Gibt es auf der geplanten Route Gefahren?“

Entsprechend setzt sich die Elektronische Seekarte aus den typischen Komponenten eines Informationssystems zusammen. Diese beinhalten gemäß [3]:

- eine CPU (Central Processing Unit)
- einen Bildschirm (Ausgabe) für die Darstellung der digitalen Karte
- Eingabemöglichkeiten
- einen Datenspeicher (Seekartendaten sowie aktuelle Ereignisse (Wegpunkte, zurückgelegter Weg, …))

- Kommunikationsschnittstellen
- Schnittstellen zu externen (Navigations-) Geräten
- (Grafische) Nutzerschnittstelle ((G)UI)
- Software

An dieser Stelle sollte ersichtlich sein, welches die Navigationsinformationen sind und wie das Konzept der elektronischen Seekarte durch Visualisierung dieser Informationen den Nautiker unterstützen soll. Mobilität ist dabei aber nicht vorgesehen und der Begriff der Mobilität ist schwer zu definieren. Die *wikipedia* kennt mit der räumlichen, der sozialen, mit der Elektro- und der virtuellen Mobilität bereits vier unterschiedliche Mobilitätsformen und beschreibt die Mobilität allgemein als Befähigung einer Person oder eines Gegenstandes zum Wechseln des Ortes. Dabei ist der Mobilitätsbegriff - um ihn vom Begriff der Beweglichkeit zu unterscheiden - immer im Zusammenhang mit einer Handlung oder einem Motiv zu sehen. [6] Für diese Arbeit ist die Mobilität im Kontext mobiler Systeme von Interesse. Deshalb seien hier die Begriffe „reale (physische) Mobilität“ und „digitale Mobilität“, wie sie bei [7] definiert werden, übernommen. Ersteres ist die körperliche Bewegung des Nutzers und die Nutzung des technischen Systems während der Bewegung, letzteres die „Mobilität der Informationen“ im Sinne eines entfernten Zugriffs auf Informationen oder Anwendungen. Dem folgend besteht nach [7] das mobile System aus mindestens einer Anwendung und einem mobilen Endgerät, wobei das mobile Endgerät definiert ist als ein einzelnes Gerät, das:

- mit Prozessoren ausgestattet ist
- drahtlos und mit Batterien an jeden Ort transportiert werden kann
- während des Transports benutzt werden kann
- über integrierte Ein- und Ausgabeschnittstelen verfügt
- alle Komponenten in einem Gehäuse vereint

## 1.1 Aufgabestellung, Gliederung und Ziel der Arbeit

Das Ziel der Arbeit ist es, eine Anwendung zur Visualisierung von Navigationsinformationen auf einem mobilen System zu entwickeln. Die Anwendung soll nicht kommerziellen Zwecken dienen, sondern die prozessierten Daten eines Navigationssystems innerhalb eines Forschungsprojektes darstellen.

Abbildung 1 stellt den Ablauf dar, dem diese Arbeit folgt. Im Folgenden werden ab Kapitel 1.2 zuerst das eben erwähnte System, für das die Visualisierung entwickelt wird und die darzustellenden Daten beschrieben sowie Anwendungsfälle für die Nutzung der mobilen Visualisierung eingegrenzt. Danach werden in Kapitel 2 Nutzungsanforderungen für die Anwendungsfälle identifiziert und darauf aufbauend konkrete Anforderungen an eine mobile Visualisierung auf Basis der bestehenden Spezifikationen bzw. Standards an Navigationsanzeigen auf Schiffsbrücken ermittelt und hinsichtlich ihrer Vereinbarkeit mit dem mobilen Aspekt untersucht und entsprechend kommentiert, ergänzt, abgewandelt oder verworfen. Auf Basis der ermittelten konkreten Anforderungen wird in Kapitel 3 eine mobile

Visualisierung für die zuvor erwähnten Daten und Anwendungsfälle entworfen und anschließend in Kapitel 4 implementiert. Kapitel 5 dokumentiert die finale Anwendung und beschreibt, wie die Funktionsweise anhand zweier Testeinsätze im Labor und auf einer Messkampagne validiert wird und inwieweit die Anforderungen umgesetzt werden konnten. Modul- und Komponententests, die während der Implementierung durchgeführt werden, werden - sofern sie erfolgreich verlaufen - in dieser Arbeit nicht dokumentiert.

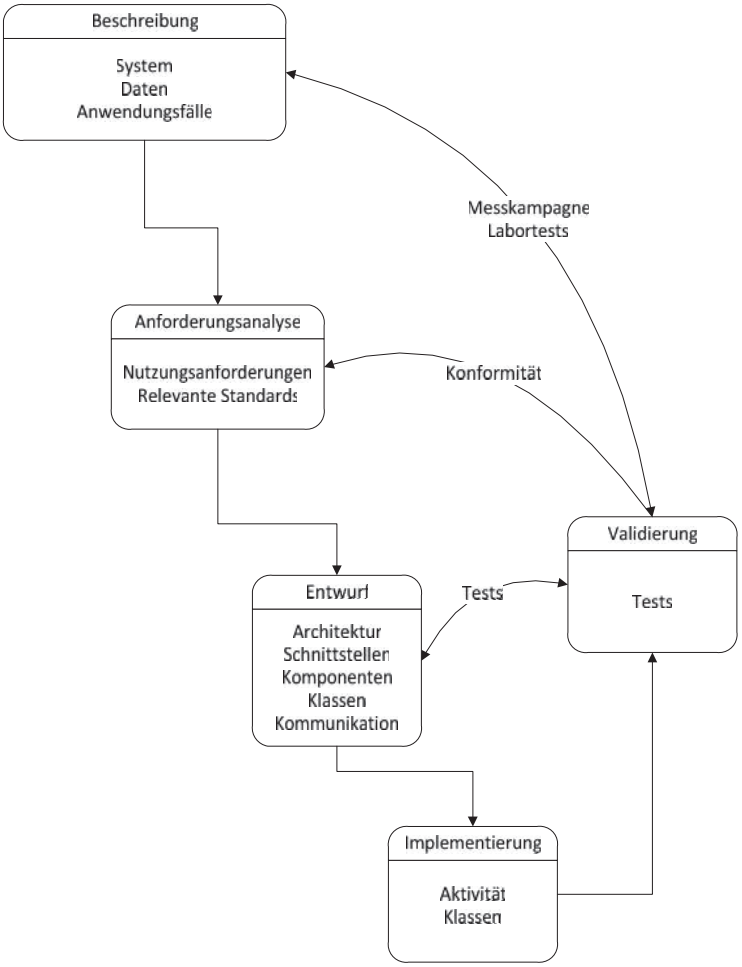


ABBILDUNG 1: DER ABLAUF, DEM DIE ARBEIT FOLGT

## 1.2 NAVIGATION AKTUELL

Heute werden als elektronische Kartensysteme (ECS: Electronic Chart Display) diejenigen elektronischen Seekarten bezeichnet, die die in der IMO Resolution A.817(19) aufgeführten Mindestanforderungen der Internationalen Seeschifffahrts-Organisation (IMO: International Maritime Organization) an ein Electronic Chart Display Information System (ECDIS) nicht erfüllen. [8]

Die IMO legt in Übereinkunft mit der SOLAS-Konvention Art und Umfang der auf Hochseeschiffen mitgeführten Ausrüstung fest. Die an Bord mitzuführenden Navigationsausrüstung und -systeme sind in Regel 19 der SOLAS-74 Kapitel V geregelt. So müssen Schiffe unabhängig ihrer Größe unter anderem einen Magnetregelkompass, Seekarten oder ein ECDIS sowie einen Empfänger für ein weltweites Satelliten- oder terrestrischem Funknavigationssystem zur selbstständigen Bestimmung und Aktualisierung der Schiffsposition mitführen. [SOLAS V, 2.1] Für Schiffe ab einer Bruttoraumzahl (BRZ) von 300 ist das Mitführen einer Echolotanlange, einer Radaranlage und einem Gerät zum Messen der Geschwindigkeit vorgeschrieben. [SOLAS V, 2.3] Zusätzlich gilt für Schiffe ab 300 BRZ in Auslandfahrt und Frachtschiffe ab 500 BRZ in Inlandfahrt die Ausrüstungspflicht eines Automatischen Schiffsidentifizierungssystems (AIS: Automatic Identification System). [SOLAS V, 2.4]

ECDIS integriert all diese Systeme. Es erlaubt auf einem einzigen Bildschirm nach [9] die Darstellung der folgenden Informationen und Sensordaten:

- eine Seekarte
- Teile der Seehandbücher, des Nautischen Funkdienstes und des Leuchtfeuerverzeichnisses
- das Radarbild
- Kurslinie und momentaner Standort des eigenen Schiffes.
- alphanumerische Navigationsdaten
- AIS-Symbole

Die Grenzen dieses Navigationssystems bestehen darin, dass die Qualität der Positions-, Geschwindigkeits- und Zeitinformationen (PVT: Position, Velocity, and Timing) aus dem GNSS-Empfänger (Global Navigation Satellite System) und Navigationsdaten aus den bordseitigen Geräten (Gyro-Kompass für den rechtweisenden Kurs, Speed Log für die Geschwindigkeit über Grund) entweder durch den Nautiker selbst bewertet werden müssen. Besitzt das Schiffe ein Integriertes Navigationssystem (INS: Integrated Navigation System), ist eine Bewertung der Daten anhand ihrer Plausibilität und Konsistenz möglich. Daten sind plausibel, wenn sie sich innerhalb eines erwarteten Wertebereichs befinden und konsistent, wenn sie beispielsweise widerspruchsfrei hinsichtlich redundanter Datenquellen sind. Bedingt durch unzureichende Redundanz der Sensoren und ohne die Verwendung von datenfusionsbasierten Methoden auf Rohdatenebene, lässt sich so allerdings nicht die Integrität aller PVT- und Navigationsdaten ermitteln. [1]

## 1.3 NEUES KONZEPT DER E-NAVIGATION-STRATEGIE

Der Schiffssicherheitsausschuss (MSC: Maritime Safety Committee) der IMO beschloss auf seiner 81. Sitzung, die Entwicklung einer „e-navigation“-Strategie voranzutreiben. Das Ziel der Strategie war es, ein Konzept zu entwickeln, mit dem sich bestehende und neue Navigationsgeräte und -instrumente in einem umfassenden System integrieren lassen. [10] „E-navigation“ ist definiert als das bord- und landseitige abgeglichene Sammeln, Integrieren,

Austauschen, Darstellen und Analysieren von maritimen Informationen unter Nutzung von elektronischen Geräten zur Verbesserung der Navigation von Kai- zu Kaikante und dafür notwendiger Dienste zur Gewährleistung der Sicherheit auf See und zum Schutz der Meeresumwelt. [11] Die „e-navigation"-Strategie wurde 2008 durch die IMO in der MSC 85/26 (Anhang 20) angenommen.

Im Rahmen der „e-navigation"-Strategie arbeitet das Projekt „Maritime Verkehrstechnik" (MVT) des Instituts für Kommunikation und Navigation des DLR (Deutsches Zentrum für Luft- und Raumfahrt) am Standort Neustrelitz an den Fragestellungen, „wie das maritime Verkehrssystem harmonisiert, technologisch ausgebaut und optimiert werden" kann, um Wirtschaftlichkeit und Sicherheit des Seeverkehrs zu steigern. Das maritime Verkehrssystem ist „die Summe aller strukturellen Komponenten, die erforderlich sind, um Personen und Güter auf dem Seeweg transportieren zu können." Dazu zählen „Häfen und Seewege mit ihren Leitsystemen, Schiffe […] sowie Einrichtungen, mit deren Hilfe die Transport- und Verkehrsprozesse sich planen, managen und umsetzen lassen". [1]

Aufbauend auf eigenen Studienergebnissen hat MVT drei Forschungsschwerpunkte entwickelt, deren Fokus auf dem „Ausbau des maritimen Verkehrssystems" und der „schrittweise[n] Einführung von Daten- und Systemintegrität in dedizierte Navigationsfunktionen" liegt. Integrität ist eine der „primäre[n] Anforderungen an sicherheitskritische Systeme" und sie ist definiert als „die Fähigkeit eines Systems, den Nutzer über die augenblickliche Verwendbarkeit des Systems (Systemintegrität) oder bereitgestellter Daten (Datenintegrität) zu informieren". Integritätsaussagen werden durch zusätzliche Funktionen im System gewonnen, die die Integrität des Systems oder der Daten überwachen. Die Aussage, ob mit Daten oder mit dem System gearbeitet werden kann, liefert die Integritätsüberwachung mittels eines Alarmpegels. [1]

Die drei MVT-Forschungsschwerpunkte befassen sich mit der Integritätsüberwachung von Globalen Navigationssatellitensystemen (GNSS) und Ergänzungsdiensten, mit der Überwachung und Bewertung der Verkehrslage anhand von Radar und AIS und mit einer „Multisensorbasierte[n] Unit zur bordseitigen Bestimmung von Positions-, Navigations- und Zeitdaten (PNT-Unit)" aus Messungen der Bordsensoren. [1] Die Daten, die die PNT-Unit erzeugt, sind Gegenstand der Visualisierung der Navigationsinformationen.

### *1.3.1 Beschreibung der PNT-Unit*

Die PNT-Unit ist eine (sich in Entwicklung befindende) Navigationseinheit für den maritimen Bereich, die die Rohdaten verschiedener schiffseitiger GNSS-Empfänger (code- und trägerphasenbasierte Entfernungsmessungen zu den Satelliten [1] ), terrestrischer Dienste und schiffseigener Sensoren verarbeitet, so dass die Position, Geschwindigkeit und Lage des Schiffes sowie die Integrität der Daten jederzeit hochgenau ermittelt werden können. [12] Dies ist in Abbildung 2 dargestellt. Die sicherheitskritische Bedeutung liegt in der Kenntnis der eigenen Position und Lage im Vergleich zu anderen Verkehrsteilnehmern und zum Verkehrsraum. [1]

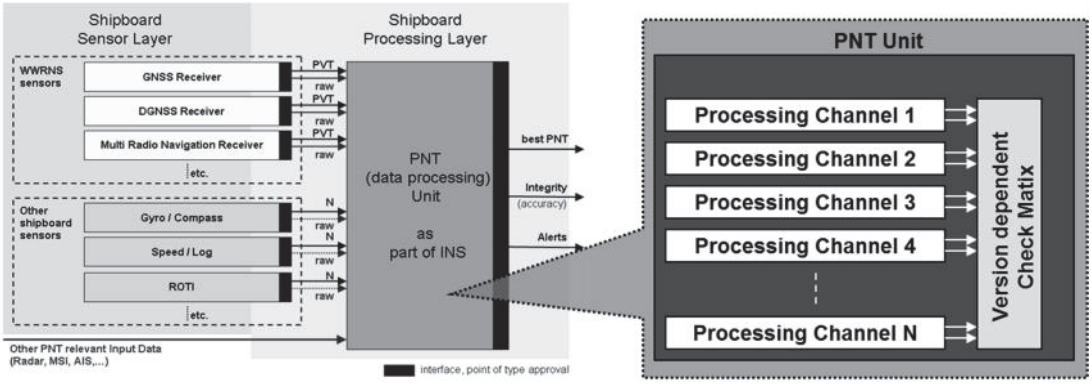


ABBILDUNG 2: REALISIERUNG DER PNT-UNIT (ÜBERNOMMEN VON ABBILDUNG 1-1 AUS [12])[2]

Ein terrestrischer Dienst ist beispielsweise der differentielle GNSS-Dienst (DGNSS) IALA Beacon (IALA: International Association of Lighthouse Authorities), der küstennah zusätzliche codebasierte Korrektur- und Integritätsdaten bereitstellt und damit die aus den GNSS-Entfernungsmessungen bestimmten PVT-Informationen hinsichtlich deren Genauigkeit verbessert. [1] Das DGNSS-Konzept beruht auf der Annahme, dass sich Fehler (Uhrfehler, Ausbreitungsfehler) auf die Absolutposition benachbarter Empfänger ähnlich oder gleich auswirken, nicht aber auf die Positionsunterschiede. Wenn sich mindestens ein GNSS-Referenzempfänger an einer bekannten Position befindet, können die erwarteten Entfernungen zu den bekannten Satelliten mit den gemessenen verglichen werden und die Unterschiede als Korrekturen an die beteiligen DGNSS-Empfänger übermittelt werden. [13]

Trägerphasenbasierte DGNSS-Verfahren oder Echtzeitkinematik (RTK: Real Time Kinematic) GNSS-Verfahren besitzen eine Datenverbindung zwischen einer Referenzstation mit bekannten Koordinaten und einem beweglichen Empfänger, der auch Rover genannt wird. Bei RTK-Messungen müssen die Mehrdeutigkeiten - die ganzzahlige Anzahl an Wellenlängen zwischen der Referenzstation und dem Satelliten - bestimmt werden, um die Position des Rovers zu ermitteln. Gelingt die Bestimmung der Mehrdeutigkeiten, liegt eine „Fix“-Lösung (Mehrdeutigkeitsfixierung) vor, andernfalls spricht man von einer „Float“-Lösung. Im Rahmen des MVT-Projekts wurde MGBAS (Maritime Ground Based Augmentation System) als phasenbasierter DGNSS-Dienst verwendet.

Die Daten der Sensoren und Dienste werden von einem in C++ geschriebenen Datensammler („Recorder“) entgegen genommen, der jedes Datenpaket mit einem Zeitstempel[3] in UTC versieht. Dies ist in Abbildung 3 dargestellt. Der Datenstrom wird von dem Recorder für eventuelle Nachprozessierungen in eine SQLite-Datenbank geschrieben und in Echtzeit per TCP (Transmission Control Protocol) an die PNT-Unit zur weiteren Prozessierung gesendet. Alternativ ist es möglich, mit einem Datenabspieler („Rawdata Caster“) die Sensordaten aus der Datenbank zur Nachprozessierung oder zur Simulation einer Echtzeitanwendung abzuspielen. [1]

[2] WWRNS: weltweit verfügbare Funknavigationssysteme
[3] Die Rechneruhr wird über UDP (User Datagram Protocol) mit den Uhren der GNSS-Empfänger synchronisiert, was die Ungenauigkeit der Systemzeit auf unter 1 ms reduziert. [1]

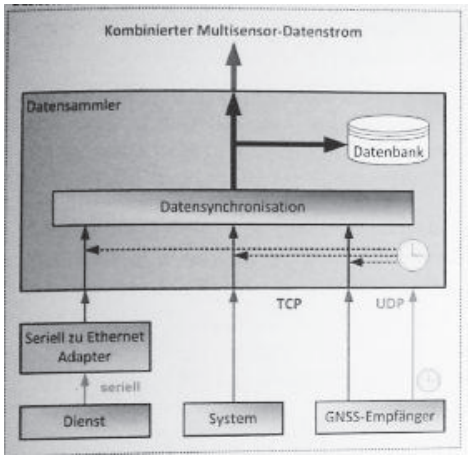


ABBILDUNG 3:SCHEMA DES DATENSAMMLERS (ÜBERNOMMEN VON [1], ABB. 15)

Die Sensordaten werden in der PNT-Unit in Prozessoren verarbeitet. Mit dem Begriff Prozessor ist eine Klasse in einem C++-basierten Multi-Sensor-Echtzeit-Framework gemeint, der einen spezifischen Algorithmus realisiert. Das Framework sorgt für das Datenmanagement (Synchronisierung, Transport, etc. der Daten) und entkoppelt so die Algorithmen von den Datenquellen. Die Prozessoren setzen Sensorfusionsalgorithmen ein und lassen sich zu Prozessierungsketten zusammenfassen, die nach [1] auch „Bestandteil zukünftiger INS sein können“. Die Ketten können in Prozessierungskanälen parallel ablaufen.

Das Ergebnis der Prozessierung sind die PNT-Daten sowie Integritäts- und Alarminformationen, die über eine Schnittstelle als Datenstrom bereitgestellt werden. An dieser Schnittstelle kann eine Anwendung, wie zum Beispiel eine Visualisierung, die PNT-Informationen abgreifen und weiter verarbeiten.

### *1.3.2 PROZESSIERUNGSKANÄLE*

Wie in Abbildung 2 dargestellt, laufen die Prozessierungen in Prozessierungskanälen parallel, in denen Sensordaten und Prozessierungsergebnisse fusioniert werden können. Die Parallelität erlaubt es anschließend, aus mehreren Ergebnissen die geeignetste Lösung wählen zu können. Dies kann die schnellste oder die genaueste Lösung sein. [12] Derzeit sind nach [1] acht Prozessierungskanäle implementiert. Kanal 1 bis 4 sind verschiedene GNSS-basierte Verfahren zur PVT-Bestimmung, von denen 3 (IALA Beacon) und 4 (MGBAS) Ergänzungsdaten terrestrischer Dienste hinzuziehen. Prozessierungskette 2 nutzt das Receiver-Autonomous-Integrity-Monitoring (RAIM).

RAIM ist ein GNSS-basiertes Integritätsüberwachungsverfahren, das fehlerbehaftete Satellitensignale (Ausreißer in der Positionsgenauigkeit) von der Prozessierung einer Positionslösung ausschließen kann. Es nutzt die Redundanz bei mehr als fünf Satelliten aus. Der RAIM-Algorithmus der PNT-Unit bestimmt eine Abschätzung des horizontalen

Positionsfehlers – den Horizontalen Protection Level (HPL). Integrität ist gegeben, wenn der HPL unter einem festgelegten Alarmwert liegt. [1] Die [IMO A.915(22)] legt diesen Wert für küstennahe Navigation bei 25 m und im Hafenbereich bei 2,5 m fest.

Die Kanäle 5 bis 8 nutzen, auf Bayes'schen Filtern basierende, Extended Kalman Filter (EKF) zur Sensorfusion von GNSS-Daten mit den Daten einer inertialen Messeinheit (IMU: Inertial Measurement Unit).

Bei dem EKF werden „normalverteilte Fehler und ein näherungsweise linearisiertes Systemmodell angenommen". [1] Es werden nach [1] entweder die Positions- und Geschwindigkeitsdaten des GNSS-Sensors mit den Daten der IMU fusioniert. Dies nennt man lose gekoppelte (*loosely coupled*) Integration. Oder es werden die GNSS-Rohdaten mit den Daten der IMU fusioniert, was als eng gekoppelte (*tightly coupled*) Integration bezeichnet wird. Das eng gekoppelte Verfahren erlaubt es, Satellitensignale auf Rohdatenebene hinsichtlich ihrer Integrität zu überwachen und entsprechend von der weiteren Prozessierung auszuschließen.

Die acht Prozessierungsketten setzen sich nach [1] aus folgenden Verfahren zusammen, wobei für die GNSS-Daten GPS-Signale (Global Positioning System) verwendet werden:

(1) GPS L1-Code-basiertes Single Point Positioning (SPP)
(2) SPP mit RAIM
(3) codebasiertes differentiellen GPS mit IALA Beacon (CDGPS)
(4) phasenbasiertes differentielles GPS mit MGBAS (PDGPS)
(5) lose gekoppelte IMU mit SPP und RAIM
(6) eng gekoppelte IMU mit GPS
(7) lose gekoppelte IMU mit CDGPS
(8) eng gekoppelte IMU mit PDGPS

### *1.3.3 PNT-DATEN*

Die PNT-Unit erzeugt die folgenden Daten: Positionsdaten als Länge, Breite und Höhe des Consistent Common Reference Point (CCRP)[4] im WGS 84 (World Geodetic System), primäre Navigationsdaten zur Beschreibung der horizontalen Lage und Bewegung des Schiffes, sekundäre Navigationsdaten zur Beschreibung der dreidimensionalen Lage und Bewegung des Schiffes und Zeitdaten in einem gebräuchlichen Format wie UTC (Universal Time Coordinated). Die primären Navigationsdaten beinhalten Geschwindigkeit über Grund (SOG: Speed over ground), Geschwindigkeit durch Wasser (STW: Speed through water), Drehrate (ROT: Rate of Turn), Steuerkurs (Heading) und Kurs über Grund (COG: Course over ground). Die sekundären Navigationsdaten beinhalten den Roll-, Nick-, und Gierwinkel

[4] der CCRP ist die Referenzposition des eigenen Schiffes, auf die sich alle horizontalen Messungen (Richtung, Kurs, Geschwindigkeit, etc.) beziehen. [DIN-EN-62388, 3.12]

(roll, pitch, yaw) des Schiffes. Dazu kommen die PNT-relevanten Integritätsdaten, wie der RAIM HPL.

Die PNT-Daten werden als Datenstrom im JSON-Datenformat (JavaScript Object Notation) per TCP/IP (Transmission Control Protocol/Internet Protocol) und als Textdatei bereitgestellt.

## 1.4 Anwendungsfälle und relevante Navigationsinformationen

Es existieren zwei Anwendungsfälle, für die eine mobile Visualisierung der PNT-Daten benötigt wird: für Präsentationszwecke und Labortests und für das Bergungsschiff *Baltic Taucher II*. Für die Anwendungsfälle sind nicht alle Daten der Ausgabeschnittstelle von Relevanz und die relevanten Informationen sind für beide Anwendungsfälle unterschiedlich. Es ist beiden Anwendungsfällen gemein, dass die Daten – sofern es sinnvoll ist – auf einer Karte dargestellt werden sollen.

### *1.4.1 Anwendungsfall: Präsentationszwecke und Labortests*

Bei diesem Anwendungsfall müssen alle von der PNT-Unit bereitgestellten Daten, die die Position, Geschwindigkeit und Lage des Schiffes sowie die Integrität der Daten beschreiben, zum Zweck der Nachprozessierung („postprocessing“) im Labor oder zur Präsentation der Ergebnisse des MVT-Projekts als simulierte Echtzeit visualisiert werden. Die relevanten Daten sind in diesem Fall:

- der Zeitstempel im UTC-Format
- die Ergebnisse der Prozessierungskanäle (geographische Koordinaten im WGS84 und Höhe in Meter)
- die Roll-, Nick- und Gierwinkel des Schiffes
- die Geschwindigkeitsvektorkomponenten des Schiffes in Nord-, Ost- und Zenitrichtung
- der Kurs über Grund (COG)
- der Schutzgrenzwert der Empfängerautonomen Integritätsüberwachung (RAIM-HPL) in Meter
- der RAIM-Alarmwert (0, 1, 2 oder 3 entsprechen den Zuständen keine Daten, Dienst nicht vorhanden, Position unsicher oder Position sicher)
- die Qualität der Lösung bei Prozessierung mit phasenbasiertem DGNSS (keine Daten, Fix oder Float) (PDGNSS)
- eine Matrix der beobachteten Satelliten bestehend aus 4-elementrigen Vektoren (ID, Elevation, Azimut, Innovation)

Die Innovation ist ein Wert, der beschreibt, wie sehr die erwarteten Pseudoentfernungen von den gemessenen abweichen.

### 1.4.2 ANWENDUNGSFALL: BALTIC TAUCHER II

Die Baltic Taucher II ist ein Bergungsschiff, das in der Ost- und Nordsee Metalle und gefährliche Objekte wie Munition vom Meeresgrund birgt. Durch Kooperation mit der Firma „Baltic Taucher“ kann die Baltic Taucher II im Rahmen des MVT-Projekts als Versuchs- und Demonstrationsträger genutzt werden. [1] Die Baltic Taucher II ist zwar nur 29 m lang, verfügt aber aufgrund ihrer Schiffsnutzung über die Ausrüstung größerer Schiffe, die seitens MVT durch weitere Geräte ergänzt wurde. Die zur Verfügung stehenden Sensoren sind nach Abb. 12 aus [1][5]: Echolot, Speed Log, GNSS Empfänger, Empfänger für IALA Beacon und PDGNSS Korrekturdaten, Radar, AIS, Gyrokompass und eine IMU.

Die Baltic Taucher II fährt im Rahmen ihres Auftrags zu den Koordinaten einer Zielposition, positioniert sich mit vier Seilwinden stabil über der Position und lässt einen Taucher herab, der das zu bergende Objekt, spiralförmig ausgehend von dem Punkt an dem er auf den Grund trifft, sucht. Das Objekt wird anschließend mit einem Kran geborgen. Um die Arbeit des Tauchers zu beschleunigen, ist die Baltic Taucher II an hochgenauen Positions- und Lagedaten interessiert. (Abbildung 4)

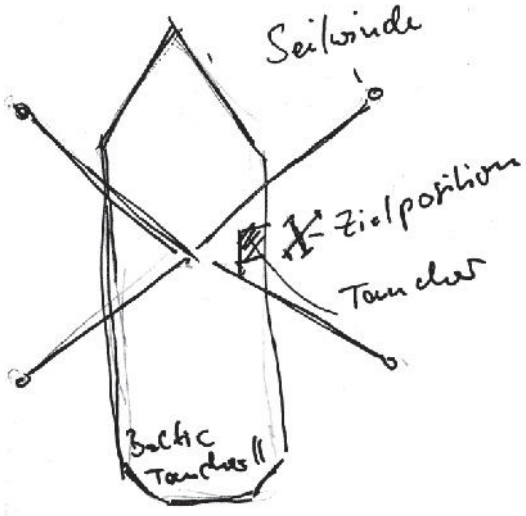


ABBILDUNG 4: SKIZZE DES BALTIC TAUCHER ANWENDUNGSFALLS

Wie im Anwendungsfalldiagramm in Abbildung 5 dargestellt, wird die Nutzerin oder der Nutzer (beispielsweise der Nautiker) die Visualisierung dazu verwenden, um seine momentane Position und Lage auf einer Karte einzusehen und mit den Zielkoordinaten des Objekts, die er zuvor in der Anwendung eingetragen hat, vergleichen, um mit diesen Informationen das Schiff an die gewünschte Position zu navigieren. Es ist vorstellbar, dass er dabei direkt an dem Punkt steht, an dem der Taucher herabgelassen wird und die mobile

[5] Nur Speed Log, Echolot, Radar, AIS und Gyrokompass gehören zur Standardausrüstung. [1]

Anwendung zur Visualisierung der eigenen Navigationsinformationen genutzt wird. Die relevanten PNT-Informationen reduzieren sich für den Nutzer bei diesem Anwendungsfall auf:

- der Zeitstempel im UTC-Format
- das Ergebnis der Prozessierung mit PDGNSS (geographische Koordinaten im WGS84)
- die Qualität der Lösung bei Prozessierung mit PDGNSS (keine Daten, Fix oder Float)
- die Geschwindigkeitsvektorkomponenten des Schiffes in Nord-, Ost- und Zenitrichtung
- der Kurs über Grund (COG)

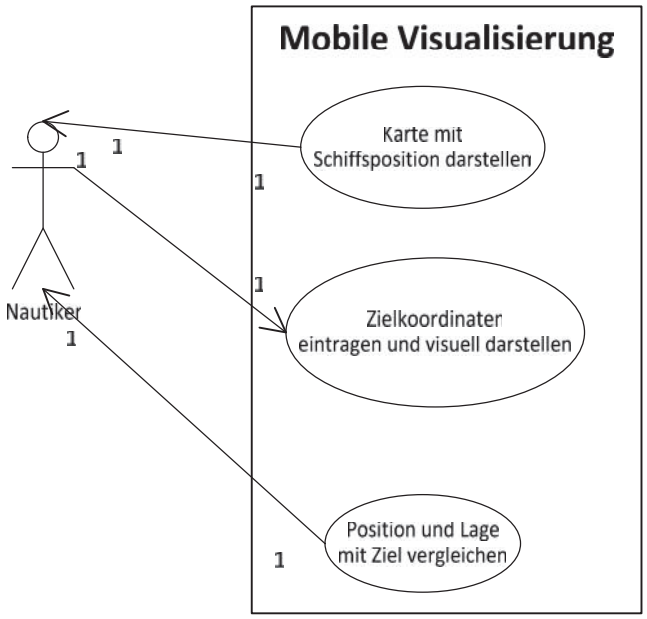


ABBILDUNG 5: ANWENDUNGSFALLDIAGRAMM BALTIC TAUCHER II

# 2 ANFORDERUNGSANALYSE

## 2.1 NUTZUNGSANFORDERUNGEN

Da die PNT-Unit kein kommerzielles Produkt ist, gibt es auch keine Kunden, mit denen die Anforderungen hätten ausgearbeitet werden können. Deshalb wurden im Rahmen der MVT-Arbeitsgruppe die folgenden Nutzungsanforderungen hinsichtlich der mobilen Visualisierung von PNT-Informationen für die in 1.4 beschriebenen Anwendungsfälle festgelegt. Das Ziel hinter der Entwicklung eines Prototyps ist es auch, Erfahrungen mit mobilen Anwendungen auf See zu sammeln.

- Es müssen PNT-Daten visualisiert werden.
- Die Visualisierung beinhaltet eine Karte, die Kartendaten müssen aber nicht von offiziellen Quellen stammen.
- Das Anzeigegerät muss mobil verwendbar sein.
- Das Anzeigegerät darf für die Dauer der Aufgabe nicht ausfallen.
- Das Anzeigegerät muss robust sein gegenüber äußeren Bedingungen.
- Das Anzeigegerät darf den Nutzer bei der Aufgabe nicht behindern oder gefährden.
- Die Visualisierung muss einfach und intuitiv zu bedienen sein.
- Die Einarbeitungszeit muss kurz sein.
- Die Darstellung der Informationen muss verständlich und erkennbar sein.
- Die dargestellten Informationen müssen sich auf das Wesentliche reduzieren lassen.

## 2.2 ERMITTLUNG VON TECHNISCHEN UNTERSETZUNGEN DER NUTZUNGSANFORDERUNGEN

Dieses Kapitel dient der Ermittlung von technischen Untersetzungen zu den oben benannten Nutzungsanforderungen. Untersuchungsgegenstand sind bestehende Standards für Anzeigegeräte auf Brücken, die hinsichtlich ihrer Anwendbarkeit auf die zuvor getroffenen Nutzungsanforderungen untersucht werden.

### *2.2.1 KONFORME STANDARDS*

Für ECDIS gibt es eine geräteunabhängige Spezifikation in Form von verschiedenen Normen und Standards der IEC (International Electronic Commission) und MSC. Die entsprechenden

IMO konformen INS (Integrated Navigation System) und IBS (Integrated Bridge System) Anforderungen finden sich für folgende Anwendungsbereiche in folgenden Normen[6]:

- ECDIS: IEC 60945, IEC 61162, IEC 62288, IEC 61174, MSC.232(82)
- Bildschirm und Anzeige (Display): MSC.191(79)[7]
- Symbole, Begriffe und Abkürzungen: SN/Circ.243[8]

#### 2.2.1.1 Der Leistungsstandard IMO Resolution MSC.191(79)

Die IMO Resolution MSC.191(79) vereinheitlicht die Anforderungen für die Darstellung von navigationsbezogenen Informationen auf einer Schiffsbrücke, um sicherzustellen, dass alle Navigationsanzeigen einer einheitlichen Mensch-Maschine-Schnittstellenphilosophie und -umsetzung folgen. Die Resolution gilt für ECDIS und Radaranzeigegeräte und für alle Anzeigegeräte, die sich auf der Brücke eines Schiffes befinden. Die entsprechenden Leistungsstandards lassen sich auf jedes Anzeigegerät anwenden, das mit dem Navigationssystem verbunden ist oder für das einzelne Leistungsstandards noch nicht angenommen wurden.

#### 2.2.1.2 Das IMO Rundschreiben 243

Symbole, die der Darstellung von operativen Informationen dienen, werden in der IMO „Safety of Navigation Circular“ 243 (SN/Circ.243) beschrieben. Die SN/Circ.243 beschreibt, wie navigationsbezogene Symbole und Begriffe und deren Abkürzungen auf Navigationssystemen an Bord von Schiffen genutzt werden sollten. Sie gilt für alle Systeme an Bord von Schiffen, die der Navigation dienen, wie Radar, ECDIS und AIS.

[6] Von näherer Bedeutung und unter dem Aspekt, dass mit der mobilen Visualisierung kein ECDIS nachgebaut werden soll, sind die Normen zu Bildschirm und Anzeige (MSC.191(79)) und zu Symbolen, Begriffen und Abkürzungen (SN/Circ.243) für die Untersuchung. Anforderungen, die sich explizit auf ECDIS (oder Radar) beziehen, werden unter dieser Vorgabe ignoriert.

[7] „Performance Standards for the presentation of navigational related information on shipborne navigational displays“

[8] “Guidelines for the presentation of Navigation-related Symbols, Terms and Abbreviation”

### 2.2.2 Art der Nutzung

Entsprechend der Anforderung, dass die Visualisierung an Bord eines Schiffes mobil nutzbar sein soll, wird die mobile Nutzung wie folgt definiert:

Die Visualisierung lässt sich ohne Einschränkung der Funktionalität an Bord eines Schiffes mobil – entsprechend der Definition des mobilen Endgerätes - nutzen. Der Nutzer hält das Gerät in der Hand oder befestigt es stationär an seinem Arbeitsplatz (z.B. mit Hilfe einer entsprechenden Halterung). Die stationäre Nutzung wird immer noch als mobil betrachtet, wenn sich das Anzeigegerät (samt Halterung) zu einem anderen Arbeitsplatz transportieren und dort erneut stationär befestigen lässt. Die mobile Nutzung erfordert, dass die Übertragung von Daten zur Darstellung kabellos möglich sein sollte.

Entsprechend der Anforderung, dass die Anwendung die Nutzerin oder den Nutzer nicht behindern oder gar gefährden darf, sollte das mobile Endgerät eine Möglichkeit bieten, es sicher abstellen oder festmachen zu können. Das Anzeigegerät kann für diesen Zweck hardwareseitig die Möglichkeit bieten, es irgendwo fest zu installieren.

Wenn das Gerät fest auf der Schiffsbrücke installiert werden soll, ist zu bedenken, dass die Brücke des Schiffes „überladen“ ist und wenig Platz bietet für eine feste Installation des darstellenden Gerätes (siehe Abbildung 50). Die Brücke ist so aufgeteilt, dass der Nautiker alles von einem Platz aus im Blick hat. Für eine weitere feste Anzeige (z.B. ein Tablet in einer Docking Station) besteht wenig oder kein Platz. Bei der festen Integration des Anzeigegerätes in die Instrumentenarchitektur auf der Schiffsbrücke (oder einem anderen Arbeitsplatz an Bord des Schiffes), darf somit die Funktionalität des Arbeitsplatzes nicht eingeschränkt und dort bestehende Anzeigen nicht verdeckt werden.

### 2.2.3 Bedingungen an Bord eines Schiffes und sich daraus ergebende Konsequenzen für Mobile Navigationsgeräte

Es existieren verschiedene Aspekte an Bord eines Schiffes bzw. auf dessen Brücke, die (fast) immer wirken und damit unmittelbar immer Einfluss auf das Anforderungsprofil einer Visualisierung haben. Die Umgebungslichtbedingungen, die Betrachtungsentfernung und der Betrachtungswinkel sind solche Aspekte, die sich in den bestehenden Standards für Brückenausrüstung niederschlagen. Die Anforderung, dass die dargestellten Informationen in der mobilen Visualisierung erkenntlich sein sollen, unterliegt auch den Auswirkungen eben benannter Aspekte. Für eine mobile Anwendung kommt zusätzlich hinzu, dass auch Einflüsse außerhalb der Schiffsbrücke wirken, wie beispielsweise:

- die Entfernung zum Sender bei kabelloser Datenübertragung
- die Abdeckung des Signals bei kabelloser Datenübertragung
- der direkte Einfluss des Wetters und der Witterung
- die Kompaktheit der Wege auf dem Schiff (steile Treppen, Leitern, enge Gänge, etc.)

- die mechanische Festigkeit und Robustheit des Gerätes gegenüber Herunterfallen

#### 2.2.3.1 Umgebungslichtbedingungen

Es werden in den relevanten Normen (2.2.1) hinsichtlich der Betrachtung von Bildschirmen und der Erkennbarkeit bzw. der Lesbarkeit der darauf dargestellten Informationen drei Umgebungslichtzustände benannt. Licht bei **Tag**, Licht während der **Dämmerung** und Licht bei **Nacht**. Das Umgebungslicht hat Einfluss (und spiegelt sich dementsprechend in den Anforderungen der Normen und Standards wider) auf die Lesbarkeit von Informationen und Text (MSC.191(79), 5.2.1), auf Farben (MSC.191(79), 5.3.2), Kontrast und Helligkeit des Bildschirms (MSC.191(79), 8.1.1), auf den Betrachtungswinkel (MSC.191(79), 8.5) und allgemein alles weitere, das an der Visualisierung oder am Anzeigegerät durch Lichteinstrahlung kaum oder gar nicht mehr erkenntlich sein wird. Umgekehrt hat die Strahlkraft des Bildschirms Einfluss auf die Nachtsicht des wachhabenden Offiziers, was zwingend das Vorhandensein eines Nachtmodus auch für die mobile Visualisierung bedingt.

Die Norm MSC.191(79) geht von Anzeigegeräten aus, die auf der Schiffsbrücke fest installiert sind. Dem folgend beschränken sich auch einzelne Anforderungen der Norm, die Lichtbedingungen berücksichtigen müssen, auf die Lichtverhältnisse auf der Schiffsbrücke. Für die Umsetzung der Anforderungen an die mobile Visualisierung und die angedachte mobile Nutzung müssen auch die Umgebungslichtbedingungen auf dem gesamten Schiff (außerhalb der Brücke, unter Deck, etc.) berücksichtigt werden. Im Sinne bestmöglicher Vereinfachung wird angenommen, dass die Lichtintensität der jeweiligen Umgebung immer einer möglichen Umgebungslichtbedingung zu einer bestimmten Tageszeit auf der Brücke entspricht. Jede Anforderung, die bei Tages-, Dämmerungs- und Nachtlichtbedingungen auf der Brücke gelten muss, gilt demnach auch immer automatisch auf dem Rest des Schiffes.

Tabelle 1 ist zu entnehmen, welche Intensitäten des einstrahlenden Umgebungslichtes zu welcher Tageszeit zu erwarten sind.

| Umgebungsbedingung | Intensität des Lichts |
|---|---|
| Tag | 200 cd/m² ± 50 % |
| Dämmerung | 10 cd/m² ± 50 % |
| Nacht | Dunkelheit (der Bildschirm ist die vorherrschende Lichtquelle) |

TABELLE 1: UMGEBUNGSLICHTBEDINGUNEN (NACH DIN-EN 62388, S. 51 UND TABELLE 1 IEC 62288, „AMBIENT LIGHT CONDITIONS“)

In der [IEC 62288, 4.3.1.1] ist gefordert, dass die maximale Helligkeitsstufe des darstellenden Gerätes eine Mindestintensität von 85 cd/m² in der Mitte des Bildschirms aufweisen sollte. Der Bildschirm sollte hinsichtlich der Nachtsicht des Nutzes bis mindestens 1cd/m² oder darüber hinaus herunterjustierbar sein. Unter allen Umgebungslichtbedingungen sollte das Endgerät Schrift, Symbole und Grafiken lesbar sowie bei der Darstellung einer Karte die

schwarze Farbe bei Tageslichtbedingungen mit nicht mehr als 0,52 cd/m² Intensität und visuell von einem dunkelgrauen Hintergrund unterscheidbar darstellen.

Diverse online abrufbare Vergleiche der Bildschirme von gegenwärtig erhältlichen Endgeräten, wie es beispielsweise bei [14] nachzulesen ist, belegen, dass moderne Anzeigegeräte die Normvorgaben hinsichtlich der Helligkeit– selbst bei stark reduzierten Helligkeitseinstellungen auf unter 50% - weit übertreffen und hinsichtlich der schwarzen Farbe unterbieten. Anzeigegeräte, die keine eigene Hintergrundbeleuchtung haben und damit nicht für den Einsatz bei Nacht tauglich sind, da Informationen nicht mehr erkennbar sind, sind entsprechend MSC.191(79) nicht erlaubt. Dies trifft aber für moderne mobile Endgeräte nicht mehr zu.

#### 2.2.3.2 Betrachtungsentfernung und Nennbetrachtungsabstand

Die Betrachtungsentfernung in den relevanten Normen und Standards geht von der Situation auf einer Schiffsbrücke aus. Die Bildschirme und Kontrollen des Navigationssystems sind fest installiert und die Position, aus der der oder die Nutzer mit dem System arbeiten ist variabel im Rahmen der Größe der Brücke.

Bei der mobilen Visualisierung verhält es sich anders herum. Die Entfernung zu den Kontrollen und zum Bildschirm ist relativ gleich (wenn die Visualisierung gemäß 2.2.2 mobil und nicht stationär verwendet wird) und die Position ist - entsprechend der Definition der Mobilität – variabel, bezogen auf das ganze Schiff. Einschränkungen bei der Arbeit ergeben sich hier eher durch Verbindungsabbrüche bei zu großer oder ungünstiger Entfernung zur Datenquelle.

Als Faustformel gibt die Norm an, dass die Schriftgröße (2.2.6.1) in Millimeter nicht weniger als das 3,5-fache des Nennbetrachtungsabstands in Metern aber auch nicht weniger als 11 Pixel betragen sollte. Der Nennbetrachtungsabstand ist der Abstand, aus dem die Anzeige betrachtet wird. Für ein mobiles Endgerät kann angenommen werden, dass dieser Abstand einer durchschnittlichen Armlänge entspricht. Die Norm empfiehlt, dass er in der beiliegenden Dokumentation angegeben wird. [IEC 62288, 4.3.2.1]

Für navigationsbezogene Symbole (2.2.6.4) wird ein Nennbetrachtungsabstand von 1 m angenommen [IEC 62288, Anhang A.4]. Ein Symbol sollte auf dem Nennbetrachtungsabstand mindestens ein Größe-zu-Abstand-Verhältnis von 5 mm pro Meter (17 Bogenminuten) aufweisen.

Der Abstand zwischen Gerät und Betrachter beträgt bei der mobilen Visualisierung bei mobiler Verwendung nicht mehr als die Armlänge des Benutzers. Nimmt man für die mobile Verwendung testweise als Nennbetrachtungsabstand eine Armlänge von 0,74 m an, wie es im 50. Perzentil bei Männern nach DIN 33402-2:2007 als „Reichweite nach vorne“ aufgeführt ist, dann sollte die Schriftgröße nicht geringer sein, als 3,5*0,74 mm. Das heißt nicht geringer als 2,6 mm. Hingegen bei (optionaler) stationärer Nutzung muss berücksichtigt werden, dass der Nennbetrachtungsabstand größer als die Armlänge ist.

#### 2.2.3.3 Betrachtungswinkel und Schiffsbewegungen

Die Norm fordert, dass Schrift, Symbole und Grafiken aus allen Blickrichtungen, und unter Berücksichtigung des Umgebungslichtes, auf das Anzeigegerät von zumindest zwei Nutzern, stehend oder sitzend, von allen Positionen auf der Schiffbrücke aus gelesen werden können. [MSC.191(79), 8.5] Bei mobiler Nutzung der Visualisierung beträgt die Entfernung, aus der der Nutzer auf die Anzeige schaut, maximal eine Armlänge. Bei einem mobilen Endgerät kann immer gewährleistet werden, dass die Betrachtung relativ zentral stattfindet und dadurch Reflektionen auf der Anzeige minimiert oder ausgeschlossen werden, bzw. dass sich das Gerät immer so drehen lässt, dass die Darstellung erkennbar bleibt. Die Anforderung an einen zweiten oder noch weitere Nutzer ergibt in dem Fall auch keinen Sinn und kann als nicht erforderlich betrachtet werden. Bei stationärer Verwendung an einem Arbeitsplatz hingegen sollte diese Norm bedacht werden, wenn die Erkennbarkeit der Darstellung aus allen Positionen wichtig ist.

Es ist zu bedenken, dass (im Handel erhältliche) mobile Geräte oft beim Drehen oder Kippen die Anwendung neu ausrichten. Das heißt, sie stellen die Anwendung nie auf dem Kopf dar. Dieser Effekt kann bereits bedingt durch die Bewegungen des Schiffes auftreten. Um dies zu vermeiden, sollte die Anwendung eine Einstellmöglichkeit aufweisen, mit der sich dieses Verhalten abstellen lässt.

### *2.2.4 Robustheit und Schutzgrade gegenüber äußeren Einflüssen*

Eine Anforderung an die mobile Visualisierung verlangt, dass das Anzeigegerät robust gegenüber äußeren Einflüssen ist. Äußere Einflüsse können Spritzwasser, Chemikalien, Öl oder aggressive Stoffe, Salz (in Luft und Wasser) oder der direkte Einfluss der äußeren Temperatur oder der Temperatur im Schiffsinneren sein.

#### 2.2.4.1 Witterungsbedingungen und Schutzgrad

Auf der Schiffsbrücke wirkt das Umgebungslicht unmittelbar auf die Erkennbarkeit der Darstellung. Außerhalb der Brücke ist das Anzeigegerät zusätzlich den Witterungsbedingungen ausgesetzt, was direkt oder indirekt Einfluss auf die Verwendbarkeit der Anwendung hat. Witterungsbedingungen wie Regen, Wind, Salzwasser, etc. wirken direkt auf das Anzeigegerät und dessen Nutzer. Der Einfluss des Wetters kann so weit gehen, dass das Gerät nicht mehr benutzbar ist; beispielsweise durch Wassereinfluss oder durch Sicherheitsgründe, die den Nutzer dazu zwingen, es aus der Hand zu geben (2.2.4.2). Solche Bedingungen sind nicht beeinflussbar – ihren Auswirkungen kann aber durch Sicherheitsvorkehrungen oder Schutz des Anzeigegeräts begegnet werden. Wenn das Anzeigegerät über einen berührungsempfindlichen kapazitiven Bildschirm verfügt, ist zu

bedenken, dass bereits geringe Wassermengen auf dem Bildschirm ein Erkennen der Gesten verhindern; was diese Technologie eigentlich von der Verwendung auf hoher See ausschließt.

Indirekt wirkt das Wetter zudem auf die Umgebung (Leitern, Treppen, etc.). Das Anzeigegerät darf den Nutzer entsprechend Anforderung bei der Bewegung und Arbeit auf dem Schiff nicht einschränken, behindern oder gar gefährden. Es sollte entweder klein genug sein, um in einer Tasche oder am Armgelenk befestigt zu werden oder über eine entsprechende Tragevorrichtung verfügen, die den Nutzer nicht einschränkt, behindert oder gefährdet, bzw. über einen Ständer verfügen, mit dem es abgestellt werden kann.

Die Schutzart oder der IP-Code (*international protection*) gibt nach DIN 40050 oder IEC 60529 den Grad des Schutzes von elektrischen Geräten gegenüber dem Eindringen von Fremdkörpern wie Fingern aber auch gegenüber Staub und Wasser, Dampf, Öl, Säuren, etc. an. Der IP-Code besteht aus den Buchstaben *IP* gefolgt von 2 Ziffern, wobei die erste Ziffer den Schutz gegen Fremdkörper und die zweite den Schutz gegen Flüssigkeiten angibt. Entsprechende genormte IP-Schutzartprüfungen können Geräte in einen IP-Schutzgrad einordnen. Der Buchstabe X steht für eine Kennziffer, die nicht angegeben werden muss. Für ein mobiles Anzeigegerät an Bord eines Schiffes und zur Erfüllung der Anforderung an die Robustheit gegenüber äußeren Bedingungen sind somit die folgenden Schutzgrade für die mobile Visualisierung relevant:

- IP54 bzw. IP55 gegen Staubablagerungen und Sprühwasser bzw. Strahlwasser aus allen Richtungen. Es kann an Deck des Schiffes davon ausgegangen werden, dass Wasser aus allen Richtungen kommt.
- IP56 bzw. IP57 gegen Staubablagerungen und Wassereindringen durch kurze Überflutung bzw. kurzes Eintauchen in Wasser. Dies ist der Fall bei unruhiger oder stürmischer See.
- IP58, wenn die Anwendung dauerhaft unter Wasser nutzbar sein soll.

#### 2.2.4.2 Das Schiff als Einflussfaktor für die mobile Visualisierung und Sicherheit des Nutzers

Das Schiff selbst wirkt in vielfacher Hinsicht auf die Leistung der mobilen Visualisierung, die für ein fest installiertes System wie ECDIS in der Form nicht gegeben sind. Störende bzw. abschirmende Effekte bedingt durch das verbaute Material (Wände, Decken, etc.) können auf die Qualität der (drahtlosen) Übertragung wirken bzw. diese komplett unterbinden. Die Angabe einer Maximalreichweite im Handbuch ist somit abhängig vom jeweiligen Einsatz und sollte vor Ort überprüft werden.

Schiffsbewegungen bergen die Gefahr des Fallenlassens des Anzeigegeräts und bedingen auch, dass die Anzeige schlechter ablesbar ist und den Nutzer eventuell von seiner eigentlichen Kernaufgabe ablenkt. Eine ungeschriebene Regel an Bord von Schiffen lautet „eine Hand für den Mann, eine Hand für das Schiff“, was bedeutet, dass jeder einzelne an Bord jederzeit für seine eigene Sicherheit zu sorgen hat. Die mobile Visualisierung benötigt

mindestens eine Hand des Nutzers (zum in der Hand halten) – oft aber beide Hände (zum Halten und Bedienen). Hieraus leitet sich auch die Anforderung ab, die Sicherheit des Nutzers an Bord des Schiffes vorrangig zu gewährleisten. Damit die mobile Visualisierung auch weiter nutzbar bleiben kann, kann sie um die Option einer stationären Nutzung, beispielsweise mit einem Ständer oder einer Halterung am Körper erweitert werden.

Die mobile Visualisierung sollte darüber hinaus auch gegen absichtliches oder unbeabsichtigtes Fallenlassen vom Nutzer ausreichend geschützt oder so robust sein, dass die Funktionsweise nicht beeinträchtigt wird.

### *2.2.5 INTUITIVE BEDIENBARKEIT*

Es ist sinnvoll, dass sich die mobile Visualisierung hinsichtlich Informationsanordnung und Aufteilung der Bildschirmanzeige an dem orientiert, was der Nutzer bereits kennt – beispielsweise von einem ECDIS-Gerät. Das heißt, die Nutzerin oder der Nutzer sollte intuitiv erkennen, in welchem Bereich der Anzeige welche Informationen bzw. Eingabe- oder Kontrollelemente zu finden sind.

#### 2.2.5.1 Aufteilung des Bildschirms und Anordnung der Informationen

Auf einem ECDIS dargestellte Daten und Kontrollfunktionen sind logisch so gruppiert, wie es für die momentane Aufgabe[9] sinnvoll ist. Die Aufgaben, die der Nutzer mit der mobilen Visualisierung durchführt, sind in den Anwendungsfällen (1.4) beschrieben. Für die Aufgabe wichtige und vorrangige Informationen sind permanent sichtbar oder stechen gegenüber anderen Informationen heraus. Der Standard empfiehlt dazu, dass sich dies durch Änderungen von Positionierung, Größe oder Farbe des dargestellten Elements erzielen lässt. [IEC 62388, 6.3.1.1] Übergeordnete Gruppierungen zur Einteilung von Informationen, wie sie in IEC 62288 Anhang E ausführlicher benannt werden, könnten für die darzustellenden PNT-Daten sein:

- Alarme und Hinweise
- Informationen, das eigene Schiff betreffend
- Navigationshilfsprogramme und –anzeigen
- Entfernungs- oder Maßstabsanzeigen
- Systemeinstellungen

Bei anderen Darstellungsparametern ist eine einheitliche Darstellungsform wichtig. Dies betrifft die Einheit der Messgröße bei numerischen Informationen (sofern vorhanden), die Kennzeichnung der Quelle und Bedeutung einer Information sowie qualitative oder quantitative Bewertungen der Gültigkeit oder der Integrität. Ungültige Informationen oder

[9] Die Aufgabe („task-at-hand“) meint eine konkrete navigationsbezogene, von einem Nutzer durchgeführte, Maßnahme – beispielsweise Routenplanung. [IEC 62288, 3.48]

Informationen mit geringer oder keiner Integrität können quantitativ zusammengefasst werden (absolut oder prozentual). Zum Beispiel bei Verwendung von Farbkodierung (2.2.6.2) entspricht niedrige Integrität qualitativ der Farbe Gelb und ungültige Information qualitativ der Farbe Rot.

Organisatorisch orientieren sich ECDIS-Anzeigen nach der in [MSC 191(79), 5.1.3] geforderten Aufteilung in einen operativen Bereich der Arbeitsanzeige, in dem beispielsweise die Karte dargestellt wird und einen Bereich, für die Nutzerdialoge (Menüs, Daten, Kontrollflächen, etc.). Exemplarisch sei diese Aufteilung mit Abbildung 6 dargestellt. Die „Application Area“ ist der Bereich der Kartendarstellung. Die „Sidebar“ beinhaltet verschiede Dialogbereiche für Informationen, Menüs und Nutzereingaben, die dauerhaft bzw. nicht dauerhaft zur Verfügung stehen müssen. Hinsichtlich der Anforderung an eine intuitive Bedienbarkeit ist es sinnvoll, wenn sich die mobile Visualisierung an diesem Schema orientiert.

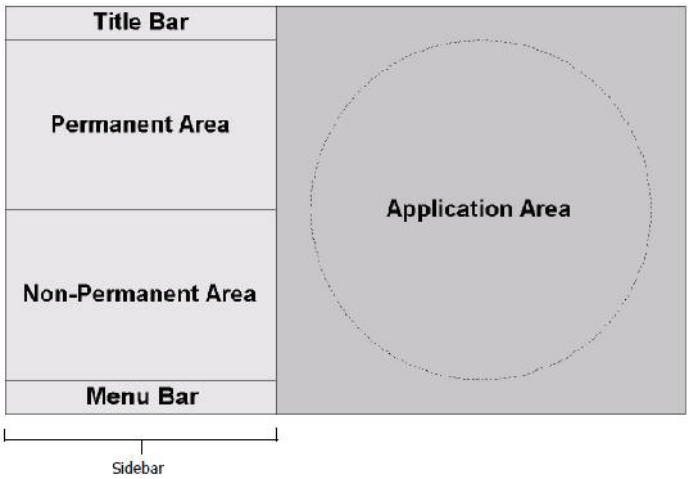


ABBILDUNG 6: ALLGEMEINE BILDSCHIRMEINTEILUNG (ÜBERNOMMEN VON [15])

### 2.2.5.2 Nutzerschnittstelle und Touchscreen-Technologie

Benutzerschnittstellen beschreiben diejenigen Schnittstellen, mit denen der Nutzer das Verhalten der Anwendung, deren Inhalt, Art der Darstellung oder Umfang manipulieren kann. Per Definition besitzt das mobile Endgerät Schnittstellen zur Ein- und Ausgabe. Typischerweise erfolgt bei mobilen Geräten die Nutzerinteraktion über einen Touchscreen (berührungsempfindlicher Bildschirm). Die Bedienung mit einer Maus oder einem Trackball ist zwar prinzipiell möglich, aber untypisch. Dadurch ergeben sich Unterschiede zu Systemen wie ECDIS. Es gibt keine Maus und damit auch keinen Mauszeiger auf dem Bildschirm. Auch wenn es möglich ist, externe Eingabegeräte an das mobile Gerät anzuschließen, die Standardeingabe sollte immer die Nutzereingabe über den Touchscreen sein. Merkmale dieser Art der Interaktion sind, dass es beispielsweise keine „Mouseover“-Ereignisse gibt, das Gerät dafür aber auf Bewegungen (Drehen, Kippen) reagieren kann.

Der Vollständigkeit halber sei hier auf zwei wesentliche für die mobile Visualisierung in Frage kommende Prinzipien, nach denen Touchscreens[10] funktionieren, hingewiesen: resistiv und kapazitiv. Resistive Touchscreens basieren auf zwei elektrisch leitfähigen aber zunächst getrennten Schichten, die durch Druck örtlich verbunden werden und so einen elektrischen Widerstand erzeugen, dessen Position ermittelt werden kann. Kapazitive Touchscreens basieren nicht auf Druck, sondern auf der Veränderung eines elektrischen Feldes an der Glasoberfläche des Bildschirms durch einen Leiter wie beispielsweise den menschlichen Körper.

Die Vorteile der resistiven Technologie sind geringe Produktionskosten, hohe Widerstandsfähigkeit gegenüber Schmutz und Wasser und die Möglichkeit, das Gerät auch mit Handschuhen oder einem Stift bedienen zu können. Nachteilig ist, dass der Kontrast des Bildschirms durch die verschiedenen Schichten gemindert wird und Multi-Touch (mehrere Gesten gleichzeitig) nicht möglich ist.

Kapazitive Touchscreens bieten aufgrund der Glasscheibe ein besseres Bild und sie sind hochempfindlich gegenüber Gesten und unterstützen Multi-Touch. Da diese Technologie jedoch auf die Leitfähigkeit des menschlichen Körpers angewiesen ist, funktioniert sie nicht, wenn der Nutzer Handschuhe trägt oder einen Stift verwendet. Der Glasbildschirm ist auch leichter zerbrechlich als ein Bildschirm aus Plastik.

Für die mobile Visualisierung muss zwischen den Vor- und Nachteilen beider Technologien abgewogen werden. Dabei spielen auch das Schiff und der angedachte Einsatzzweck der Anwendung eine Rolle. Sollte die Visualisierung unter Bedingungen verwendet werden, bei denen von Kontakt mit Wasser auszugehen ist, dann sollte auf die Verwendung eines kapazitiven Bildschirms, wie bereits in 2.2.4.1 erwähnt, verzichtet werden. Dies ist beim Anwendungsfall Baltic Taucher II der Fall.

#### 2.2.5.3 Konfigurationsschnittstelle und Systemeinstellungen

Die Konfigurationsschnittstelle erlaubt das Manipulieren von Inhalt und Art der Darstellung der Anwendung entsprechend der Wünsche des Benutzers. Wenn die Einstellungen ein Unterprogramm der Anwendung selbst sind, kann die Konfigurationsschnittstelle auch als Teil der Benutzerschnittstelle betrachtet werden. In dem Fall müssen Möglichkeiten vorhanden sein, die Einstellungen des Nutzers persistent zu speichern. Alternativ oder zusätzlich ist es möglich, die Anpassung der Einstellungen über externe Konfigurationsdateien anzubieten. In dem Fall kann es als separate Schnittstelle betrachtet werden. Die Speicherung der Systemeinstellungen findet auf dem mobilen Gerät statt.

[10] nach: http://de.wikipedia.org/wiki/Touchscreen

#### 2.2.5.4 Kommunikationsschnittstelle

Die Mobile Visualisierung benötigt eine eindeutig definierte Schnittstelle zum Empfang der Daten, die visualisiert werden sollen und gegebenenfalls ein Datenformat in dem die Daten übertragen werden. Typischerweise beruht die Kommunikationsschnittstelle, mit der Systeme wie ECDIS, Schiffssensoren und andere elektronische Geräte verbunden sind, auf dem NMEA-0183-Standard. NMEA 0183 beinhaltet auch ein standardisiertes Datenformat. Die Übertragung erfolgt über Leitungen (Bussysteme). Die mobile Visualisierung strebt gemäß der Anforderung, dass das Gerät mobil verwendbar ist, eine kabellose Kommunikation an. Demnach muss eine entsprechende Schnittstelle zur kabellosen Datenübertragung vorhanden sein. Da die Ausgabeschnittstelle der PNT-Unit bereits definiert ist, muss die mobile Visualisierung zur Verarbeitung der PNT-Daten mit dieser Schnittstelle kabellos arbeiten können.

Bei der mobilen Visualisierung spielt die Entfernung des Anzeigegerätes zur Datenquelle eine Rolle. Bei kabelgebundener Übertragung ist die Entfernung durch die Länge des Kabels begrenzt, bei kabelloser Übertragung durch die Reichweite des Signals und Einflussfaktoren des Schiffes (Abdeckung, Material). Beides wirkt einschränkend auf den mobilen Charakter der Visualisierung. Eine empfohlene Reichweite sollte im Handbuch vermerkt sein. Ein Kabel darf, entsprechend der vereinbarten Anforderung, den Betrieb an Bord des Schiffes nicht behindern oder gefährden.

### *2.2.6 VERSTÄNDLICHKEIT DER DARSTELLUNG*

Um der Anforderung zu genügen, dass die Darstellung der Informationen verständlich sein soll, ist es sinnvoll, die Informationen so darzustellen, wie es die Nutzerin oder der Nutzer (beispielsweise der Nautiker) aus von ihr oder ihm vertrauten Anwendungen (beispielsweise ECDIS) kennt.

Die IMO Resolution MSC.191(79), die IMO SN/Circ.243 und die DIN-EN-62388 Kapitel 6 „*Displays*" sind die Basis für Anforderungen an die Darstellung von Navigationsdaten, Navigationssymbolen, -begriffen und -abkürzungen auf Anzeigegeräten an Bord von Schiffen. Systeme wie ECDIS basieren auf unter anderem diesen Standards und Normen. Die Leitbilder dieser Standards, die in den Absätzen 5 und 8 der MSC.191(79) beschrieben werden, gelten für alle Anzeigen auf einer Schiffsbrücke („The general principles of these standards are applicable for all displays on the bridge of a ship"). [MSC.191(79)] Aus Gründen der Vereinfachung und im Sinne der mobilen Visualisierung werden diese auf der Brücke als auch außerhalb der Brücke als geltend betrachtet.

#### 2.2.6.1 Schrift und Sprache

Der Stil von dargestellter Schrift sollte der Norm gemäß nicht kursiv („non-italic“) und serifenlos („sans-serif“) sein. Die Schriftgröße ist an die typischen Entfernungen (auf einer Schiffsbrücke) des Nutzers zum Bildschirm angepasst. [MSC.191] Entsprechend 2.2.3.2 wird die Schriftgröße bei mobiler Verwendung der Visualisierung an die dort ausgeführten Entfernungen angepasst. Für die mobile Visualisierung sollte die Schriftgröße so gewählt werden, dass der Nutzer sie auf der Entfernung einer Armlänge noch erkennen kann.

Die verwendete Sprache ist gemäß Norm anspruchslos und leicht zu verstehen und maritime Begriffe und Abkürzungen sollten den Fachbegriffen, wie sie in der Tabelle „List of Standard Terms and Abbreviations“ in Anhang 2 auf Seite 2 der SN/Circ.243 nachzulesen sind entsprechen. [MSC.191(79), 5.2.3] Die Verwendung bestehender Begriffe und Abkürzungen muss auch für die mobile Visualisierung gelten.

#### 2.2.6.2 Farben, Intensität und Nachtmodus

Farben müssen gemäß Norm ausreichend Kontrast gegenüber dem Hintergrund bei allen Lichtverhältnissen haben. Ein Nachtmodus ist laut Norm erforderlich [MSC.191(79), 5.3.2]. Im Nachtmodus werden die Information als heller Vordergrund auf einem dunklen nicht reflektierenden Hintergrund dargestellt. So die Sicht von nebenstehenden Personen bei Dämmerung oder Dunkelheit nicht gestört.

Bei Farbkodierungen sollten sich alle Farben innerhalb eines Farbsatzes klar gegeneinander abgrenzen. So sollte Rot immer der Kodierung von Alarminformationen dienen. Blinkende Informationen sollten unbeantworteten Alarmen vorbehalten sein. Auch sollte Farbkodierung nie alleinstehend, sondern immer zusammen mit anderen Attributen wie Größe, Form oder Orientierung verwendet werden. [MSC.191(79), 5.5.3]

Mehrfarbige Bildschirme oder die Verwendung von Farbkodierungen sind nach MSC.191(79) grundsätzlich erlaubt aber nicht erforderlich, wenn es keine explizite Anforderung dafür gibt. Wenn keine Farbkodierung verwendet wird oder verwendet werden kann, muss die Farbe Rot auch nicht eine Alarminformation kodieren. Dies erlaubt demnach nicht nur die Verwendung monochromatischer Anzeigen, sondern auch die Verwendung monochromatischer Anzeigen, die Informationen in der Farbe Rot darstellen. Dazu zählen beispielsweise LED-basierte (light-emitting diode, Leuchtdiode) Anzeigen.

#### 2.2.6.3 Icons und Symbole

Nach Peirce (aus [16]) ist ein Ikon ein Zeichen oder Bildzeichen, das mit dem Objekt, das es repräsentiert, Ähnlichkeit aufweist. Bei Symbolen fehlt diese logische Verknüpfung zwischen Zeichen und Bedeutung hingegen und muss zuvor gelernt worden sein.

In den Normen wird diese Unterscheidung nicht konsequent behandelt und ikonische und symbolische Zeichen teilweise synonym verwendet. Der Begriff „Icon“ taucht in der SN/Circ.243 gar nicht auf und in der MSC.191(79) in nur einem Paragraphen. Dort wird gefordert, dass, wenn Icons verwendet werden, sollte deren Zweck anhand Aussehen, Platzierung und Gruppierung intuitiv (das heißt mit den Erwartungen des Benutzers übereinstimmend) erkennbar sein. [MSC.191(79), 5.2.4] Ansonsten ist nur von Symbolen die Rede. Erst die IEC 62288 definiert den Begriff des Icons:

Ein Icon ist gemäß [IEC 62288, 3.26] ein graphisches Symbol mit einer bestimmten Bedeutung, das Informationen unabhängig einer Sprache übermittelt. Icons können verwendet werden zur visuellen Erkennung oder Verstärkung einer Textbeschreibung, um eine Funktion aufzurufen oder um ein Objekt mit dem Cursor zu öffnen.

Die Verwendung von Icons ist aber nicht explizit gefordert. Kontrollen für allgemeine (An, Aus, Ruhemodus, Helligkeit, Kontrast) oder aufgabenbezogene Bedienelemente des Anzeigegerätes sollten anhand einer englischsprachigen Bezeichnung identifizierbar sein. Soll zusätzlich ein Icon angezeigt werden, sollte dieses aber den Tabellen E.3, E.4 oder E.5 im Anhang E der IEC 62288 folgen. [IEC 62288 4.3.4.2]

Wenn graphische Symbole für die Darstellung von navigationsbezogenen Informationen auf einer Karte verwendet werden, sollten sie aus Gründen der Wiedererkennbarkeit und um die Anforderung nach einer intuitiven Darstellung zu erfüllen den Symbolen in Anhang A der IEC 62288 / Anhang J der IEC 62388 oder denen im Anhang der SN/Circ.243 „Navigation-related Symbols“ entsprechen. Sollte dort für einen bestimmten Zweck kein Standardsymbol aufgezeigt sein, kann ein anderes Symbol verwendet werden. Für die mobile Visualisierung trifft dies derzeit nur auf die Symbole, die das eigene Schiff repräsentieren, zu.

#### 2.2.6.4 Darstellung des eigenen Schiffes

Wenn Messgrößen visualisiert werden, die sich auf das eigene Schiff beziehen, wie zum Beispiel der momentane Kurs, sollten diese sich auf den CCRP beziehen. [MSC.192, 5.9.1] Kurse, Strecken und Messwerte die neu in die oder zwischen bereits bestehenden Merkmalen der Visualisierung gezeichnet werden, sollten eine Genauigkeit aufweisen, die nicht kleiner als Maßstab und Auflösung des Anzeigegerätes ist. [IEC 62288, 5.7.2.1]

Die Norm fordert, dass bei einer grafischen Darstellung des eigenen Schiffes, es dem Nutzer möglich sein sollte, zwischen einem maßstabsgetreuen Schiffsumriss („true scale size“) und einem vereinfachenden Symbol gemäß SN/Circ.243 zu wechseln. Die Größe des Umrisses des eigenen Schiffes oder des vereinfachenden Symbols, sollte dabei der maßstabsgetreuen Größe des eigenen Schiffes entsprechen, aber nicht kleiner sein als 6 mm. [MSC.191(79), 6.1.1] Im Sinne der Anforderung an eine verständliche und intuitive Darstellung, wird das eigene Schiff wie in Abbildung 7 dargestellt.

ABBILDUNG 7: LINKS VEREINFACHTES SYMBOL, RECHTS MASSSTABSGETREUER UMRISS (NACH IEC 62288, TABELLE A.1)

Die Anwendung kann die Möglichkeit bieten, automatisch von der Darstellungsform des maßstabsgetreuen Umrisses zum vereinfachten Symbol zu wechseln, wenn die dargestellte Breite des eigenen Schiffes 3 mm unterschreitet. [IEC 62288 5.1.1.1] Diese Anforderung kann so auch für die mobile Visualisierung übernommen werden.

### 2.2.6.5 Begriffe und Abkürzungen

Der Norm gemäß sollten Begriffe in Kleinbuchstaben (Ausnahmen bilden Wörter, die mit einem großen Buchstaben beginnen) und Abkürzungen in durchgängig großbuchstabigem Text dargestellt werden. Es gibt aber Ausnahmen, bei denen die Norm Kleinbuchstaben erlaubt (als Beispiel wird „dGNSS“ genannt). Abkürzungen können kombiniert werden. Wenn die Abkürzung des Begriffs *relativ* (REL) mit einer anderen Abkürzung kombiniert wird, dann sollte *relativ* mit „R“ anstatt „REL“ abgekürzt werden. Wenn Informationen mit SI-Einheiten dargestellt werden, dann sollte deren entsprechende Abkürzung verwendet werden. [SN/Circ.243, Anhang 2] Eine Liste der Standardbegriffe und der entsprechenden Abkürzungen befindet sich, alphabetisch sortiert nach jeweils beidem, auf Seite 6 des Anhangs 2 der RESOLUTION SN/Circ.243.

### 2.2.6.6 Andere nicht optische oder akustische Hinweise

Mobile Anzeigegeräte können die Möglichkeit bieten, den Nutzer nicht nur optisch oder akustisch auf Ereignisse hinzuweisen, sondern auch haptisch mittels Vibrationen. Dies ist möglich, da davon ausgegangen werden kann, dass der Nutzer oft Kontakt mit dem Gerät hat. Da nicht immer ein direkter Kontakt des Nutzers zum Anzeigegerät besteht, sollte ein Vibrationsalarm oder -hinweis nicht alleine auftreten, sondern immer in Verbindung mit mindestens einem anderen (visuell oder akustisch). Dies ist in Anlehnung an Anforderung hinsichtlich Farbkodierung.

#### 2.2.6.7 Darstellung von Karteninformationen und Herkunft der Kartendaten

Vom Nutzer erstellte Graphiken in der Karte können gemäß [IEC 62288, 6.2.5] aus Linien, Symbolen und Referenzpunkten bestehen und sich entweder auf das eigene Schiff oder auf eine geographische Position beziehen. Die Darstellung von nicht offiziellen proprietären Karteninformationen oder von Informationen, die der Nutzer hinzugefügt hat, sollte - gemäß [MSC.191(79), 6.2.2] und soweit machbar - den IHO Standards (Farben, Symbole) folgen. Andernfalls sollte ein klarer Hinweis gegeben sein, wenn die Präsentation nicht mit den IHO Standards übereinstimmt. Status, Quelle und Aktualität der Kartendaten sollten immer ersichtlich sein.

Der in den Anforderungen 6.2.1 bis 6.2.3 verwendetet Begriff „as far as practical“, der mit „soweit praktisch machbar“ übersetzt werden kann, deutet darauf hin, dass die Darstellung von nicht offiziellen Kartendaten nicht kategorisch untersagt ist, solange die Herkunft der Daten angegeben wird. Das bedeutet, es können auch freie Kartendaten, die nicht konform zum S-52-Standard sind, verwendet werden, wenn die Herkunft entsprechend angegeben wird.

#### 2.2.6.8 Fehler in der Darstellung

Der Nutzer sollte auf Fehler in der Darstellung hingewiesen werden, wenn die Darstellung der PNT-Informationen nicht mehr mit der realen Situation übereinstimmt und die Anforderung, dass die aktuellen PNT-Daten visualisiert werden sollen, nicht mehr erfüllt werden kann. Für den Fall, dass ein hardware- oder softwareseitiger Fehler (was sich beispielsweise in einem Einfrieren des Bildes äußern kann) für Präsentationsfehler ursächlich ist, verlangt die [MSC.191(79), 5.6.3] bereits, dass der Nutzer darauf aufmerksam gemacht wird. Eine Möglichkeit diese Anforderung umzusetzen, ist nach [IEC 62288, 4.7.3.1] die dauerhafte Einblendung einer fortlaufenden Uhrzeit oder blinkender Punkte. Da bei dem mobilen Endgerät die Datenübertragung per Definition kabellos erfolgt, sollte der Nutzer auch auf Präsentationsfehler hingewiesen werden, die durch Übertragungsfehler verursacht werden. Eine Möglichkeit dies umzusetzen besteht darin, den Zeitstempel der PNT-Daten mit einer lokalen Zeit auf dem Endgerät zu vergleichen und die Nutzerin oder den Nutzer dementsprechend zu warnen.

### *2.2.7 Technologie des Anzeigegerätes und Dimmen des Bildschirms*

Zur Umsetzung der Anforderung, dass das Anzeigegerät für die Dauer der Aufgabe nicht ausfallen darf, werden genauere Anforderungen an das Anzeigegerät getroffen. Die Dauer der Aufgabe hängt vom jeweiligen Anwendungsfall ab.

#### 2.2.7.1 Erlaubte Anzeigetechnologien

Nach [MSC.191(79), 5.3.2] sollte das Anzeigegerät die Möglichkeit bieten, in einem Nachtmodus Informationen als hellen Vordergrund auf einem dunklen Hintergrund darzustellen. Umgekehrt ist ein „Tagesmodus“ (dunkle Informationen auf hellem Hintergrund) nicht explizit gefordert. Somit sind, neben handelsüblichen LCD-Anzeigen, beispielsweise Anzeigen auf LED-Basis (helle alphanumerische Informationen auf dunklem Hintergrund) auch nicht explizit untersagt, solange die Lesbarkeit der Anzeige unter allen Umgebungslichtbedingungen gewährleistet ist. Für die mobile Visualisierung kommt dies aber nicht in Frage, da gemäß der Anwendungsfälle eine Karte Bestandteil der Darstellung sein soll.

#### 2.2.7.2 Dimmen des Bildschirms und Energieversorgung

Die Norm fordert, dass der Bildschirm dimmbar ist und sich sowohl Helligkeit als auch Kontrast einstellen lassen. Die Einstellmöglichkeiten sollten so umfangreich sein, dass es möglich ist, den Bildschirm so einzustellen, dass er unter allen Umgebungslichtbedingungen leserlich bleibt. Es sollte für den Nutzer aber auch möglich sein, die Kontrast- und Helligkeitswerte auf einen vorgegebenen Standardwert zurückzusetzen [MSC.191, 8.1.2].

Generell sollte der Anzeigebildschirm beleuchtet sein, da anderweitig die Umsetzung des geforderten Nachtmodus nicht möglich ist. Die Möglichkeit des Zurücksetzens der Helligkeits- und Kontrastwerte auf voreingestellte Standardwerte ist insbesondere dann wichtig, wenn das Anzeigegerät das Herunterdimmen der Helligkeit auf 0 erlaubt. Starkes Dimmen der Anzeige kann sinnvoll sein, um den Stromverbrauch der Bildschirmanzeige zu minimieren und die Anzeigedauer zu maximieren, wenn das Anzeigegerät (vorübergehend) keine Netzverbindung hat.

Die Energieversorgung bei gleichzeitigem Dauerbetrieb der Anwendung könnte ein großes Problem darstellen, wenn die Anwendung mobil verwendet wird und ihre Energie aus einem wieder aufladbarem Speicher ohne Netzverbindung bezieht. In dem Fall ist die Lebensdauer des Speichers für die Anwendung kritisch. Hieraus ergibt sich eine Verfeinerung der Anforderung an die Ausfallsicherheit des Anzeigegerätes, dass die Energieversorgung des Anzeigegerätes so gegeben sein sollte, dass die Anzeige für die Zeitdauer der vorgesehenen Aufgabe ablesbar bleibt. Wird die Anzeige zum Zweck des Energieeinsparens gedimmt, sollte sie weiterhin unter allen Umgebungslichtbedingungen ablesbar sein. Einstellmöglichkeiten, wie das Dimmen des Bildschirms, können helfen, den Stromverbrauch zu vermindern.

# 3 ENTWURF

Es sollen Navigationsdaten in einer mobilen Anwendung für die zwei zuvor beschriebenen Anwendungsfälle (1.4) visualisiert werden. Die zu visualisierenden Daten stammen von der ebenfalls beschriebenen PNT-Unit, die Teil eines Echtzeitframeworks zur Prozessierung der Navigationsdaten ist.

Dieses Kapitel beschreibt den Entwurf der mobilen Visualisierung. Für den Entwurfsprozess wird ein objektorientierter „top-down"-Ansatz gewählt. Diagramme in Abbildungen sind, wenn nicht anders angegeben, mit der Software *Visio* in UML (Unified Modeling Language) entworfen. IDL steht für Interface Definition Language und wird im Folgenden in Diagrammen verwendet, wenn es zweckmäßig ist, Datentypen anzugeben – dies aber Programmiersprachen unabhängig erfolgen muss.[11]

## 3.1 RAHMENBEDINGUNGEN

Die (mobile) Anwendung ist nicht in das in 1.3.1 beschrieben Echtzeitframework eingebettet, sondern greift als separate eigenständige Anwendung auf den Datenausgabestrom der PNT-Unit zu. Abbildung 8 stellt schematisch dar, wie eine Anwendung auf der Anwendungsebene in das PNT-System (bestehend aus Sensor- und Prozessierungsebene) eingegliedert ist. Diese Anwendung wird die mobile Visualisierung sein.

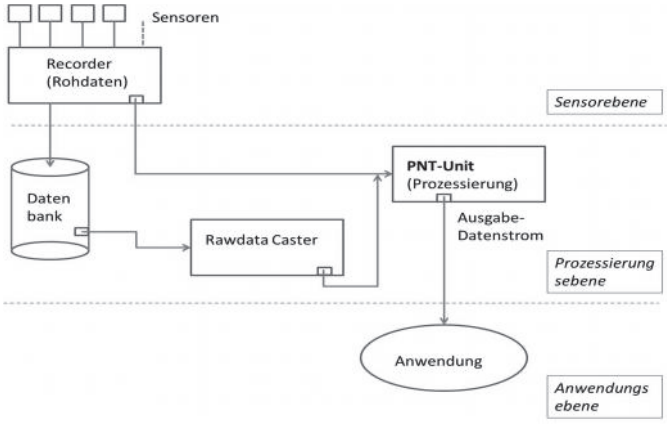


ABBILDUNG 8: DIE ANWENDUNG ALS TEIL DER ANWENDUNGSEBENE

[11] Beschreibung IDL, http://www.itwissen.info/definition/lexikon/interface-definition-language-IDL.html

### 3.1.1 Anwendungsfälle und Nachtmodus

Die Nutzung soll, entsprechend 1.4, sowohl stationär auf der Brücke als auch mobil als auch im Labor möglich sein. Als Brückengerät (und unter Umständen als Gerät für Demonstrationszwecke) wird demnach ein Ständer benötigt, der sich auf der Brückenarchitektur befestigen lässt ohne vorhandene Geräte zu beeinträchtigen oder das Brückenpersonal zu stören. Aus letzterem Grund benötigt die mobile Visualisierung zwingend einen Nachtmodus, wie es durch 2.2.6.2 vorgegeben ist. Da die Visualisierung zwei Anwendungsfällen genügen muss, wird sie von vornherein so entworfen, dass jede PNT-Information dargestellt werden kann, bzw. dass sich die Anwendung zukünftig unkompliziert um Komponenten erweitern lässt, die dies tun. Dies ermöglicht es auch, auf Erweiterungen an der Ausgabeschnittstelle zu reagieren. Um der Anforderung zu entsprechen, dass sich die Darstellung auf das Wesentliche reduzieren lassen können muss (2.1), wird es dem Nutzer möglich sein, Informationen auszublenden, die für den momentanen Anwendungsfall unwichtig sind.

### 3.1.2 Wahl des Anzeigegerätes und Schutzgrad

Die Entwicklung eines ersten Prototyps der Anwendung ist rein softwaretechnischer Natur. Ein konkretes Anzeigegerät wird nicht vorgegeben sondern richtet sich zum Implementierungszeitpunkt nach den Erfordernissen oder Möglichkeiten. Ein proprietäres Gerät ist erlaubt. Demnach muss auf die Vorgabe eines Schutzgrades, wie es in Kapitel 2.2.4 beschrieben ist, verzichtet werden, da der Fokus auf der Entwicklung eines Prototyps für die Visualisierung liegt und das Anzeigegerät bei Bedürfnissen nach mehr Schutz gegenüber Staub und Wasser angepasst werden kann. Nach 2.2.7.2 sollte das Anzeigegerät dimmbar und die Helligkeit und der Kontrast des Bildschirms einstellbar sowie auf einen Standardwert zurücksetzbar sein. Es kann offen bleiben, ob dies durch eine Hardware- oder Softwareschnittstelle oder einer Kombination von beidem erreicht wird.

## 3.2 Architektur

### 3.2.1 Schnittstellen

Dem top-down-Ansatz folgend, wird die zu entwerfende Anwendung hinsichtlich ihrer Schnittstellen betrachtet. Die mobile Visualisierung greift auf die Ausgabeschnittstelle der PNT-Unit zu, visualisiert die Daten und stellt die Informationen dem Nutzer dar. Zum Nutzer existieren zwei Schnittstellen (siehe Abbildung 9):

- Die visuelle, akustische (und gegebenenfalls haptische) Darstellung der Daten als Maschine-Mensch-Schnittstelle bzw. Ausgabeschnittstelle

- Hardware- und softwareseitige Kontrollelemente als Mensch-Maschine-Schnittstelle bzw. Konfigurationsschnittstelle, über die die Nutzerin oder der Nutzer einstellen kann, welche Informationen wie dargestellt werden sollen.

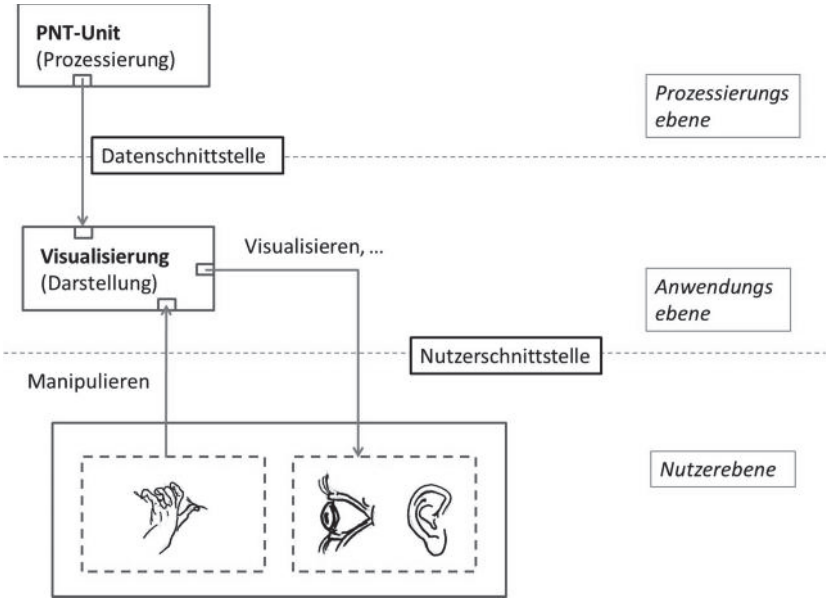


ABBILDUNG 9: SCHNITTSTELLEN DER MOBILEN VISUALISIERUNG

### 3.2.2 DATENSCHNITTSTELLE

Die PNT-Unit besitzt eine Softwarekomponente, die einen HTTP-Server implementiert, dessen Protokoll HTTP 1.0 (Hypertext Transfer Protocol) ist und über die die prozessierten Daten einer Anwendung im verbreiteten JSON-Datenformat bereitstellt werden. Der Datenaustausch zwischen der PNT-Unit und der Anwendung basiert auf der verbreiteten standard-offenen Intranet- beziehungsweise Internetkommunikation. Client und Server können somit auch auf demselben Computer laufen. Eine Client-Server-Interaktion überträgt immer eine Datenepoche.

Die Anwendung verbindet sich an einem TCP-Port mit der IP-Adresse des Servers. Die epochenweise Übertragung der Daten wird durch einen Cookie realisiert, der dem Client beim Verbindungsaufbau mitgesendet wird. Dies ermöglicht es dem Server, einen Antwortdatensatz bis zur nächsten Epoche zu verzögern, indem die HTTP-Anfrage erstmal offen gelassen wird, bis ein neuer Epochen-Datensatz bereit steht. Die Anwendung wird so mit der PNT-Unit synchronisiert. Durch das Kommunikations-bedingte Beenden der Verbindung nach jeder Epoche bzw. Übertragung eines Datensatzes, muss die Verbdingung aber neu aufgebaut werden.

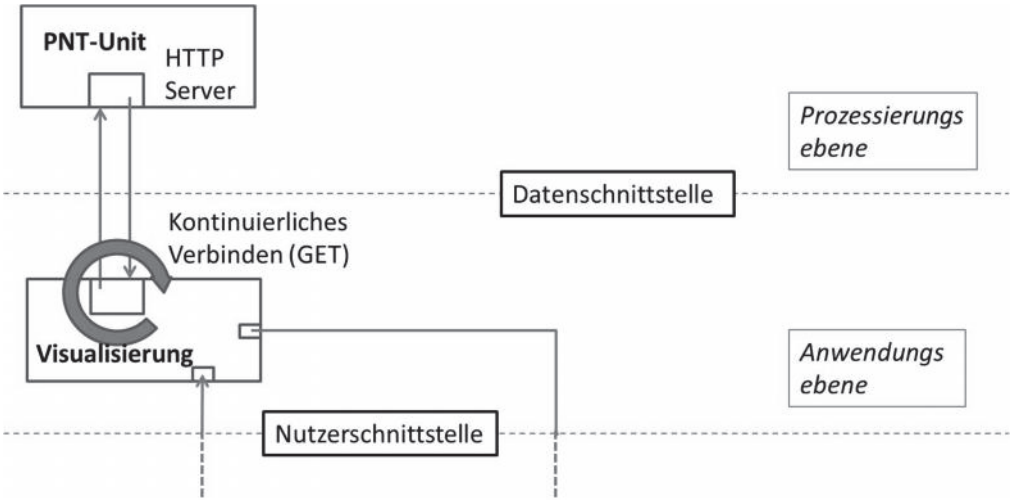


ABBILDUNG 10: DATENEMPFANG

Allgemein betrachtet existieren andere Möglichkeiten, die Kommunikation zwischen Datenquelle und Anwendung durchzuführen – beispielsweise eine direkte Verbindung über den vorhandenen Socket. Die Datenschnittstelle der PNT-Unit ist in der, in Abbildung 10 schematisch dargestellten, Form vorhanden, um AJAX-basierte Clientanwendungen (zum Beispiel Browseranwendungen) zu unterstützen (AJAX: Asynchronous JavaScript and XML).

Die Anwendung besitzt eine Komponente, die eine HTTP-Verbindung mit dem Server der PNT-Unit aufbaut, über die ein JSON-Objekt empfangen wird, welches in ein Objekt mit den PNT-Informationen transformiert wird, das anschließend an die visualisierende Komponente gesendet wird. (Abbildung 11)

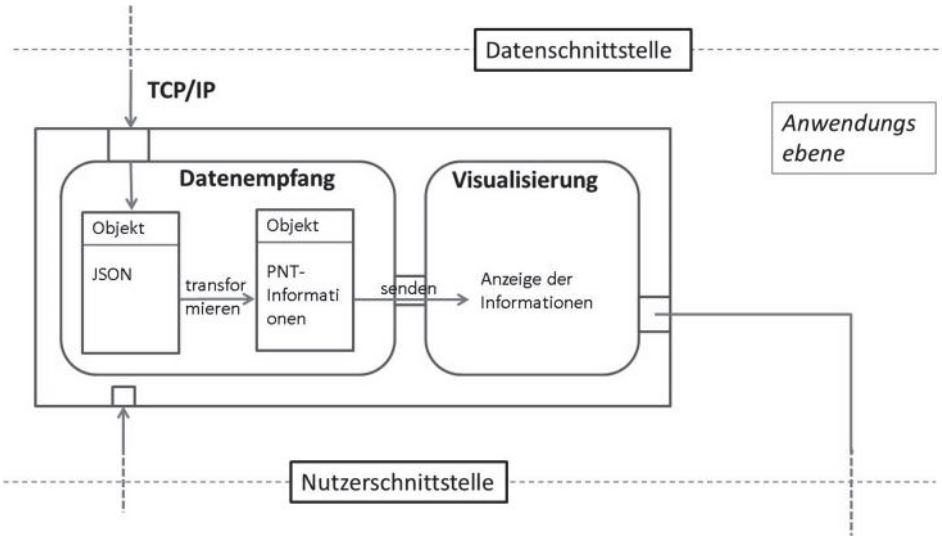


ABBILDUNG 11: DATENEMPFANG UND TRANSFORMATION

### 3.2.3 NUTZERSCHNITTSTELLE FÜR DIE AUSGABE VON INFORMATIONEN

Die Nutzerschnittstelle bezeichnet hardwareseitige Komponenten der Anwendung zum Übertragen von Bild- und Ton (und eventuell Gerätevibrationen) zum Zweck der Präsentation der Informationen für den Benutzer. Da ein proprietäres Anzeigegerät verwendet werden wird, wird an dieser Stelle auf eine konkrete Spezifikation der Nutzerschnittstelle verzichtet. Stattdessen sollen die Anforderungen an die Nutzerschnittstelle wie folgt eingegrenzt werden:

- Die Kommunikation muss vollständig über Gesten erfolgen können.
- Die Bildschirmtechnologie kann kapazitiv sein, da die Qualität der Darstellung und der Nutzerinteraktion (Multitouch-Gesten) für einen ersten Prototyp der Visualisierung höher bewertet wird, als die Robustheit des Gerätes gegenüber externen Einflüssen. Wasserbedingte Probleme mit der Bedienung werden somit vorerst in Kauf genommen.
- Ton- und Vibrationswiedergabe spielen (vorerst) keine Rolle.

Die Informationsanzeige der Nutzerschnittstelle orientiert sich in der Aufteilung des Bildschirms an ECDIS-Anzeigen, wie es in 2.2.5.1 beschrieben ist. Organisatorisch wird der Bildschirm demnach in einen Bereich für die Karte und einen für Nutzerdialoge aufgeteilt. Dies ist in Abbildung 12 dargestellt. Folgende Kategorien an Informationen aus Kapitel 2.2.5.1 treffen auf die Darstellung der PNT-Informationen zu:

- Eigenschiffinformationen – Karte und Nutzerdialog
- Navigationshilfen und –anzeigen – Karte und Nutzerdialog
- Informationen aus einer Kartendatenbank – nur Karte
- Systemeinstellungen – nur Nutzerdialog und werden über eine Konfigurationsschnittstelle erfolgen.

Zusätzlich zu diesen, rein informativen Zwecken dienenden, Bereichen beinhaltet die Anzeige verschiedene dauerhaft verfügbare Kontrollfunktionen und Informationen, die sich in einer Titel- und einer Menüleiste befinden. Dies bezieht sich auf die Anforderung (2.2.5.1), dass wichtige und vorrangige Informationen permanent sichtbar dargestellt werden sollten. Hier durch das Mittel der Positionierung.

Bei der Informationspräsentation müssen die Anforderungen bezüglich der Darstellung von Schrift und Sprache (2.2.6.1), Farben und Intensität (2.2.6.2), Icons und Symbole (2.2.6.3) sowie Begriffen und Abkürzungen (2.2.6.5) berücksichtigt werden. Per Spezifikation der Ausgabe (1.3.3) wird mit dem RAIM-Wert ein Integritätswert mit den PNT-Daten bereitgestellt. Es wird festgelegt, dass die RAIM-Alarmwerte wie folgt farblich kodiert werden:

- keine Daten oder Dienst nicht vorhanden – Weiß
- Position unsicher – Rot (Alarm)
- Position sicher – Grün (Normalzustand)

Entsprechend für die Qualität der Lösung bei Prozessierung mit phasenbasiertem DGNSS:

- keine Daten – Weiß
- Fix - Grün
- Float - Rot

Letzteres wird nur angezeigt, wenn auch wirklich eine PDGNSS-prozessierte Position visualisiert wird (in Form der eigenen Schiffsposition oder als dargestellte Bahn, vgl. 3.3.5).


ABBILDUNG 12: AUFTEILUNG DES BILDSCHIRMS

Tabelle 2 zeigt, welche PNT-Information in welchem Bereich der Anzeige benötigt wird. Die Menüleiste stellt keine Informationen dar.

| **PNT-Information** | **Bereich: Karte** | **Bereich: Nutzerdialoge** | **Bereich: Titelleiste** |
|---|---|---|---|
| ID | nein | nein | nein |
| Zeitstempel | nein | ja | ja |
| Koordinaten | ja, für Position des Schiffes | ja | ja |
| Roll-, Nick- und Gierwinkel | ja, für Ausrichtung des Schiffes | ja | nein |
| Geschwindigkeitsvektorkomp. | ja | ja | nein |
| Kurs über Grund | ja, für Ausrichtung des Schiffes | ja | nein |
| RAIM Grenzwert | nein | ja | nein |
| RAIM Alarmwert | nein | ja | nein |

| | | | |
|---|---|---|---|
| Qualität der Positionslösung | nein | ja | ja, bei PDGNSS Prozessierung |
| Satellitenmatrix | nein | ja | nein |

TABELLE 2: VERTEILUNG DER INFORMATIONEN AUF DIE BEREICHE DER ANZEIGE

### 3.2.4 KONFIGURATIONSSCHNITTSTELLE

Die Konfigurationsschnittstelle erlaubt es dem Nutzer, Einfluss zu nehmen auf Art und Umfang der Darstellung der Informationen. Dabei sollen die Nutzereinstellungen persistent gespeichert und bei jedem Programmstart wieder geladen werden. Wie in Abbildung 13 schematisch dargestellt, geschieht die Anzeige der Informationen auf Basis der hinterlegten Einstellungen.

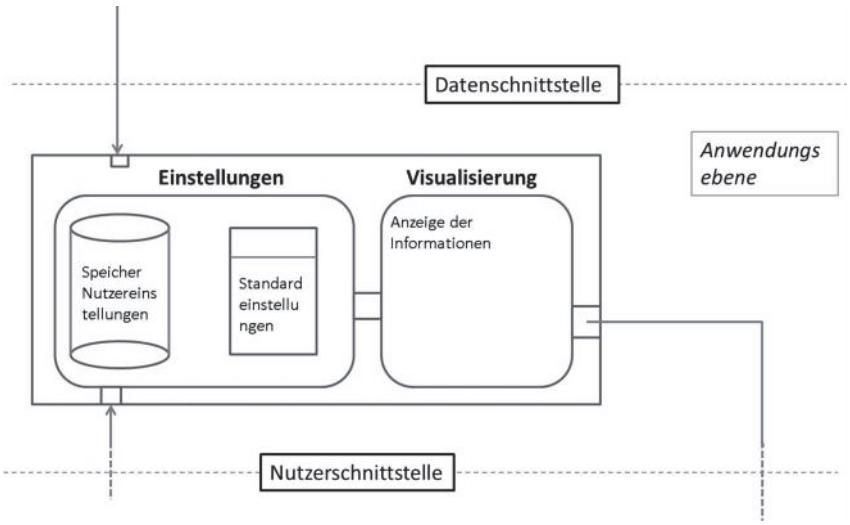


ABBILDUNG 13: EINSTELLUNGEN

Es existieren zwei Szenerien, für die die Anwendung Standardeinstellungen benötigt. Zum einen, damit der Nutzer, gemäß 2.2.5.3, die Einstellungen jederzeit auf Standardwerte zurücksetzen kann. Zum anderen, damit der Speicher der Nutzereinstellungen beim ersten Programmstart mit den Standardeinstellungen initialisiert werden kann (Abbildung 14).

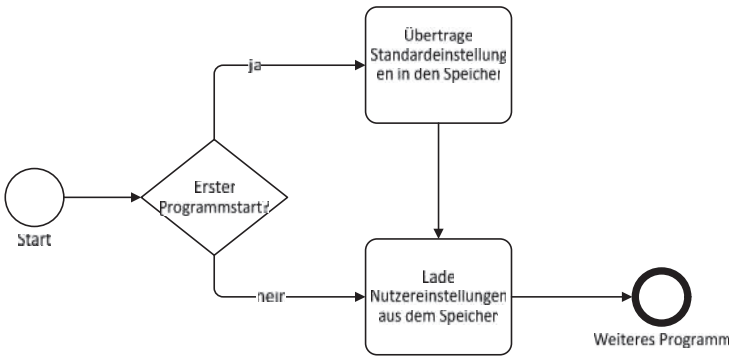


ABBILDUNG 14: LADEN DER NUTZEREINSTELLUNGEN BEI PROGRAMMSTART

### 3.2.5 KOMPONENTEN

Komponenten können, unter Beachtung der Schnittstellen, ohne weitere Änderungen am System ausgetauscht werden. Das Komponentendiagramm in Abbildung 15 stellt den Gesamtaufbau des Systems dar. Es existieren drei Komponenten: für Datenempfang und Aufbereitung, für die Darstellung von Informationen und Kontrollen und für die Verwaltung der Einstellungen.

Die Datenkomponente stellt über eine Schnittstelle ein PNT-Objekt bereit, das aus dem über eine TCP/IP-Verbindung übertragenen JSON-Objekt erzeugt wird. Die Art der Schnittstelle, über die das PNT-Objekt den darstellenden Komponenten bekannt gemacht wird, ist noch zu klären.

Die Titel-, Karte- und Dialogkomponenten der Anwendung stellen die jeweils benötigten Informationen aus dem PNT-Objekt dar. Über die Kontrollen des Menüs lassen sich der Inhalt und die Art der Darstellung der Karte und der Dialogkomponente verändern. Die Titelleiste ist entsprechend 3.2.3 permanent sichtbar. Das Menü erlaubt es auch, die Nutzereinstellungen zu ändern und persistent zu speichern. Die Informationsdarstellung berücksichtigt Nutzereinstellungen die Art und Inhalt der darzustellenden Informationen festlegen.

Die Einstellungsverwaltung beinhaltet eine Datei mit Standardeinstellungen, die beim ersten Programmstart geladen wird und auf die, entsprechend der Anforderungen, jederzeit zurückgewechselt werden kann sowie eine Möglichkeit zum persistenten Speichern der momentanen Nutzermöglichkeiten.

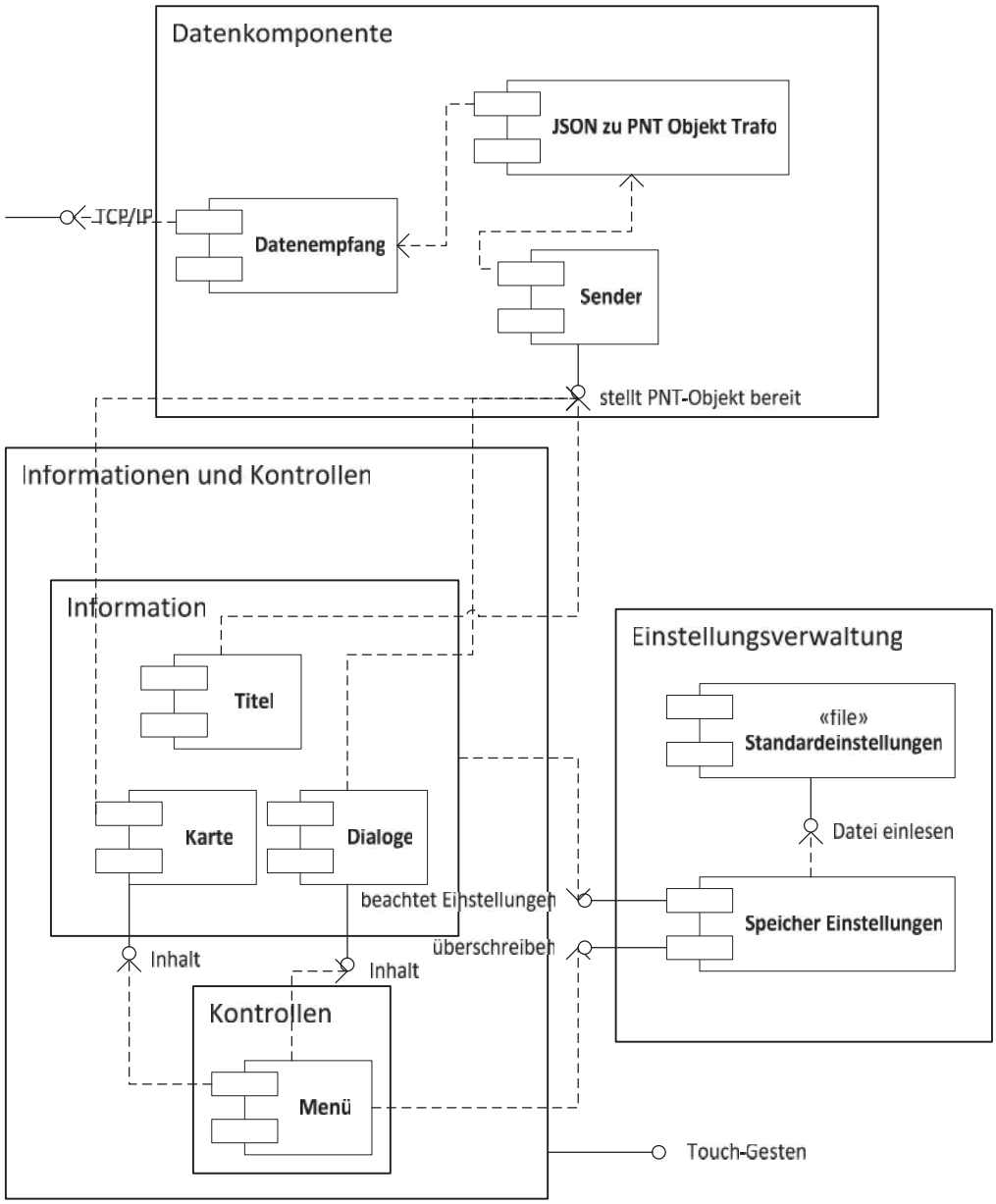


ABBILDUNG 15: KOMPONENTENDIAGRAMM DES GESAMTSYSTEMS

### 3.2.6 ACHITEKTURMODELL

Aus der Aufteilung der Komponenten zur Darstellung eines inhaltlich veränderlichen aber in seiner Struktur konstanten Datensatzes an PNT-Informationen, bietet sich strukturell ein Model-View-Controller-Muster (MVC) an, wie es in Abbildung 16 dargestellt ist. Das Modell (Model) sind die PNT-Informationen, die in der Präsentation (View) visualisiert werden soll. Jede Komponente der Präsentation liest andere Einzelinformationen aus dem Modell und ändert ihren Inhalt bei Änderungen am Modell. Die Präsentation nimmt durch entsprechende Kontrollelemente auch Nutzereingaben entgegen, die sie an die Steuerung (Controller) zur Verarbeitung weiterleitet. Die Steuerung aktualisiert sowohl das Modell, als auch die Präsentation und verteilt Objekte des Modells auf die Komponenten der Präsentation.

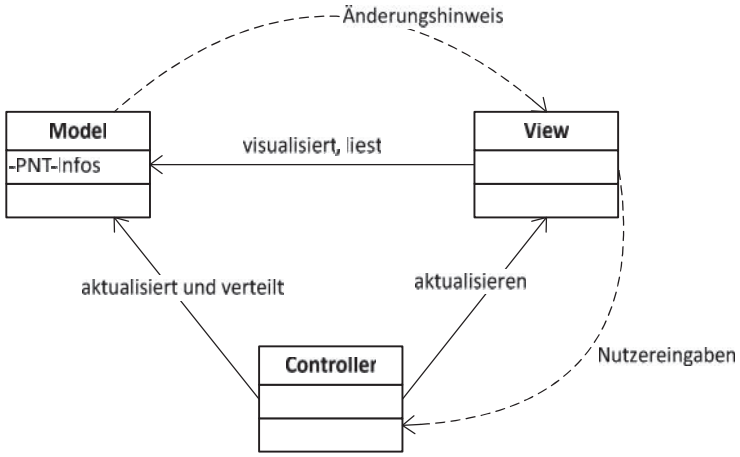


ABBILDUNG 16: MPDEL VIEW CONTROLLER MUSTER

## 3.3 KLASSEN

### 3.3.1 KLASSE PNT-INFORMATIONEN

Das Datenobjekt beinhaltet die PNT-Daten, die in 1.3.3 beschrieben wurden. Die entsprechende Klasse *PNT-Informationen* sieht wie in Abbildung 17 dargestellt aus. Der Konstruktor der Klasse erwartet idealerweise ein *JSON*-Objekt als Parameter. Es gibt keine Setter für die Attribute, da diese nicht benötigt werden. Die Klasse PNT-Informationen entspricht dem Modell des MVC-Musters.

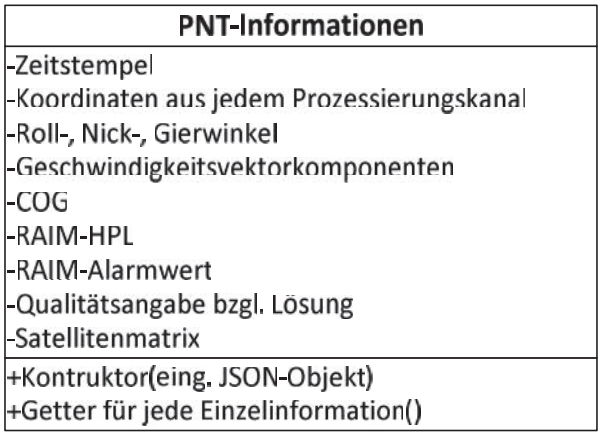


ABBILDUNG 17: KLASSE PNT-INFORMATIONEN

### 3.3.2 KOMPLETTES KLASSENDIAGRAMM

Abbildung 18 zeigt das komplette Klassendiagramm. Die Komponenten, die den Bestandteilen des MVC-Musters zugeordnet werden können sind farblich markiert. Eine Legende befindet sich in der unteren rechten Ecke der Abbildung. Die Klassen, die den Komponenten aus Abbildung 15 (Datenkomponente, Informationen und Kontrollen und Einstellungsverwaltung) entsprechen, sind zur besseren Vergleichbarkeit umrandet. Die dort beschriebene Datenkomponente ist zu einer Klasse vereinfacht, die sich mit der PNT-Unit verbinden und davon trennen kann und mit einem JSON-Objekt ein Objekt der Klasse PNT-Informationen erstellen kann.

Die Klasse *PNT-Informationen* ist als Assoziierungsklasse zwischen den Klassen für die Oberfläche als Gesamtheit des dargestellten Bildschirms (ohne die nautischen Informationen)

und der Datenempfangsklasse, durch die das *PNT-Informationen*-Objekt erstellt wird, modelliert. Die PNT-Informationen entsprechen dem Modell des MVC-Musters. Die Oberfläche übernimmt in dieser Abstraktion die Steuerung des MVC-Musters. Sie realisiert eine noch zu definierende Nutzerschnittstelle zu Verarbeitung von Nutzerbefehlen. Eine klare Trennung der Steuerung von Präsentation und Modell lässt sich an dieser Stelle nicht modellieren, da die Steuerung von der Nutzerschnittstelle abhängt und diese von dem verwendeten mobilen Gerät. Eine Festlegung ist erst mit der Implementierung möglich.

Bei jeder neuen PNT-Informations-Nachricht aktualisiert die Nutzeroberfläche den Inhalt der erbenden Klassen. Dies sind die Komponenten, die in 3.2.5 definiert wurden: jeweils eine Klasse für die Kartendarstellung, Dialogdarstellung, Menüleiste und Titelleiste. Karten- und Dialogdarstellung erben von der allgemeinen abstrakten Darstellungsklasse *PNT-Informationen Darstellung*. Dies entspricht der Realisierung der Präsentationsschicht des MVC-Musters.

Die Menüleiste hat Zugriff auf die Methoden der Einstellungen und des Datenempfängers. Von den Einstellungen sind Karten- und Dialogdarstellung sowie der Datenempfang abhängig.

Die Kartendarstellung realisiert eine Kartendatenbasis und importiert *Overlay*-Klassen zur Darstellung der verschiedenen Informationen. Die Kartendarstellung wird auf nicht offiziellen Kartendaten beruhen. Eine Entscheidung diesbezüglich wird zum Zeitpunkt der Implementierung getroffen. Die Darstellung von Zusatzinformationen, wie dem eigenen Schiff, wird sich hinsichtlich verwendeter Symbole an 2.2.6.4 orientieren. Dies wird in 3.3.5 näher betrachtet.

Die Dialogdarstellung importiert verschiedene Klassen zur Informationsdarstellung auf der Dialogebene. Die Dialogdarstellung wird auch für die Darstellung des Menüs, mit dem der Nutzer die Zielkoordinaten entsprechend des Anwendungsfalls Baltic Taucher II eintragen kann, genutzt.

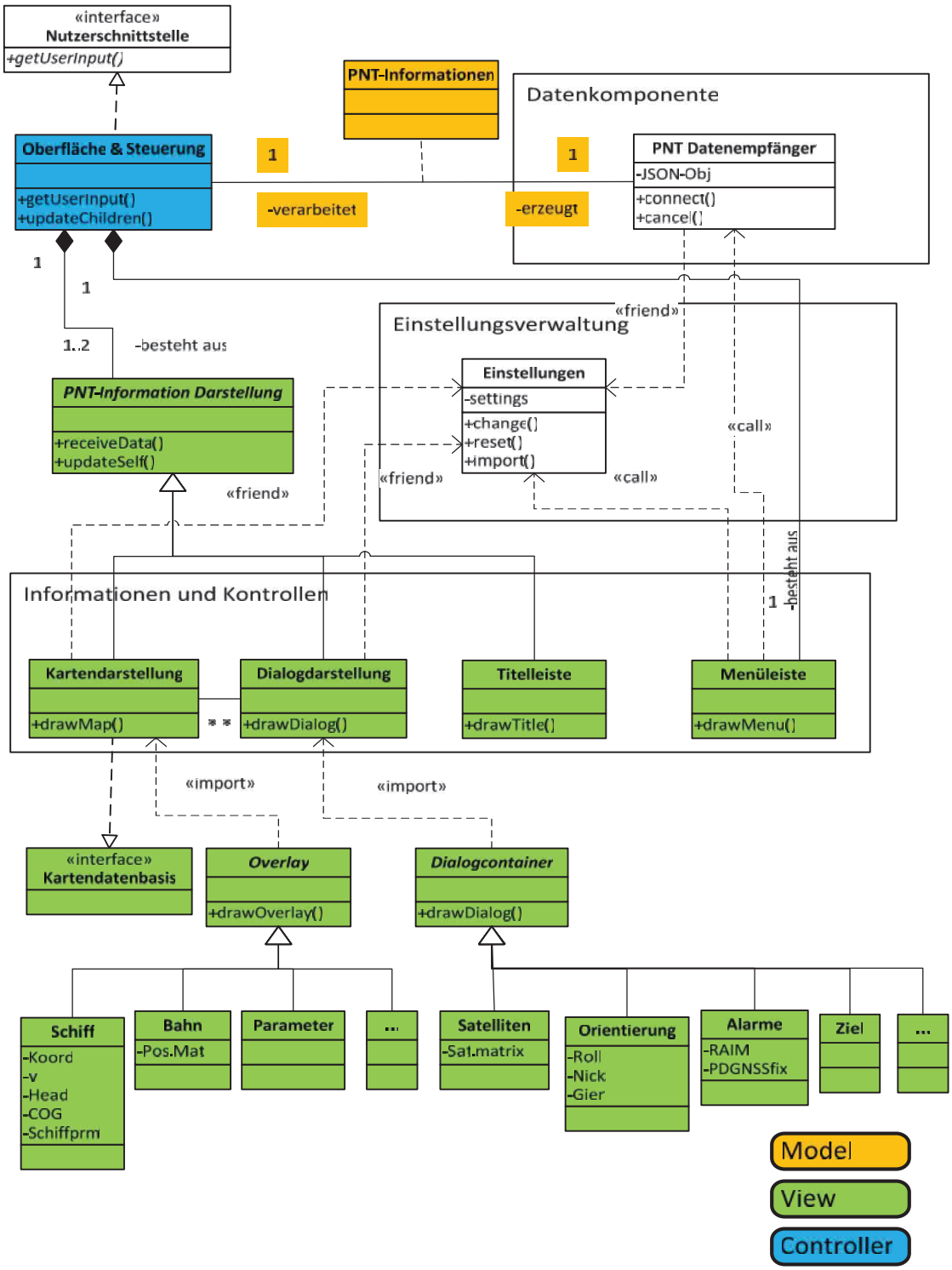


ABBILDUNG 18: KOMPLETTES KLASSENDIAGRAMM

### 3.3.3 DIE KLASSE EINSTELLUNGEN

Abbildung 19 stellt die Klasse dar, mit der die Einstellungen der Anwendung verwaltet werden. Die Klasse realisiert eine Schnittstelle, mit der es möglich ist, Standardeinstellungen (die beispielsweise als Datei hinterlegt sind) zu importieren.

Die Attribute der Klasse stehen für Kategorien, die mehrere einstellbare Parameter umfassen. Schiffsparameter beinhaltet den Schiffsnamen, die Schiffsnummer, die Rufnummer, die Schiffslänge und -breite. Die Prozessierungseinstellungen legen fest, welche Positionslösung angezeigt und auf welche das Schiff positioniert werden soll. Die Verbindungseinstellungen beinhalten die IP und den Port des Servers der PNT-Unit von dem die PNT-Daten bezogen werden. Die Karteneinstellungen enthalten die Quelle der Kartendatenbasis.

Die Methoden der Klasse erlauben das Verändern und das Zurücksetzen der Einstellungen sowie das Importieren eigener Einstellungen über dieselbe Schnittstelle über die die Standardeinstellungen geladen werden.

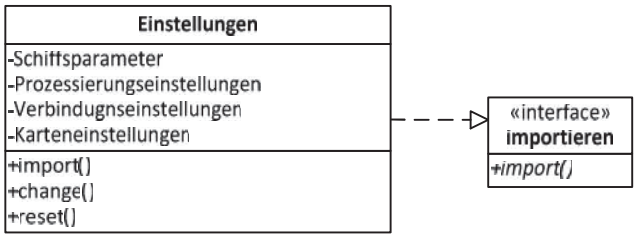


ABBILDUNG 19: KLASSE EINSTELLUNGEN

### 3.3.4 DATENEMPFÄNGER

Der PNT-Datenempfänger realisiert zwei Schnittstellen: einen HTTP-Client für die HTTP-Verbindung zum Beziehen der PNT-Daten im JSON-Format und eine Schnittstelle zum Aktualisieren des Inhalts aller momentan vorhanden Komponenten der Nutzeroberfläche (*UpdateChildren()*). Der HTTP-Client stellt drei Methoden zur Verfügung: zum Verbinden zu einer Zieladresse (*connect(url)*), der als Parameter URL (Uniform Resource Locator) übergeben wird und zum Trennen (*abort()*) der HTTP-Verbindung sowie zum Beziehen der PNT-Daten (*httpGet()*). Die PNT-Daten werden als JSON-Objekt empfangen, mit dem ein Objekt der Klasse *PNT-Informationen* instanziiert wird. Mit diesem Objekt werden die Inhalte der beteiligten Komponenten der Nutzeroberfläche aktualisiert. (Abbildung 20)

Die Modularisierung ist so gewählt, um auf Änderungen an der Datenquelle - dem objektorientierten Ansatz entsprechend - durch Austausch des jeweiligen Moduls begegnen zu

können: auf Änderung des Übertragungsformats durch Hinzufügen eines neues Konstruktors und auf Änderung des Übertragungsprotokolls durch Austausch des Clients.

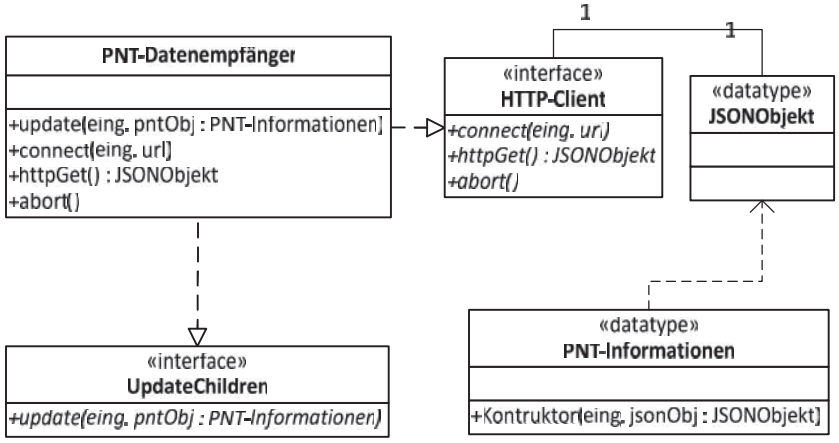


ABBILDUNG 20: DATENEMPFÄNGER KLASSENDIAGRAMM

### 3.3.5 *OVERLAYS*

Die Kartendarstellung soll getrennt sein von allen Navigationsinformationen, die auf der Karte dargestellt werden. Dafür soll eine Klasse *Overlay*, bzw. davon abgeleitete Klassen, bereitgestellt werden, die spezifische Informationen über der Karte darstellt. Dies ist in Abbildung 21 dargestellt. Es wurden drei Informationskategorien identifiziert, die auf der Karte dargestellt und für die drei Unterklassen von *Overlay* bereitgestellt werden sollen: dies sind Informationen, die das Schiff betreffen, die Darstellung der zurückgelegten Bahn und Parameter wie der Kurs.

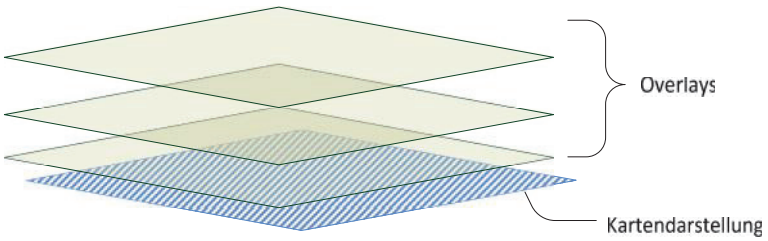


ABBILDUNG 21: OVERLAYS ÜBER DER KARTE

Die *Overlay*-Klasse beinhaltet eine Malmethode (*drawSelf()*), die als Übergabeparameter erwartet, was wo gezeichnet werden soll. Das „wo“ ist ein Objekt der Kartendarstellung, das

„was“ ist abhängig von der realisierten Unterklasse von *Overlay*. Welche konkreten Informationen in welchem *Overlay* benötigt werden, ist in Abbildung 22 dargestellt. Jedes *Overlay* hat eine *add()*-Methode, mit der epochenweise PNT-Informationen als Parameter an die Klasse zur Darstellung übergeben werden können.

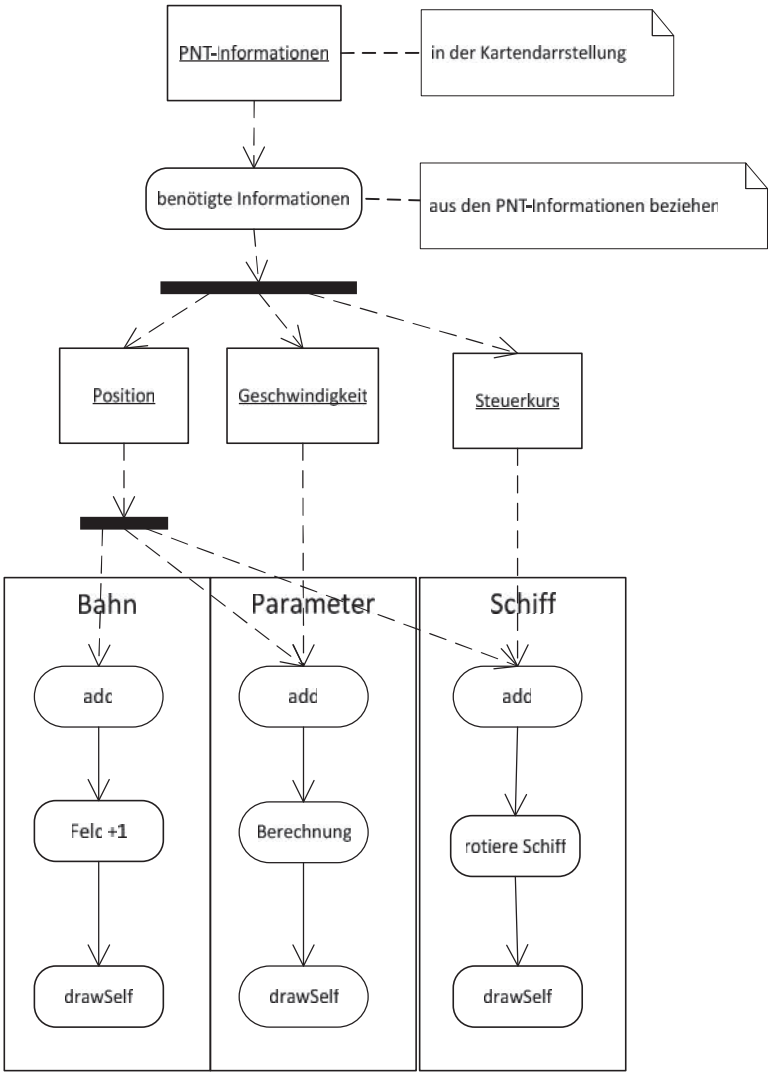


ABBILDUNG 22: VERTEILUNG DER INFORMATIONEN AUF DIE OVERLAYS

Abbildung 23 stellt die *Overlay*-Klassen und ihre Abhängigkeiten dar. Die Klasse *Overlay* steht in Abhängigkeit zu einer abstrakten Klasse „Malbares“. Diese Klasse soll an dieser Stelle nicht weiter modelliert werden. „Malbar“ ist generisch betrachtet alles, was in einer *Overlay*-Klasse grafisch erzeugt werden kann. Dies ist je nach Unterklasse und Kontext etwas anderes, worauf im Folgenden noch eingegangen wird.

Die Klasse **Overlay Bahn** stellt die prozessierten Positionen des Schiffes (den zurückgelegten Weg) dar. Sie steht in Abhängigkeit zu den Einstellungen, da dort hinterlegt ist, welche Prozessierungsergebnisse dargestellt werden sollen. Das „Malbares"-Objekt der Klasse ist die befahrene Bahn. Die Bahnwerte (Koordinaten) werden in einem Feld gespeichert. Die *add()*-Methode der Klasse erwartet als Parameter ein Koordinatenobjekt, bzw. ein Feld von Koordinaten.

Die Klasse **Overlay Parameter** stellt Schiffsparameter dar. Ein Parameter kann jede physikalische Größe sein, die die Schiffsposition beschreibt. Als beispielhafter Parameter sei hier die Visualisierung der Geschwindigkeit als Vektorpfeil modelliert. Dieser setzt, entsprechend 2.2.6.4, am CCRP des Schiffes an. Die Vektorkomponenten in Ost- und Nordrichtung befinden sich in den PNT-Informationen, der Ursprung des Vektors ist die prozessierte Schiffsposition. Diese drei Informationen bilden die Parameter der *add()*-Methode. Intern lässt sich nun der Vektor der Geschwindigkeit berechnen und als Pfeil darstellen. Die Darstellung ist wieder ein Objekt von „Malbares".

Die Klasse **Overlay Schiff** stellt das zu visualisierende Schiff dar. Die *add()*-Methode der Klasse bekommt als Parameter die Position und den Steuerkurs, womit sich das Schiff positionieren und ausrichten lässt; außerdem die Geschwindigkeit, um diese in einem Popup-Fenster darstellen zu können. Bei jedem Aufruf von *add()* wird das Schiff neu gezeichnet – entsprechend wird eine *remove()*-Methode angeboten, um jede alte Darstellung zu entfernen. Die *rotate()*-Methode erlaubt es, das Schiffssymbol zu rotieren. So muss es nur einmal gezeichnet werden und kann in jeder folgenden Zeichnung in den jeweiligen Steuerkurs rotiert werden. Das Schiffssymbol kann, entsprechend 2.2.6.4 ein vereinfachtes Symbol sein oder der maßstabsgetreue Umriss. Beides sind Objekte der Klasse „Malbares" oder einer Unterklasse davon. Zur Entscheidung, welches Symbol zu verwenden ist, gibt es eine boolesche Variable „*drawAsTrueScale*", die entsprechend eines noch zu treffenden Kriteriums wahr oder falsch ist. Die Klasse Overlay Schiff steht in Abhängigkeit zu den Einstellungen. Von dort bezieht sie Schiffsparameter (Name, Schiffsnummer, etc.), die zusammen mit der übertragenen Geschwindigkeit in einem Popup-Fenster (*showPopup()*-Methode) auf der Karte dargestellt werden können. Das Popup-Fenster soll, in Anlehnung an ECDIS momentane und dauerhafte Schiffsparameter darstellen und durch eine Geste (analog einem Mausklick) auf das Schiffssymbol ein- oder ausgeblendet werden können. Dafür realisiert die Klasse Schiff Overlay eine (Ereignis-)Methode der Nutzerschnittstelle, die an dieser Stelle *onTap()* heißen soll.

Jede Abhängigkeit einer Klasse von den Einstellungen bedeutet auch, dass eine Änderung der darstellungsbezogenen Einstellungen, eine Änderung der schiffs- oder bahnbezogenen Kartendarstellung bewirkt. Das heißt, wenn beispielsweise, der Nutzer den Namen des Schiffes in den Einstellungen ändert, ändert sich zeitgleich der Name des Schiffes in den entsprechenden Darstellungen.

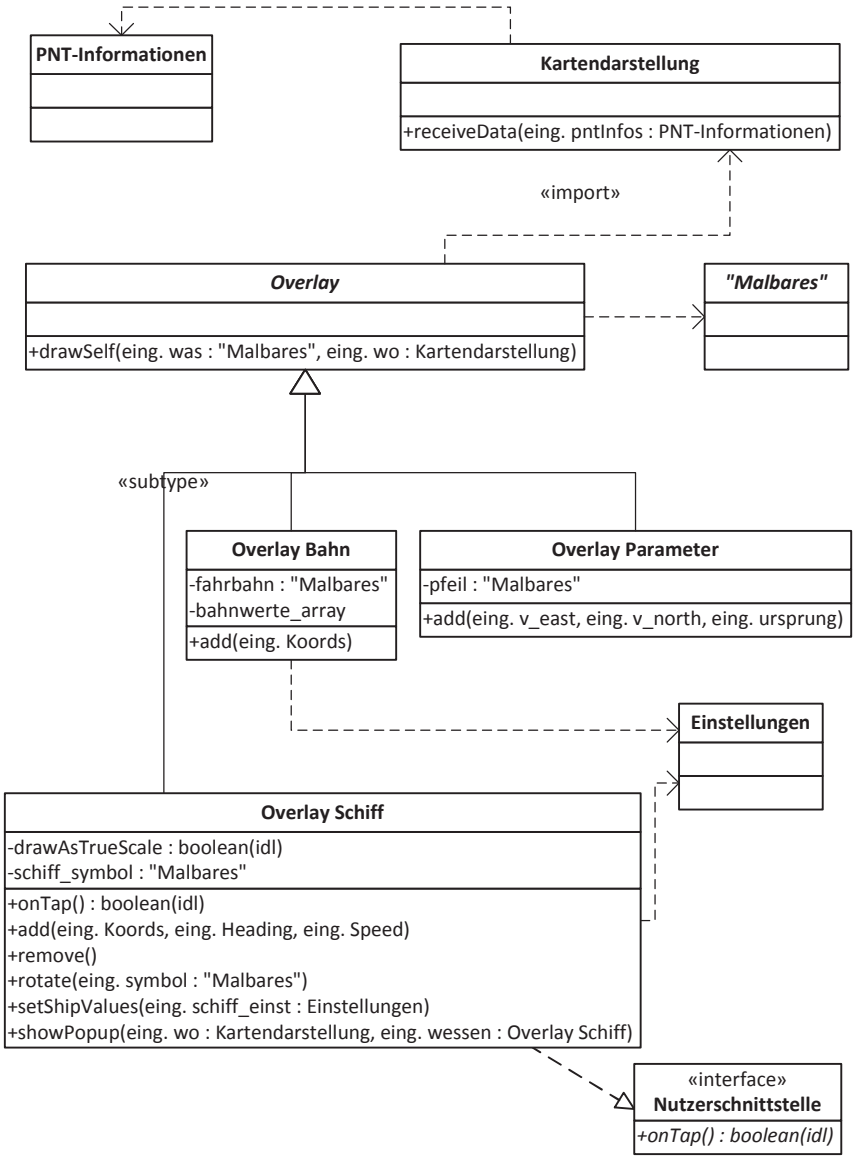


ABBILDUNG 23: KLASSENDIAGRAMM OVERLAY

### 3.3.6 MENÜLEISTE UND TITELLEISTE

Die Menüleiste hat Zugriff auf Funktionen des Datenempfängers und der Einstellungen. Sie bietet darum selbst Menüelemente, mit denen sich die Anwendung mit dem Datenserver verbinden und trennen lässt sowie die Einstellungen aufrufen oder in irgendeiner Weise manipulieren lassen.

Die Titelleiste stellt wichtige Informationen dar, die permanent sichtbar sein müssen. Dazu zählen die momentane Position des Schiffes und die Zeit. Die epochenweise Aktualisierung

der Zeit setzt gleichzeitig den in 2.2.6.8 erwähnten Vorschlag um, ein Element der Benutzeroberfläche als Indikator für eine bestehende Verbindung auszuweisen.

### 3.3.7 DIALOGE

Gemäß 2.2.5.1 stellt der Dialogbereich der Anwendung Daten und Menüs zur Nutzerinteraktion dar. Es sind auch mehrere Einzeldialoge möglich und die Darstellung der Informationen kann sowohl grafisch als auch in alphanumerischer Form sein.

Das oberste Element ist ein **Dialogcontainer**. Dieser entspricht dem Nutzerdialog aus Abbildung 12. Der Dialogcontainer hat, wie in Abbildung 18 dargestellt, Zugriff auf ein Objekt mit PNT-Informationen (da er von der Klasse „PNT-Informationen Darstellung“ erbt) und kann aus mehreren Einzeldialogen bestehen oder leer sein. Jeder einzelne Dialog realisiert die Nutzerschnittstelle und verfügt damit über Methoden zur Verarbeitung von Gesten. Ein Dialog setzt sich zusammen aus einer belieben Anzahl an Nutzermenüs sowie aus höchstens einem Objekt zur grafischen Darstellung der Informationen. *Nutzermenüs* sei eine hier nicht weiter ausgeführte Klasse, die Kontrollelemente für die Nutzerinteraktion bereitstellt. Dialoge lassen sich ausblenden (*hide()*) und wieder einblenden (*show()*).

Die Klasse ***KonkreteMalerei*** dient der grafischen Darstellung der PNT-Informationen. Sie erbt von der abstrakten parametrisierten Klasse ***Malerei***. Die Idee hierbei ist, dass für jede einzelne konkrete PNT-Information ein Dialog bereitgestellt wird, der diese Information visualisiert. Die Konkrete PNT-Information ist eine Einzelinformation aus dem PNT-Informationen-Objekt. Ein Objekt mit einer Einzelinformation kann von jedem Datentyp sein, weshalb die Klasse zum Zwecke der Vereinfachung als abstrakte Klasse modelliert ist. Eine konkrete PNT-Information kann beispielsweise die Geschwindigkeit als Gleitkommazahl oder die Satelliten als Matrix sein. In dem Beispiel hätte der Dialogcontainer zwei Dialoge zur Darstellung dieser Informationen. Im Fall der Satelliten-Matrix könnten die Satelliten in einem Skyplot grafisch visualisiert werden. Die Klasse *KonkreteMalerei* steht somit für eine unendlich große Menge an Klassen, die der Visualisierung von einer konkreten PNT-Information als Grafik in einem Dialog dienen.

Die Klasse *Malerei* bietet dafür vier notwendige Methoden. *Update()* aktualisiert die Grafik epochenweise mit einer neuen PNT-Information. *NotifyForRedraw()* erzwingt ein erneutes grafisches Darstellen des KonkreteMalerei-Objekts und *GetDimensions()* gibt Breite und Höhe der Grafik zurück. Beide Methoden werden benötigt, für den Fall, dass beispielsweise ein anderer Dialog ausgeblendet wird und die verbleibenden Dialoge um den freigewordenen Bereich des Dialogcontainers neu skaliert werden müssen. Dies lässt sich gut am Beispiel des Satelliten-Skyplots erklären, dessen kreisrunde Grafik immer vollständig in den umrandenden Dialog passen muss. *DrawSelf()* ist die Malmethode zur Visualisierung aller vorhandenen Informationen.

Das Klassendiagramm für die Dialoge ist mit den jeweiligen Abhängigkeiten in Abbildung 24 dargestellt.

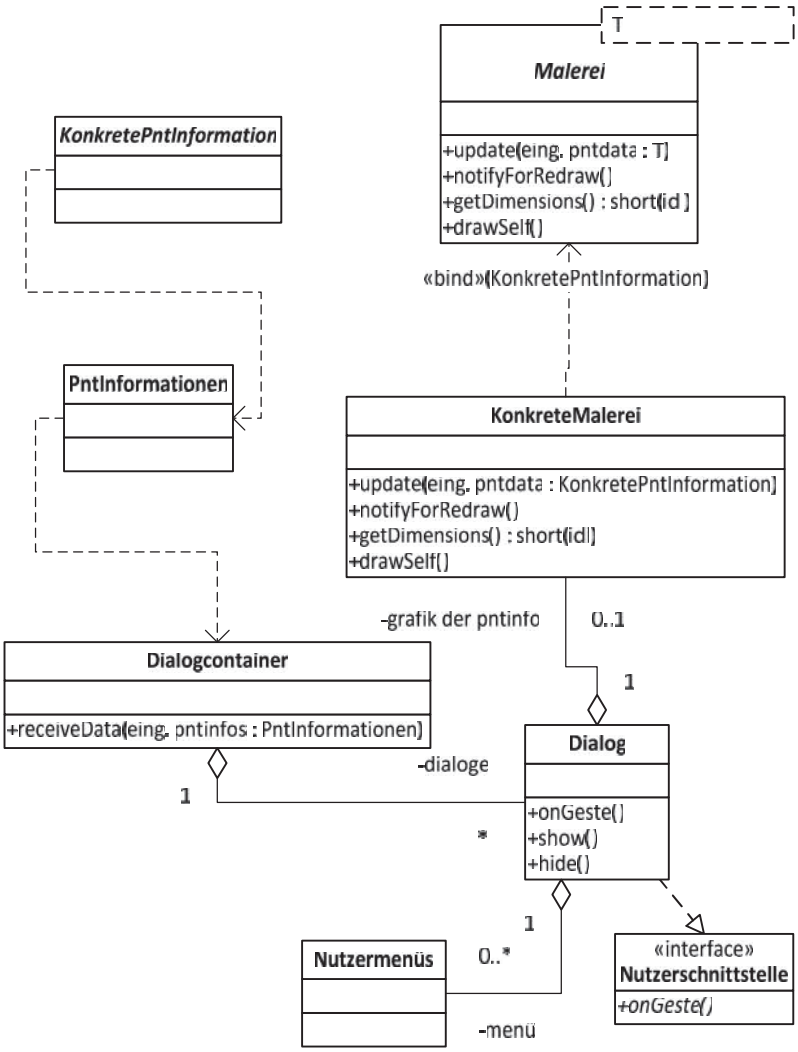


ABBILDUNG 24: KLASSENDIAGRAMM DIALOGE

## 3.4 KOMMUNIKATION

### *3.4.1 DATENEMPFANG*

Aus dem allgemeinen Klassendiagramm wird deutlich, dass jede Komponente der Nutzerschnittstelle andere Informationen des PNT-Datenobjekts benötigt. Im Sinne des objektorientierten Ansatzes werden die jeweils benötigten Informationen nicht auf die einzelnen Komponenten verteilt, sondern jede Komponente erhält Zugriff auf das PNT-Objekt und arbeitet dann mit den Informationen, die sie zur Darstellung benötigt. Dies folgt dem Beobachter-Muster, wonach die Komponenten als Beobachter das PNT-Objekt und seine Schnittstellen kennen, aber nicht anders herum. Die Komponenten selbst sind der Steuerung

bekannt. So wird auch die Kapselung der Komponenten gewährleistet und es können zukünftig weitere Komponenten, die von der abstrakten Klasse *PNT-Informationen Darstellung* abstammen, zur Nutzeroberfläche hinzugefügt werden.

Der Verbindungsaufbau ist vereinfacht dargestellt. Die interne Anfrage-Antwort-Kommunikation zwischen der HTTP-Schnittstelle des Datenempfängers und der Ausgabeschnittstelle der PNT-Unit sollen hier nicht modelliert werden. Die Menüleiste bietet Elemente, mit denen die Nutzerin oder der Nutzer als ein Akteur die Methoden des Datenempfängers (Verbindung aufbauen und trennen) auslösen kann. Dies wird auch nicht modelliert. Die Sicht auf die Kommunikation ist intern.

Bevor der Datenempfänger sich mit der PNT-Unit verbinden kann, benötigt er eine URL (bestehend aus Protokoll, IP, TCP-Port und Pfad zur JSON-Ressource), über die er Zugriff auf die Ressource mit den PNT-Daten bekommt. Diese Information wird vor dem Verbindungsversuch von den Einstellungen, in denen die Information hinterlegt ist, abgefragt.

Auf der URL erfolgt der Verbindungsaufbau in einer Schleife solange bis eine Verbindung besteht. Es besteht die Möglichkeit, dass ein Verbindungsaufbaus aus verschiedenen Gründen nicht möglich ist (falsche URL, PNT-Unit nicht eingeschaltet, Verbindungsprobleme, etc.). Da es die Absicht der Nutzerin oder des Nutzers ist, eine Verbindung herzustellen, probiert der Datenempfänger dies bis eine Verbindung besteht oder der Versuch beendet wird (um beispielsweise den Verbindungsfehler zu beheben). Während des Verbindungsversuchs wird auf der Oberfläche eine Information ausgegeben, dass die Anwendung versucht, sich zu verbinden. Die Information wird ausgeblendet, wenn eine Verbindung besteht.

Mit erfolgter Verbindung startet eine *Session* zur Übertragung der PNT-Daten. Der Datenempfänger führt in einer Endlosschleife eine *httpget()*-Anfrage aus, um epochenweise ein JSON-Objekt übertragen zu bekommen. Mit dem JSON-Objekt wird ein Objekt der Klasse PNT-Informationen instanziiert, welches der Oberfläche bekannt gemacht wird, um den Inhalt der darstellenden Klassen mit den momentanen PNT-Informationen zu aktualisieren. Die Antwort kann allerdings auch eine Fehlermeldung sein, bedingt durch eine - nicht durch die Anwendung - erzwungene Beendigung der Session. In dem Fall soll der Datenempfänger die Verbindung neu aufbauen. Dies entspricht einem Sprung zurück in die obere der beiden Schleifen.

Die Session wird beendet durch Aufruf der entsprechenden Methode des Datenempfängers. Abbildung 25 stellt die Kommunikation als Sequenzdiagramm dar.

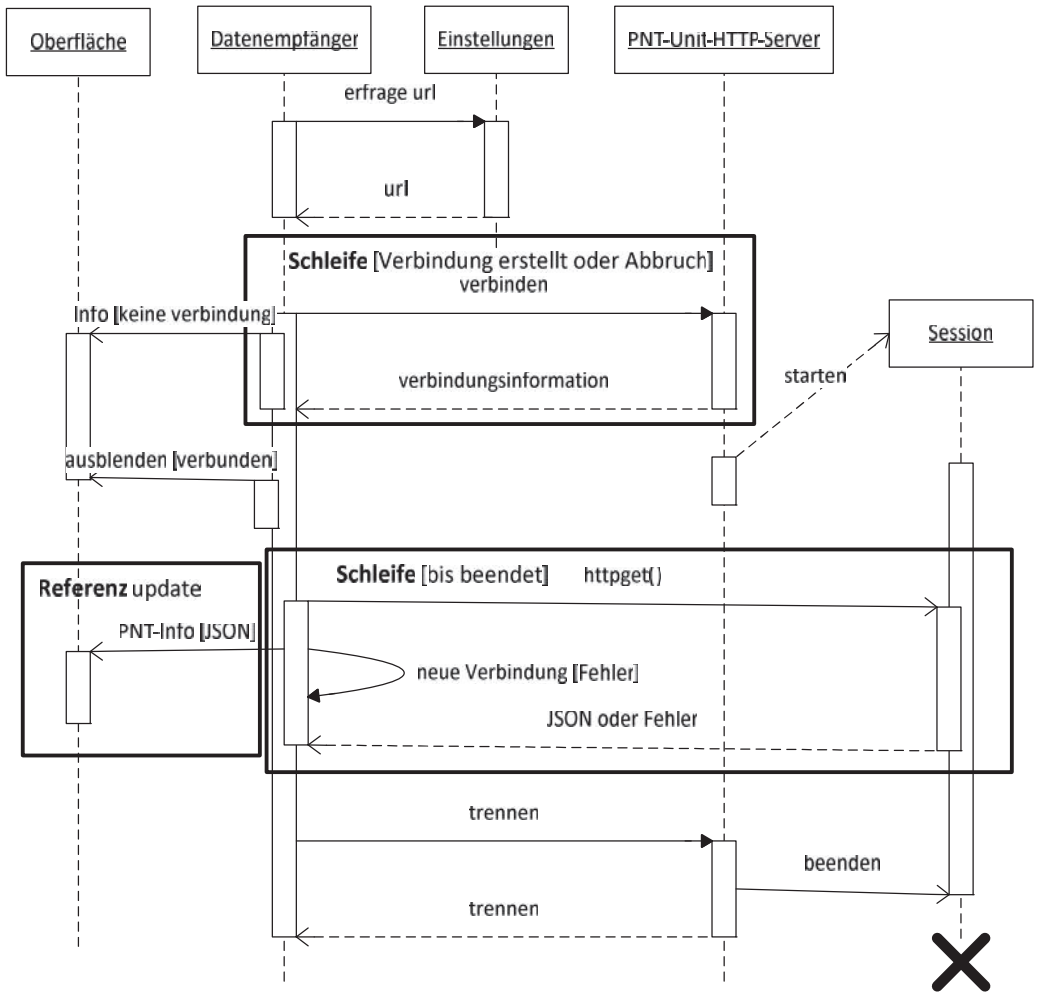


ABBILDUNG 25: SEQUENZDIAGRAMM VERBINDUNGSAUFBAU

### *3.4.2 Aktualisierung der Komponenten der Oberfläche*

Die epochenweise Aktualisierung der darstellenden Klassen der Visualisierung mit neuen PNT-Informationen wird durch die ***Referenz update*** aus Abbildung 25 angedeutet und ist in Abbildung 26 ausführlich modelliert. Ausgangspunkt ist der Datenempfänger, der zu einer Epoche ein *JSON-Objekt* empfangen hat und dieses durch Aufruf der implementierten *update()*-Methode der Oberfläche bekannt gibt. Die Oberfläche realisiert mit dem JSON-Objekt ein neues, PNT-Objekt genanntes, Objekt der Klasse PNT-Informationen.

Das PNT-Objekt (PNT-Obj) wird asynchron an die beteiligten Komponenten der Oberfläche gesendet. Modelliert in Abbildung 26 sind beispielhaft die Komponenten zur Karten- und Dialogdarstellung. Die Kommunikation erfolgt für jede weitere Komponente (beispielsweise die Titelleiste) gleich, da alle Komponenten die *receiveData()*-Methode der Elternklasse implementieren. Die Oberfläche lässt sich so auch um beliebige weitere Komponenten

ergänzen. Mit den jeweils benötigten Einzelinformationen aus dem PNT-Objekt aktualisieren die Komponenten ihren Inhalt parallel.

Das Weiterreichen des (vollständigen) JSON- bzw. PNT-Objekts ist jeweils als asynchroner Methodenaufruf modelliert, um zu gewährleisten, dass die interne Kommunikation angesichts der hohen Datenrate die darstellenden Klassen nicht in ihrer Kernaufgabe beeinträchtigt.

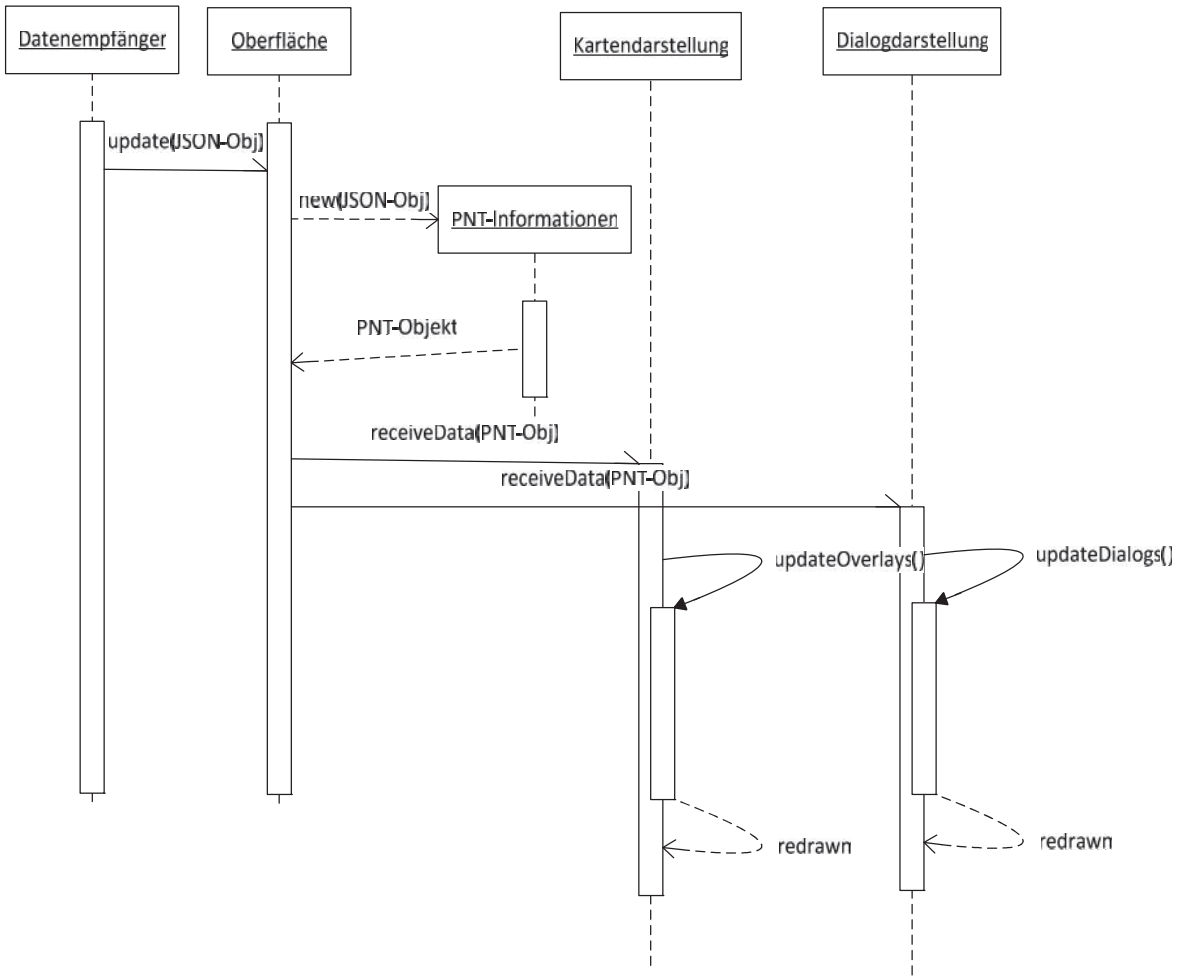


ABBILDUNG 26: SEQUENZDIAGRAMM UPDATE MIT PNT-INFORMATIONEN ZU EIENR EPOCHE

# 4 IMPLEMENTIERUNG

Für die Implementierung wird das Betriebssystem **Android** in der **Version 4.4.2** verwendet. Android bietet einen guten Kompromiss aus bereitgestelltem Framework zur Umsetzung des Architekturentwurfs, vorhandenen Schnittstellen und Preis für die für einen Prototyp benötigte Hardware sowie eine große Fülle an Geräten, die die Anwendung darstellen können. Als Endgerät wird ein handelsübliches Android-fähiges Gerät verwendet, das im Bereich der 10-Zoll-Bilddiagonale liegt. Typische Geräte dieser Kategorie besitzen einen kapazitiven Bildschirm. Dieser ist für Anwendungen auf See zwar nur bedingt geeignet, für einen ersten Prototyp genügt es aber und als Vorführgerät ist es vorteilhaft. Zu einem späteren Zeitpunkt ist es immer noch möglich auf ein Gerät mit hohem IP-Schutzgrad (vgl. 3.1.2) zu wechseln, wenn dies notwendig sein sollte. Aus den vorgenannten Gründen werden im Vorfeld zwei Entscheidungen getroffen:

- Es wird die Android-Kompatibilitätsbibliothek verwendet, um Abwärtskompatibilität bis Version 2.3 (API Level 9) des Betriebssystems zu erreichen. So sind 99,3% aller auf dem Markt vorhandenen Android-Geräte hinsichtlich der installierten Android-Version abgedeckt [17]
- Der Bildschirminhalt wird auf Basis von Fragmenten (Klasse Fragment[12]) entwickelt, um unterschiedliche Bildschirmgrößen gleichzeitig zu unterstützen. Eine große Mehrheit der Android-Geräte (81,7% nach [17]) besitzt nur einen „normal“-großen Bildschirm (ca. 4 Zoll Bildschirmdiagonale). Fragmente erleichtern im Sinne der Objektorientierung das Modularisieren des Bildschirms.

Android erfüllt die Anforderungen an die Nutzerschnittstelle (3.2.3) und es bietet fertige Pakete, um die Konfigurationsschnittstelle und die Schnittstelle für den Datenempfang auf TCP/IP-Basis zu implementieren. Als freie Software erleichtert Android die Implementierung durch die Möglichkeit, Pakete und Bibliotheken Dritter unproblematisch einbinden zu können. Die hohe Verbreitung von Android[13] erleichtert die Erfüllung der Anforderung an eine leichte und intuitive Bedienbarkeit. Ein weiterer Grund war, dass seitens des Autors Vorkenntnisse in der Entwicklung mit Android vorhanden waren.

MVC ist in Android nicht unterstützt. Es lässt sich argumentieren, dass Layout-Ressourcen, die in XML-Dateien (Extensible Markup Language) definiert werden können, die Präsentation sind und die Java-Klassen, die die Daten beinhalten das Modell – eine Steuerung lässt sich aber nicht als Einheit umsetzen. Dies rührt daher, dass in Android die Darstellung und Verwaltung der Oberfläche und die Manipulation deren Inhalts durch ein einzelnes Objekt der Klasse *Activity* übernommen wird. Die *Activity* bietet ein Fenster, in die die Oberfläche der Nutzerschnittstelle platziert wird sowie Rückruf-Methoden, in denen auf verschiedene System- und Nutzerereignisse reagiert werden kann. Eine *Activity* ist immer der Eingangspunkt der Anwendung. Ordnet man nun die *Activity* der Präsentation zu, verletzen die Rückruf-Methoden, die der Steuerung zugeordnet werden müssen, das MVC-Muster.

[12] http://developer.android.com/reference/android/app/Fragment.html
[13] fast 85% nach [17]

Ordnet man die *Activity* der Steuerung zu, dann verletzt die Möglichkeit, dass die *Activity* - beispielsweise in den Rückruf-Methoden - direkten Zugriff auf die Oberfläche hat und sich selbst bei der Erstellung mit einer Präsentationsressource instanziiert, MVC. Um das MVC-Muster dennoch zu folgen, werden folgende Entscheidungen getroffen:

- Da es nicht Gegenstand der Arbeit ist, MVC in Android abzubilden, ist es vertretbar, das Muster zugunsten der vorhandenen Android-Paradigmen aufzuweichen, wenn es keinen Mehrwert bietet, MVC durch einen „Workaround" zu erzwingen.
- Das Modell ist eine Java-Klasse und bildet die PNT-Informationen ab. Dies ist konform mit MVC.
- Es gibt in der Anwendung nur eine *Activity*. Dies ist die *Activity*, die beim Start der Anwendung ausgeführt wird. Diese *Activity* übernimmt teilweise die Rolle der Steuerung. Sie verwaltet die Präsentation und manipuliert die Beobachter hinsichtlich Änderungen am Modell, indem sie eine entsprechende *update()*-Methode bereitstellt, über die die Komponenten der Präsentation über Änderungen am Modell benachrichtigt werden können.
- Die *Application*-Klasse wird überschrieben und dient als Teil der Steuerung zur Speicherung und Abfrage von Zuständen der Anwendung. Ein Objekt der Klasse ist vom Android-System global abrufbar.
- Da es nur eine *Activity* gibt, erfolgt die Präsentation in Fragmenten. Die Fragmente stellen die Beobachter dar. Entsprechend der Aufteilung des Bildschirms in logische Teile (wie in Abbildung 12 dargestellt) wird je ein Fragment für die Kartendarstellung und die Dialogdarstellung bereitgestellt. Die Fragmente sind Bestandteil der einen *Activity* und werden durch sie instanziiert. Für die Titelleiste und Menüleiste kann auf die Android-interne *Actionbar* zurückgegriffen werden.

## 4.1 Bildschirm und Schnittstellen

### *4.1.1 Benutzerschnittstelle*

Nutzereingaben über den berührungsempfindlichen Bildschirm werden durch das Android-Frameworks erfasst. Durch „Hook"-Methoden ist es möglich, auf diverse Ereignisse zu reagieren. Andernfalls soll die Verarbeitung der Ereignisse komplett dem Android-System überlassen bleiben. Entsprechend 3.2.3 wird der Bildschirm in drei Bereiche aufgeteilt, wie es in

Abbildung 27 oben ersichtlich ist: es gibt zwei Hauptbereiche für die Karte und für Dialoge, die den Großteil des Bildschirms beanspruchen und ein Menü in Form der Android-typischen *ActioBar* als Menüleiste. Die Menüleiste besitzt als eingegliederte Komponente eine sogenannte *Custom ActionBar*, die die vierte Komponente der Aufteilung des Bildschirms (vgl. Abbildung 12), die Titelleiste, umsetzt.

Da das Ausfüllen dieses Gerüsts Fragment-basiert ist und Fragmente nur den zuvor definierten Fragment-Containern zugewiesen werden, ist es möglich, Karte und Dialogbereich zu vertauschen. Wenn einer der Container, wie in

Abbildung 27 oben, größer ist als der andere, erlaubt es dies, dem Nutzer die Option anzubieten, den aus seiner Sicht wichtigeren Bereich in den größeren Container zu verschieben. Im Menü der *ActionBar* muss dafür ein entsprechendes Kontrollelement geschaffen werden.

Abbildung 27 unten zeigt die zweite Option der Nutzeroberfläche, einen der beiden Container auf den ganzen Bildschirm zu vergrößern, um so entweder nur die Karte oder nur den Dialogbereich anzuzeigen. Auch dafür muss ein Kontrollelement in der Actionbar angeboten werden.

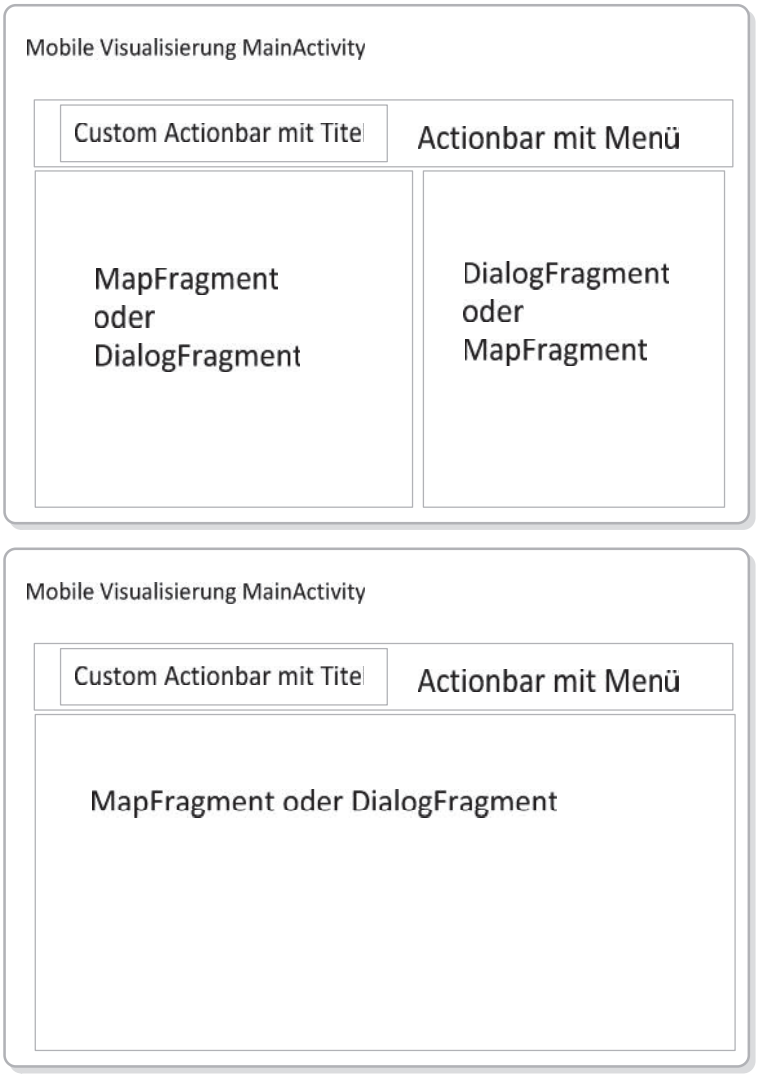


Abbildung 27: Nutzeroberfläche in Android Implementierung

### 4.1.2 MENÜ DER ACTIONBAR

Das Menü der *ActionBar* wird in der *MainActivity* (4.3.3) aufgebaut. Es beinhaltet Kontrollelemente für folgende Aktionen:

- Tauschen und Vergrößern der Fragment-Container (4.1.1)
- Aufrufen der Einstellungen (4.1.3)
- Verbinden mit und Trennen von der PNT-Unit (4.3.4)
- Zentrieren des Kartenausschnitts über dem Schiff

Die Kontrollelemente rufen Funktionen, die in den jeweils in Klammern angegebenen Kapiteln beschrieben werden, auf.

### 4.1.3 KONFIGURATIONSSCHNITTSTELLE UND EINSTELLUNGEN

Die Strukturierung der Einstellungen ist in XML-Ressourcen festgelegt. Entsprechend der in 3.3.3 festgelegten Unterteilung der Einstellungen in Kategorien, existiert für jede Kategorie ein XML-Dokument. Einer einzelnen Einstellung kann so ein Schlüssel in der Form *‚android:key="KEY_SHIP_NAME"‘* zugewiesen werden, mit dem der momentane Wert der Einstellung manipulier- und abrufbar ist. Die XML-Dokumente werden in einer Klasse *SettinsgActivity.java* zu einem einheitlichen Bildschirm zusammengefügt. Die Klasse kann über ein Kontrollelement in der Menüleiste aufgerufen werden. Die Verwaltung der Einstellungen und die Speicherung der Schlüssel-Wert-Paare über die Lebenszeit eines Programmablaufs hinaus, werden an das Android-System delegiert. Um den Wert eines Einstellungsparameters zu einem beliebigen Zeitpunkt abrufen zu können, muss ein Einstellungsobjekt vom Android-System angefordert werden, von dem sich der Wert über seinen Schlüssel abfragen lässt.

Die Standardeinstellungen, sind als Textdatei hinterlegt, welche dem Format, wie in Abbildung 28 ersichtlich, folgt. Der Wert in den eckigen Klammern entspricht dem Schlüssel, der darauf folgende Wert dem Wert einer Einstellung. Kommentare können mit einer Raute ausgezeichnet werden. Um die Textdatei in die Einstellungen zu übertragen, bietet die *SettinsgActivity.java* eine Methode, um Einträge als Schlüssel-Wert-Paar aus der Datei in die Einstellungen zu übernehmen. Kommentare ignoriert die Methode. Es ist auch möglich, eine eigene Konfigurationsdatei auf das Gerät zu laden und in die Einstellungen zu übertragen, wenn das Format der Datei der Spezifikation entspricht.

```
#Kommentar
[SHIP_NAME]
Baltic Taucher 2

[SHIP_IMO]
9096387

[SHIP_CALLID]
DECT2

[SHIP_LENGTH]
29.0

[SHIP_WIDTH]
7.0

[SHIP_CONTOUR]
10.5 0.0 0.0
9.0 3.5 0.0
-18.5 3.5 0.0
-18.5 -3.5
9.0 -3.5 0.0
```

ABBILDUNG 28: FORMAT DER STANDARDEINSTELLUNGEN

Damit Änderungen an den Einstellungen mit von den Einstellungen abhängigen Elementen in anderen Klassen synchronisiert werden können, müssen die entsprechenden Klassen die *OnSharedPreferenceChangeListener*-Schnittstelle implementieren. Da alle Elemente der Oberfläche Bestandteil der *MainActivity* sind, reicht es, wenn die *MainActivity* diese Schnittstelle implementiert und Änderungen entsprechend an die Unterklassen weiterreicht.

## 4.2 LEBENSZYKLUS DER ANWENDUNG

Abbildung 29 skizziert den Ablauf der Anwendung, beginnend beim Programmstart bis zu dem Punkt, ab dem die Nutzerin oder der Nutzer die Kontrolle über das Programm hat. Entsprechend dem Wesen einer Android-Anwendung besteht das Programm aus Komponenten die nur einmal beim Programmstart und solchen die beliebig oft instanziiert werden. Nach Start des Programms wird ein *Application*-Objekt[14] angelegt und Variablen, die den Zustand verschiedener Aspekte der Anwendung speichern gesetzt (vgl. 4.3.2). Darauf wird die *MainActivity* gestartet. Dies kann sich im weiteren Programmverlauf beliebig oft wiederholen. Jedes Mal, wenn das Anzeigegerät gedreht oder gekippt wird oder wenn die *Activity* in den Hintergrund verschoben und wieder in den Vordergrund geholt wird, wird sie neu aufgebaut, wobei Zustände und Inhalt verloren gehen, wenn diese nicht behandelt

[14] http://developer.android.com/reference/android/app/Application.html

werden[15]. Um diesen Wiederherstellungsprozess zu handhaben, bietet die Klasse *Activity* verschiedene Callback-Methoden, in denen eben benannte Aspekte gesichert und wieder hergestellt werden können. Dafür wird die *Application*-Klasse überschrieben und um entsprechende Zustandsvariablen ergänzt.

Nachdem die *MainActivity* gestartet wurde, wird ein *Application*-Objekt nebst einem Einstellungsobjekt angefordert, mit denen entweder ein definierter Startzustand oder der jeweils letzte Zustand wieder hergestellt wird. Danach kann der Nutzer mit der Anwendung arbeiten.

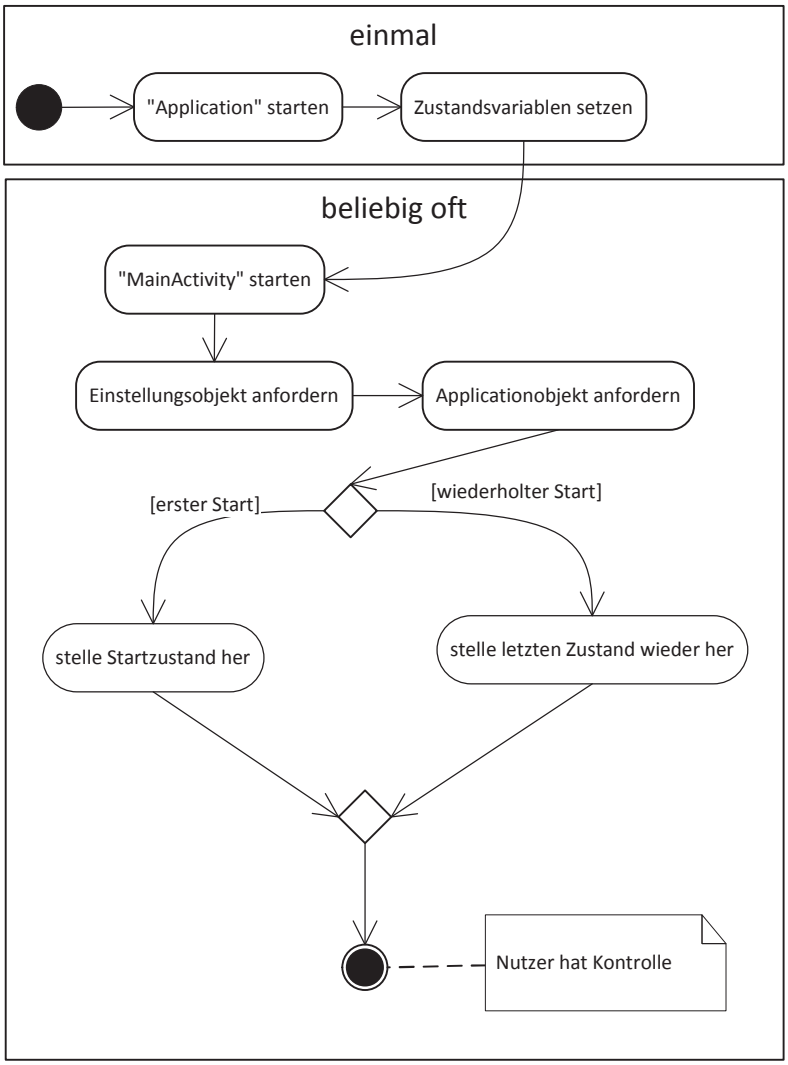


ABBILDUNG 29: FLUSSDIAGRAMM INSTANZIIERUNG MAINACTIVITY

[15] http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle

## 4.3 WICHTIGE KLASSEN

### *4.3.1 DIE MODELL-KLASSE PNTDATA*

Für das Modell ändert sich gegenüber dem Entwurf aus Abbildung 17 nicht viel. Das Modell der Anwendung wird mit der Klasse *PntData* realisiert. Die PNT-Informationen speichert die Klasse in einer *Map*. Jede einzelne PNT-Information wird in der *Map* anhand ihres Schlüssels, so wie es im JSON-Objekt empfangen wird, als *PntString* verwaltet. *PntString* ist eine Wrapper-Klasse des Typs *java.lang.String*, die eine einzelne PNT-Information als *String*-Variable hält und Methoden zur Typumwandlung in verschiedene Basistypen bietet. Die Schlüssel befinden sich als statische Felder in der *PntData*-Klasse. So ist es beispielsweise möglich, eine Anfrage an ein *pntData*-Objekt der Form *„pntData.get(PntData.KEY_TIME_GPS_MILLISEC).toLong()“* zu stellen, um den Zeitstempel des PNT-Datensatzes als Long-Wert zu erhalten. Darüber hinaus stellt die *PntData*-Klasse „Getter“-Methoden für jede einzelne benötigte Information zur Verfügung, die die jeweilige Information in dem Datentyp, wie es an anderer Stelle benötigt wird, und teilweise bereits formatiert oder aus verschiedenen Informationen berechnet, zurück. „Setter“-Methoden gibt es nicht, da in Übereinstimmung mit dem Entwurf die *Map* ausschließlich über den Konstruktor der Klasse instanziiert wird.

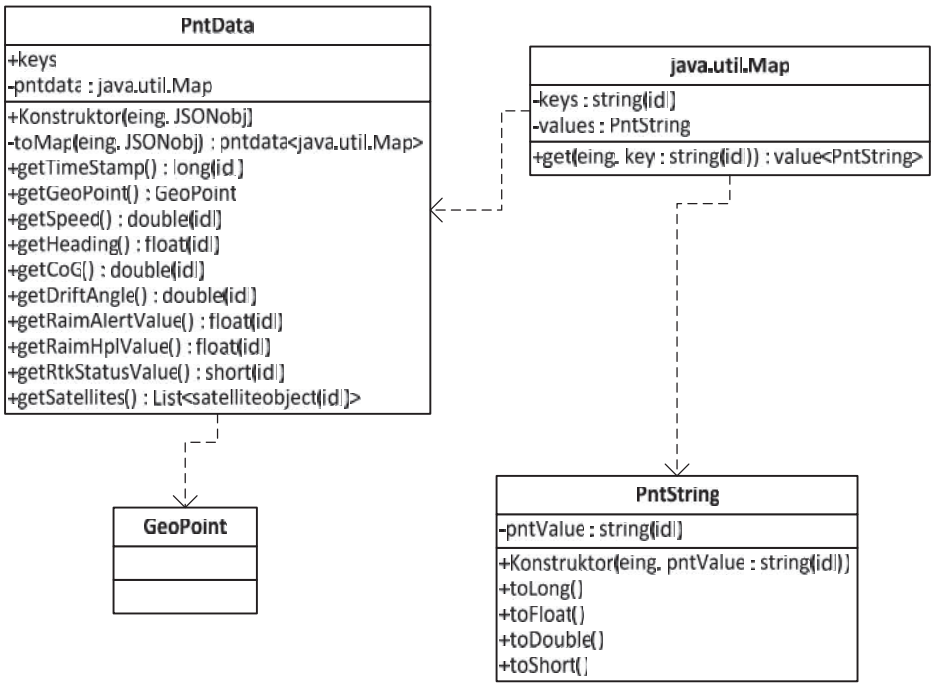


ABBILDUNG 30: DIE PNTDATA KLASSE

### 4.3.2 DIE APPLICATION-KLASSE

Die *Application*-Klasse bietet global Zugriff auf globale Zustände der Anwendung, die durch das Android-System verwaltet werden. Android instanziiert ein Objekt dieser Klasse beim Programmstart. Die Klasse wird überschrieben und zum Zweck der Verwaltung von Zuständen der Anwendung erweitert. Ein Objekt zum Setzen oder Abfragen eines dort verwalteten Zustandes kann durch *„(PntContext) getApplicationContext()*“ abgefragt werden.

Abbildung 31 zeigt die Klasse mit den wichtigsten Feldern und Methoden. Folgende Zustände werden als boolescher Wert verwaltet:

- Befindet sich die Anwendung im Modus mit zwei Fenstern oder mit einem?
- Ist die Anwendung mit der PNT-Unit verbunden?
- Wird gerade versucht, eine Verbindung aufzubauen?
- Zeigt die Karte ein Popup-Fenster mit Informationen des Schiffs, wie in den Ausführungen zu *Overlays* in 4.3.5 beschrieben?

Außerdem werden Instanzen der momentanen Fragmente in einer Datenstruktur *Map* vorgehalten, um die Fragmente während des Startprozesses der *MainActivity* den Containern korrekt zuzuweisen. An Methoden bietet die Klasse Setter und Getter für jeden Zustand und Methoden, die alle Zustände speichern oder wieder herstellen, sowie eine Methode, um einen definierten Startzustand herzustellen. Der Startzustand ist so definiert, dass alle PNT-Informationen und Dialoge, die für den Anwendungsfall Baltic Taucher II (1.4.2) benötigt werden, dargestellt werden.

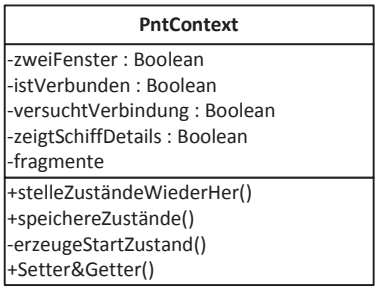


ABBILDUNG 31:APPLICATION KLASSE

Die Zustände der Anwendung hätten alternativ auch in einem Singleton-Objekt verwaltet werden können. Die Entscheidung fiel auf das Erweitern der *Application*-Klasse, da diese Klasse vom Android-System verwaltet wird und so mit Sicherheit über die komplette Lebensdauer der Anwendung vorhanden ist und, auch wenn die Anwendung in den

Hintergrund verschoben wird, wird der Speicher nicht freigegeben bzw. die Attribute der Klasse nicht zurückgesetzt.

### 4.3.3 DIE MAIN-ACTIVITY

Die *MainActivity* ist die einzige *Activity* der Anwendung, die der Visualisierung von Schiffsinformationen dient. Die Einstellungen (4.1.3) sind auch als *Activity* umgesetzt, sie dienen aber nicht primär der Visualisierung. Die *MainActivity* ist mit den wichtigsten Methoden, Feldern und Schnittstellen in Abbildung 32 dargestellt.

Als Schnittstellen realisiert die Klasse *onSharedPerferenceChangeListener*, um auf Änderungen an den Einstellungen zu reagieren, *OnDialogItemCheckedListener*, womit auf alle Menüinteraktionen im Dialogbereich reagiert werden kann und *OnNavigationDestinationSetListener*, was die Zielkoordinaten, die der Nutzer im Dialogbereich (4.3.6) setzt, an das Kartenfragment zur Darstellung weitergibt. Als private Felder besitzt die Klasse eine Instanz der Einstellungen und des „*Contexts*“ (4.3.2), um Zustände der Anwendung abzufragen oder zu setzen sowie ein Objekt des Datenempfängers (4.3.4). Die Lebenszyklus-Methoden *onCreate, onSavedInstanceState* und *onPause* werden genutzt, um die Zustände zu speichern oder wieder herzustellen, bzw. zu instanziieren. Die Menüleiste (4.1.2) wird in der Rückruf-Methode *onCreateOptionsMenu* erstellt. Nutzerinteraktionen mit Kontrollelementen des Menüs werden in *onOptionsItemSelected* behandelt. Dazu bietet die Klasse Methoden, um sich mit dem Datenempfänger zu verbinden (*PntConnect()*) und sich davon zu trennen (*PntCancel()*), welche die entsprechenden Klassenmethoden des Datenempfängers aufrufen. Die empfangenen PNT-Informationen werden in der *update*-Methode an die Fragmente weitergeleitet.

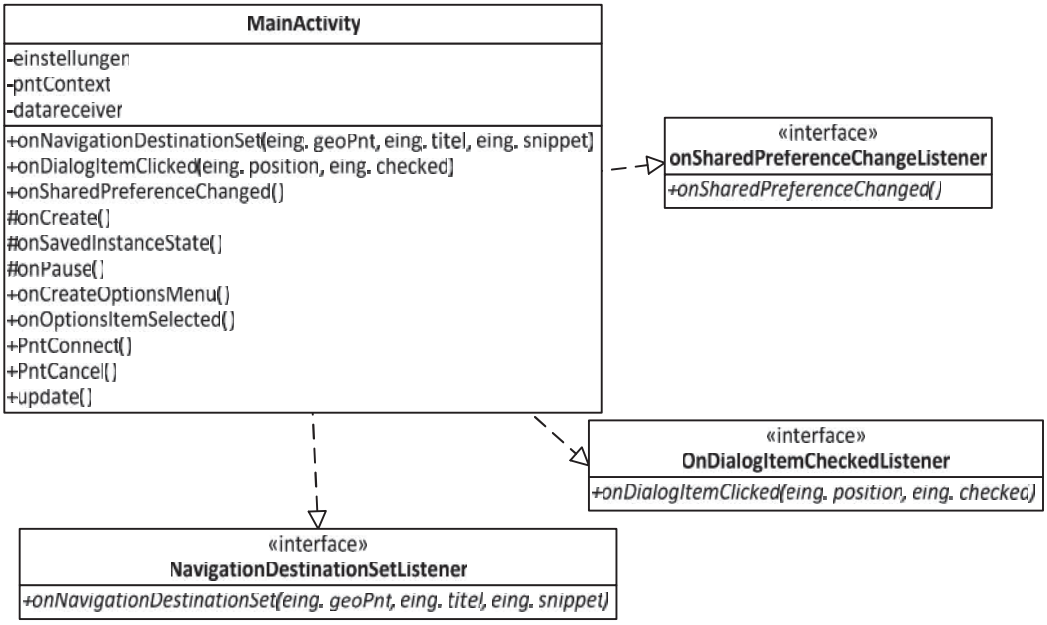


ABBILDUNG 32: KLASSE MAINACTIVITY

### 4.3.4 DER DATENEMPFÄNGER

Entsprechend 3.3.4 beinhaltet der Datenempfänger (*PntDataReceiver*) einen HTTP-Client (*HttpHelper*), der die Methoden besitzt, mit denen sich die Anwendung mit der PNT-Unit verbinden, davon trennen und per *httpGet*-Anfrage die PNT-Informationen als String-Objekt im JSON-Format beziehen kann. Für die Methoden zum Verbinden und Trennen werden Kontrollelement im Menü bereitgestellt (vgl. 4.1.2).

Da der Datenempfänger, wie in Kapitel 3.4.1 beschrieben, in einer Schleife läuft, erweitert die Klasse die Android-Klasse *android.os.AsyncTask*, um die Kommunikation mit der PNT-Unit in einem Thread im Hintergrund auszuführen. Die Schleife wird in diesem Thread ausgeführt. Auch wenn die Klasse *AsyncTask* für kurze Hintergrundoperationen gedacht ist, kann so sichergestellt werden, dass Vordergrundoperationen der Nutzerschnittstelle nicht beeinträchtigt werden. Zur Kommunikation mit der *MainActivity* als Verwalter der Nutzerschnittstelle, wird die Methode *puplishProgress* genutzt. Diese Methode dient eigentlich dazu, um beispielsweise während eines Download-Prozesses den Fortschritt zum Zweck der Darstellung an die Nutzerschnittstelle zu senden. Im Datenempfänger wird die

Methode jedes Mal aufgerufen, wenn die *HttpHelper*-Klasse ein neues PNT-Datenobjekt empfangen hat.

Der Hintergrundprozess und damit ein Verbindungsaufbau wird mit *execute* gestartet und mit *cancel* wieder unterbunden. Der Status des Prozesses lässt sich mit *getStatus* abfragen. Der Datenempfänger besitzt auch ein Objekt der *Application*-Klasse (4.3.2), um den Status der Verbindung zu setzen.

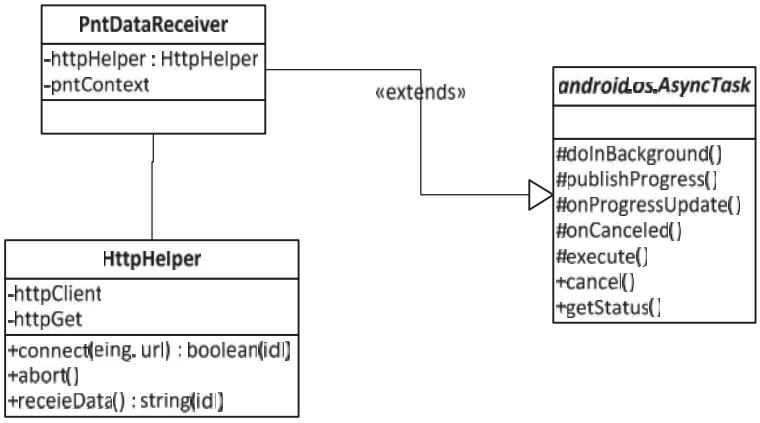


ABBILDUNG 33: DATENEMPFÄNGER

Abbildung 34 zeigt die Zustände, in denen sich der Datenempfänger im Hintergrundprozess befinden kann, nachdem in der *MainActivity* ein *execute*-Ruf ausgeführt wurde. Zuerst wird versucht eine HTTP-Verbindung mit der PNT-Unit aufzubauen und entsprechend des Rückgabewertes des *httpHelper*-Objekts die Zustandsvariablen in der Umgebung der Anwendung gesetzt (4.3.2). Wenn keine Verbindung aufgebaut werden konnte, wird der Hintergrundprozess beendet und er muss von außen neu gestartet werden. Andernfalls wird in dem Thread eine Endlosschleife gestartet, in der epochenweise PNT-Daten vom *httpHelper*-Objekt angefragt werden. Wenn die Daten ein regulärer JSON-String sind, wird durch einen Ruf an *publishProgress()* die *update*-Methode der *MainActivity* (4.3.3) aufgerufen. Dies geschieht in der *onProgressUpdate*-Methode des asynchronen Hintergrundprozesses. Die Schleife kann währenddessen in die nächste Iteration gehen. Der Hintergrundprozess ist beendet, wenn das *httpHelper*-Objekt einen Fehlerwert zurückgibt oder wenn der Nutzer die Verbindung manuell beendet.

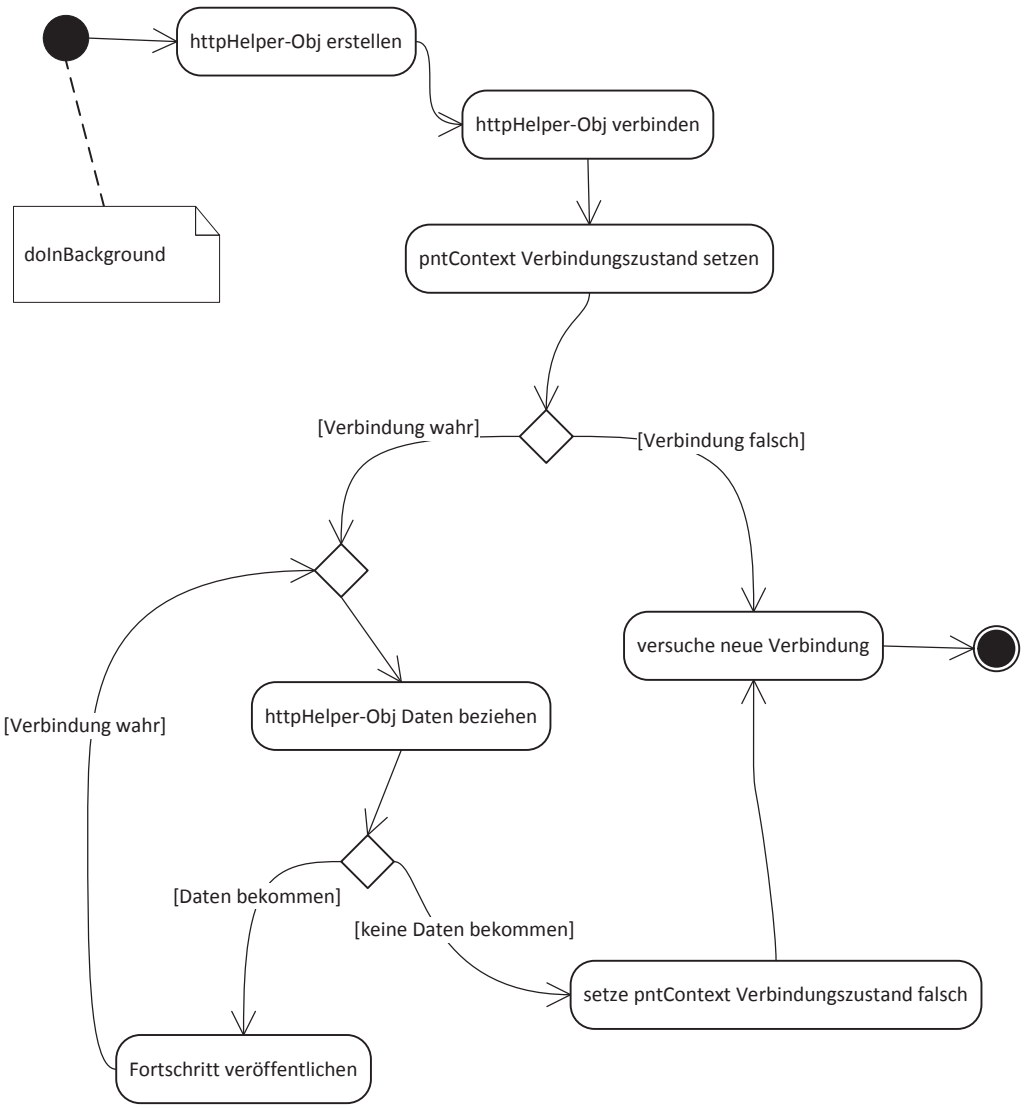


ABBILDUNG 34: DATENEMPFÄNGER ZUSTANDSDIAGRAMM

### 4.3.5 KARTENDARSTELLUNG UND KARTENDATENBASIS

Wird für die Kartendarstellung Google Maps oder die freie, an Google Maps angelehnte, *openstreetmap (osm)*-Bibliothek Osmdroid verwendet, dann wird die Karte in einer *MapView* dargestellt. Eine *MapView* ist eine Unterklasse der *View* [16] und bildet damit eine nutzerschnittstellenbezogene Layout-Komponente. Typischerweise wird die Karte in der *MapView* aus einzelnen Kacheln (Tiles) zusammengestellt, die von einer entfernen (Online-

[16] http://developer.android.com/reference/android/view/View.html

)Quelle bezogen werden. Die Quelle der Kacheln sind bei Google Maps und *osm* ein entsprechender Server als Datenquelle.

Einer Osmdroid-MapView kann im Konstruktor ein *TileProvider* übergeben werden, der Anfragen nach Kacheln verarbeitet und für eine Koordinate Kacheln bereitstellt. Die *TileSource*-Klassen kapseln Informationen über eine Kachel und stellen Rendering-Methoden und verschiedene Methoden zum Beziehen der Kachel von einer Online-Ressource als Stream bzw. aus einer lokalen Dateiressource bereit.

Während *osm* mit speicherintensiven Rasterdaten arbeitet, stellt MapsForge[17] eine Bibliothek bereit, um (offline) speichersparende Vektordaten in einer *MapView* zu rendern. Um die Vorteile eines vektorbasierten Datenbestandes mit den Möglichkeiten, eine *MapView* um rasterbasierte *Overlays* (eigenes Schiff, Seezeichen, Bahn, etc.) zu verbinden, soll kurz beschrieben werden, wie es umgesetzt wurde, Kartendaten aus einer vektorbasierten Datenbasis (*MapsForge*) in eine rasterbasierte *MapView* zu rendern. Die Skalierungen der Karte richten sich nach den Möglichkeiten der Osmdroid-Bibliothek. Typischerweise wird der Kartenausschnitt einer *MapView* bei einer Änderung des Maßstabs in Höhe und Breite um den Faktor 2 verändert.

Der Ansatz knüpft an das bei [18] beschriebene Vorgehen an und wurde für eine aktuelle Android-4.x-kompatible *MapsForge*-Version adaptiert. Für die Kartendarstellung steht eine *GenericMapview* genannte Klasse bereit, die die *ViewGroup*[18]-Klasse *FrameLayout* erweitert. *GenericMapview* bietet eine Methode *setTileProvider()*, mit der ein *TileProvider* übergeben werden kann, mit dem eine Osmdroid-*MapView* instanziiert und die dann als privates Attribut gespeichert wird. Die *MapView* wird dem Container (*GenericMapview*) als Kind hinzugefügt und kann über eine *get*-Methode (*getMapView()*) wieder abgefragt werden. So ist es auch möglich, eine herkömmliche Osmdroid-*MapView* mit Online-Kachelquelle im selben Layout-Element darzustellen, wie eine aus einem MapsForge-Datensatz gerenderte Karte.

[17] http://wiki.openstreetmap.org/wiki/Mapsforge
[18] eine Containerklasse zum halten anderer View-Objekte, vgl. http://developer.android.com/reference/android/view/ViewGroup.html

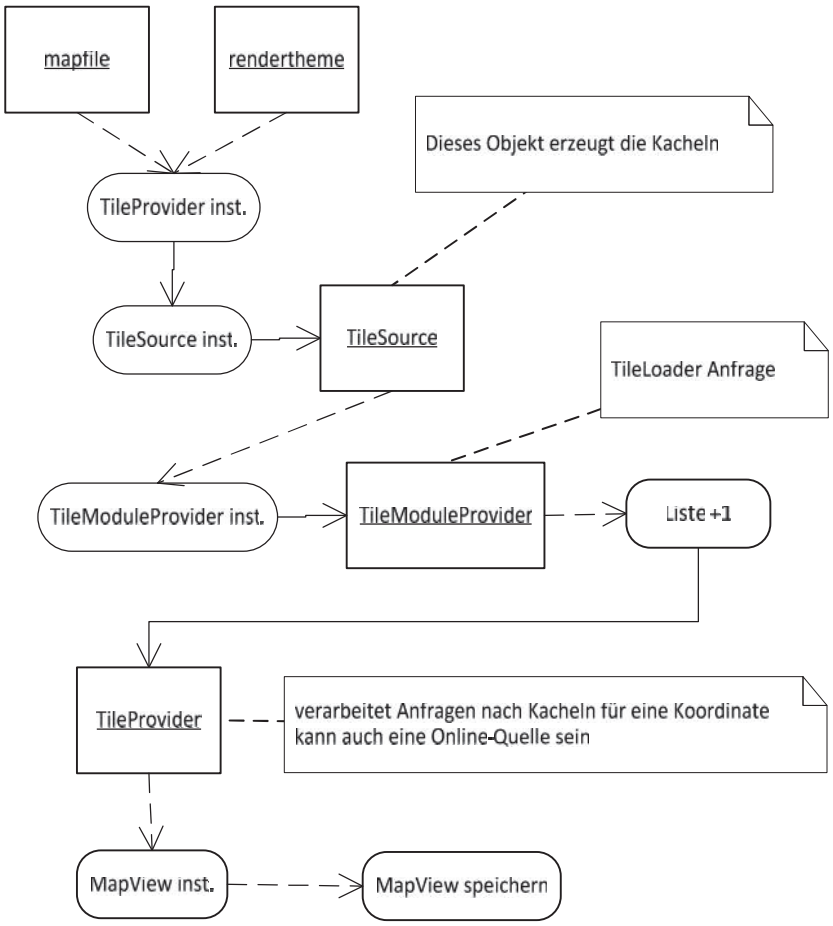


ABBILDUNG 35: ABLAUF EINER MAPVIEW INSTANTIIERUNG MIT VEKTORDATENBASIS

Abbildung 35 stellt den Ablauf dar, wie aus einer vektorbasierten Kartendatei (*mapfile*) und einer XML-basierten Anweisung, wie die Kartendaten in der *MapView* gerendert werden sollen (*rendertheme*), eine Kartendarstellung entsteht, die in einer rasterbasierten *MapView* verwendet werden kann. Die *MapsForgeTileProvider*-Klasse ist der *TileProvider*, mit dem die *MapView* instanziiert wird (Konstruktor), wenn die Datenquelle ein lokal vorhandener Vektordatensatz ist. Sie wird mit dem Pfad zu den Daten (*mapfile*) und einer XML-Ressource mit Renderanweisungen (*rendertheme*) instanziiert. Es wird ein neuer *TileModuleProvider* instanziiert und einer Liste von *TileModuleProvider*n hinzugefügt. So ist es möglich, dass ein anderes Provider-Modul die Anfrage an eine Kachel bearbeitet, wenn eine vorherige Anfrage fehlschlägt oder dass ersatzweise eine Kachel eingesetzt werden kann, bis eine bessere bereitgestellt werden kann. Der Konstruktor des *MapsForgeTileModuleProvider* übernimmt ein *MapsForgeTileSource*-Objekt und startet bei einer *TileLoader*-Anfrage einen Renderauftrag mit dem Objekt.

*MapsForgeTileSource* rendert aus einer MapsForge-Datenbasis (übergeben als Pfad im Konstruktor) die dortigen Vektordaten zusammen mit Renderanweisungen (*rendertheme*) in

ein Grafikobjekt (*Drawable*), das an die *MapsForgeTileProvider*-Elternklasse (*org.osmdroid.tileprovider.MapTileProviderArray*) und damit an das Osmdroid-Framework zurückgegeben wird. Die Kachel wird nun in der Osmdroid-*Mapview* dargestellt.

Abbildung 36 stellt das MapsForgeForOsmdroid-Klassendiagramm mit allen Unterklassen und Abhängigkeiten dar.

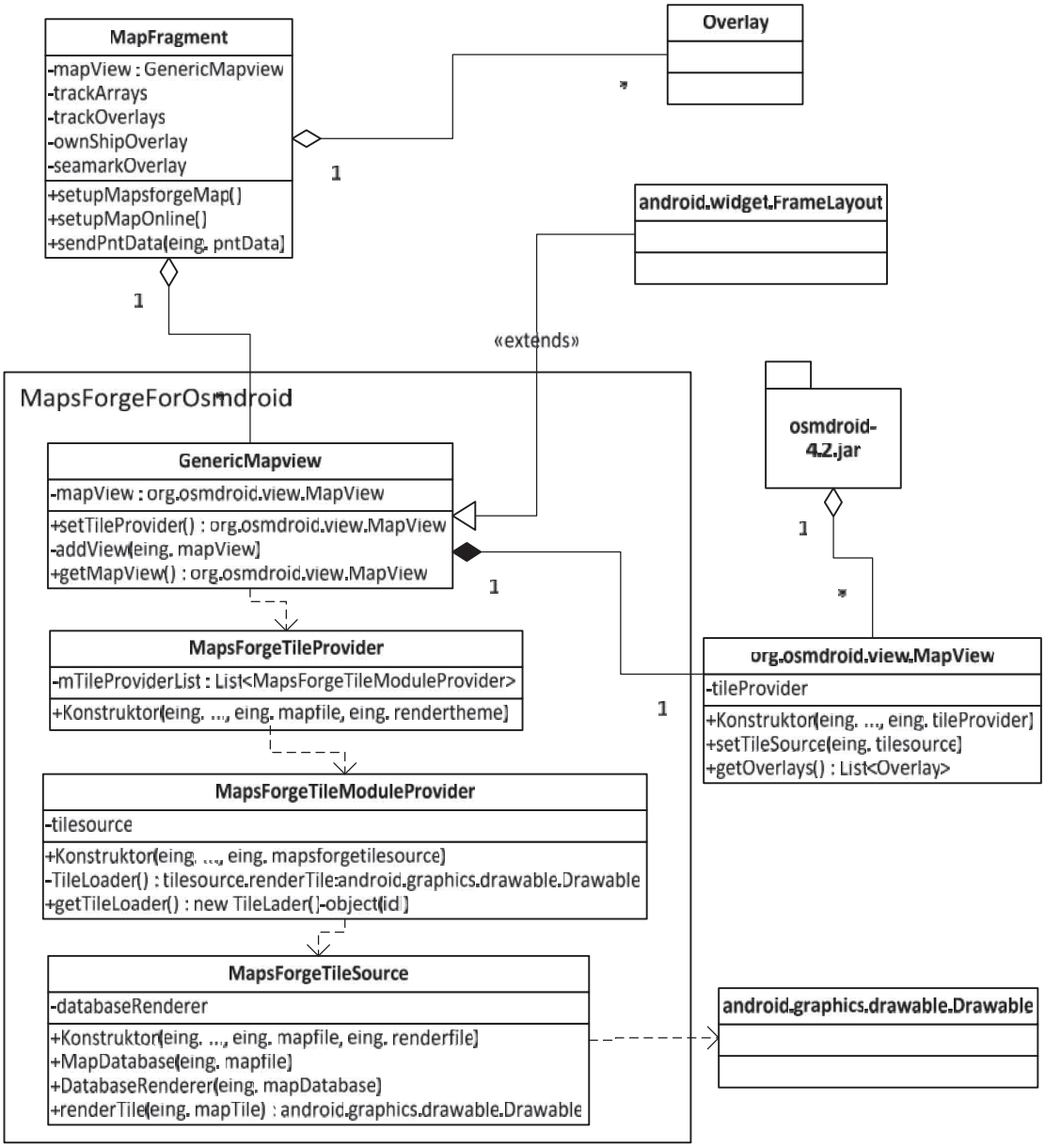


ABBILDUNG 36: MAPSFORGE FÜR OSMDROID

Die im *MapFragment* für die Kartendarstellung verwendete *MapView* ist somit ein *View*-Container vom Typ *GenericMapview*, dessen Inhalt (*View*) eine Osmdroid-*MapView* ist. Das

*MapFragment* setzt sich aus einem oder mehreren *Overlays* (Schiff, Bahn, Seezeichen) und der *GenericMapview* zusammen. Die *MapView* speichert alle angefügten Overlays in einer Liste, die sich über eine Methode (*getOverlays()*) abfragen lässt. So hat das *MapFragment* jederzeit Zugriff auf alle *Overlays* der *MapView* und kann diese entsprechend manipulieren – sie beispielsweise mit neuen PNT-Daten aktualisieren.

Die *Overlays* erben von der Klasse *Overlay*, die unter anderem eine *draw()*-Methode bereitstellt, in der der Inhalt des *Overlays* gezeichnet wird und die dafür von spezialisierten *Overlay*-Klassen überschrieben werden muss. Die Übergabeparameter sind ein *Canvas*-Objekt, auf dem die Grafik erstellt wird und ein *MapView*-Objekt der Kartendarstellung, zu der das *Overlay* gehört. Die Implementierung sei anhand des *Overlays*, das das eigene Schiff darstellt, beschrieben (Abbildung 37) – das Vorgehen ist bei jedem weiteren *Overlay* analog.

Die *Overlay*-Klasse kann ein oder mehrere *Overlay* Items (Klasse *OverlayItem*) - Entitäten mit Koordinaten und einer Beschreibung, die in der Karte dargestellt werden sollen - verwalten. Für die Darstellung des eigenen Schiffes gibt es nur ein Objekt dieser Klasse. Für die Visualisierung der Schiffsbahn oder von Geschwindigkeitsvektoren kann darauf verzichtet werden. In dem Konstruktor des *Overlays* wird ein Grafikparameter übermittelt (Marker), der als Schiffssymbol für das *Overlay* Item dient. Das Symbol entspricht dem geforderten Symbol aus 2.2.6.4. *Overlay* Items sind fokussierbar, das heißt der Nutzer kann sie anwählen. Dies wird genutzt, um ein Fenster (Popup) mit Informationen zu dem jeweiligen Item (das eigene Schiff in dem Fall) einzublenden (*popupView* und *getPopupView()*). Das Fenster ist ein Objekt der *View*-Klasse und kann über entsprechende Methodenaufrufe (*addView(), removeView()*) in der *draw()*-Methode der *MapView* hinzugefügt werden. Da ein Teil der Schiffsinformationen in den Einstellungen hinterlegt sind, beinhaltet das *Overlay* ein Objekt der Einstellungen und realisiert eine Schnittstelle, über die es auf Änderungen an den schiffsbezogenen Einstellungen reagieren kann. Die momentanen veränderlichen Schiffsinformationen (Position, Geschwindigkeit, Kurswinkel) werden über eine Methode (*addItem()*) durch das *MapFragment* an das *Overlay* gereicht. So wird ein neues Schiff dem *Overlay* hinzugefügt und das alte über eine entsprechende Methode (*remove()*) entfernt.

Die *draw()*-Methode wird vorrangig dafür genutzt, um den maßstabsgetreuen Umriss (vgl. 2.2.6.4) des Schiffes (*shipContour*) zu zeichnen. Koordinaten, bezogen auf den Referenzpunkt (CCRP) des Schiffes, befinden sich in den Einstellungen und sind über das entsprechende Einstellungsobjekt abrufbar. Das Schiff wird dabei immer erst in Richtung 0° bezogen auf die Karte gezeichnet und dann in Richtung der Peilung (*heading*-Parameter) gedreht. Die Speicherung der Koordinaten in den Einstellungen ist in Kapitel 5.1.7 beschrieben.

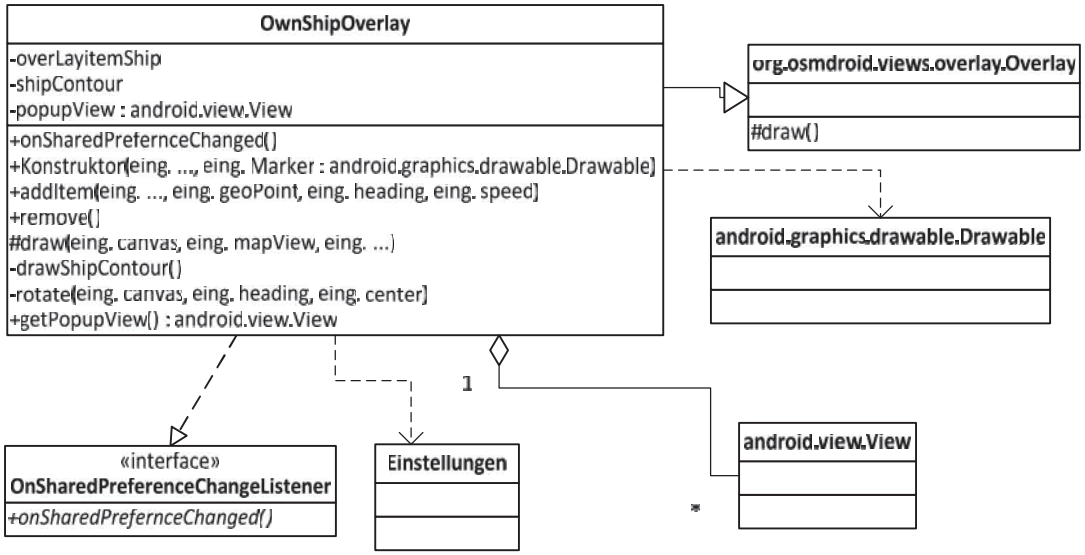


ABBILDUNG 37: OVERLAY AM BEISPIEL DES EIGENEN SCHIFFES

### 4.3.6 DIALOGBEREICH

Abbildung 38 zeigt das Klassendiagramm für den Dialogbereich. Das *DialogFragment* besteht aus einer Menge an *View*-Elementen und einer Menge an *ViewGroup*-Elementen, wobei die *Views* den dynamisch veränderlichen Inhalt darstellen und die *ViewGroups* Container für die *Views* sind. Da beide Elemente vom Typ *View* sind bzw. davon erben, kann der Dialogbereich komplett in XML umgesetzt werden. Das *View*-Element ist eine *PaintView*, die die *onDraw()*-Methode der *android.view.View*-Klasse überschreibt. Für jede relevante Einzelinformation gibt es eine entsprechende (xyz-)*PaintView*. „Xyz“ steht für das Präfix der *PaintView*, z.B. *Satellite* bei *SatellitePaintView*. Die Visualisierung der Information wird auf dem Canvas-Objekt der *onDraw()*-Methode durchgeführt.

Die abstrakte Klasse *PaintView* bietet eine *update()*-Methode, um den darzustellenden Inhalt zu aktualisieren. Der Übergabeparameter (*xyz_Info*) hängt von den konkret darzustellenden Informationen (Satelliten, Alarme, etc.) ab. Dies muss in der jeweiligen, von *PaintView* erbenden Klasse, umgesetzt werden. Da die *PaintView* das Kind eines *ViewGroup*-Elements (*LinearLayout*) ist, sind seine Dimension und Position an dieses gebunden. Um den für die

Darstellung verwendbaren Bereich der Anzeige zu erhalten, hat die *PaintView* entsprechende Methoden (*getParentWidth()*, *getParentHeight()*).

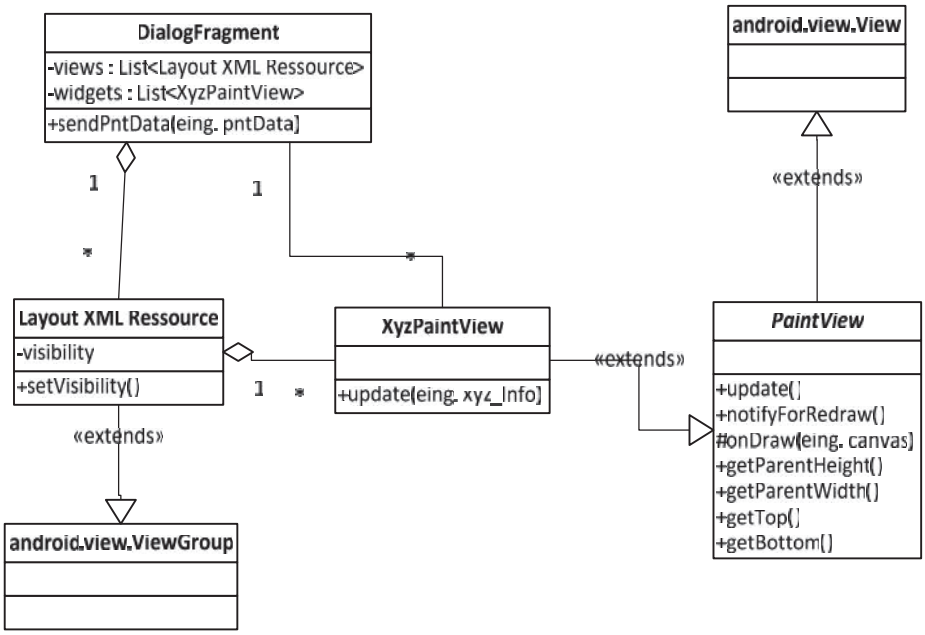


ABBILDUNG 38: DIALOG KLASSENDIAGRAMM

Abbildung 39 skizziert den Aufbau des Dialogbereichs. Das *DialogFragment* setzt sich aus mehreren thematisch verschiedenen Layout-Containern (Satelliten, Alarme, etc.) zusammen, die wiederum mehrere Layouts für dem Thema zugehörige Einzelinformationen beinhalten können. In diesen Layout-Elementen befinden sich die thematisch entsprechenden *PaintViews* und eventuell weitere Interface-Elemente, wie Auswahlmenüs, Schalter, Eingabeformulare, etc. So gehört zur *RaimPaintView* ein Auswahlmenü, mit dem ortsabhängig (See, Hafenbereich, Kai) ein Grenzwert eingestellt werde kann. Das *DialogFragment* bietet Methoden, um die hierarchisch am höchsten stehenden Layout-Container über deren *visibility*-Variable ein- oder auszublenden. So ist es möglich, die Anzeige auf thematisch ausgewählte Informationen zu reduzieren, wie es auch in den Nutzeranforderungen gefordert ist.

Eine Ausnahme zu den bisher beschriebenen grafikbasierten Dialogen zur Darstellung von Informationen, stellt der Dialog zum Eintragen der Zielkoordinaten dar, wie er für den Anwendungsfall Baltic Taucher II (1.4.2) benötigt wird. Dieser besteht nur aus in XML-Ressourcen definierten Schaltern (*Buttons*), Textfeldern und Listen – beinhaltet aber keine *PaintView*, da diese nicht benötigt wird. Abbildung 40 zeigt Nutzerformular für diese Aufgabe. Es ist möglich, Länge und Breite, einen Titel und eine Beschreibung der Position einzutragen. Der Knopf „*Apply*“ überträgt die Position in die Liste und teilt die eingetragenen Werte per Schnittstellenrückruf der *MainActivity* (vgl. 4.3.3) mit, welche die Werte an das *MapFragment* weiterleitet, um es dort als *Marker* auf der *MapView* darzustellen.

Entsprechend löscht der Knopf „*Remv*“ (kurz für remove) einen Listeneintrag und *Marker* in der Karte. „*Cancel*“ bricht einen Eintragsversuch ab und leert die vier Textfelder.

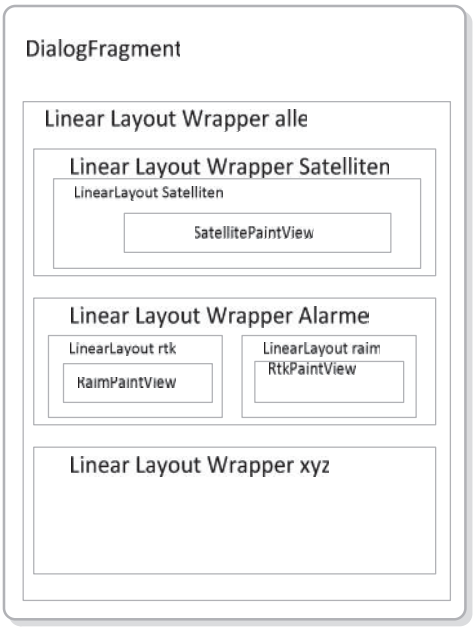


ABBILDUNG 39: LAYOUT ZUSAMMENSETZUNG DIALOGFRAGMENT

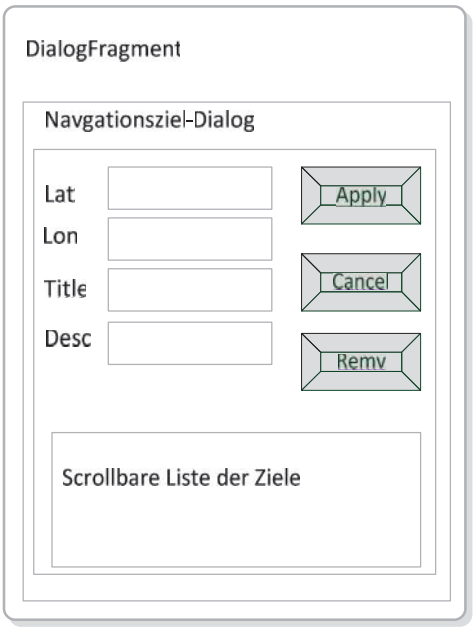


ABBILDUNG 40: DER EINZELNE DIALOG ZUR EINGABE DER ZIELKOORDINATEN

# 5 BESCHREIBUNG UND VALIDIERUNG

In diesem Kapitel soll demonstriert werden, wie der Prototyp des mobilen Systems (mobiles Gerät und mobile Anwendung) die Anforderungen an die mobile Visualisierung umsetzt.

## 5.1 DEMONSTRATION DER MOBILEN VISUALISIERUNG

### *5.1.1 DAS ANZEIGEGERÄT*

Wie in 3.2.3 entworfen, wurde die Anwendung auf einem Anzeigegerät mit kapazitivem Bildschirm implementiert. Für die stationäre Nutzung (3.1.1) befindet sich das Gerät in einem schwenkbaren Ständer, dessen Fuß mit Klettverschluss auf der Brücke der Baltic Taucher II fest installiert werden kann (Abbildung 50). Der Ständer erlaubt das Schwenken des Gerätes in zwei Dimensionen. Das Gerät weist folgende Spezifikation auf: ARMv7 Prozessor, Android Betriebssystem Version 4.4.2, 10,1 Zoll Bildschirmdiagonale.

ABBILDUNG 41: DIE VISUALISIERUNG AUF EINEM TABLET MIT 10,1” BILDDIAGONALE

### *5.1.2 VISUALISIERUNGSLÖSUNG FÜR DEN ANWENDUNGSFALL BALTIC TAUCHER*

Abbildung 42 zeigt die Visualisierung in den beiden Modi Vollbild und geteilter Bildschirm, wie in Kapitel 4.1.1 erläutert. Der Dialogbereich zeigt im rechten Bild das Menü zum Eintragen von Zielkoordinaten, die in der Karte als „X“ zu sehen sind. Eine Informationsblase, die bei Berührung des „X“-Symbols erscheint, stellt die dem Symbol zugehörenden, zuvor im Dialog eingetragenen, Informationen (Koordinaten, Titel, Beschreibung) dar. Die Karte beruht, entsprechend 4.3.5, auf *Osmdroid* mit Vektordatenbasis. Über der Karte liegen eine Schicht für Seezeichen, eine Schicht für das eigene Schiff und der Kartenmaßstab. Das Schiff ist maßstabsbedingt als vereinfachtes Symbol dargestellt. Die

Darstellung als maßstabsgetreuer Umriss ist in Abbildung 44 ersichtlich. Im oberen Bereich der Anzeige befindet sich die Menüleiste, die in 5.1.3 näher beschrieben wird.

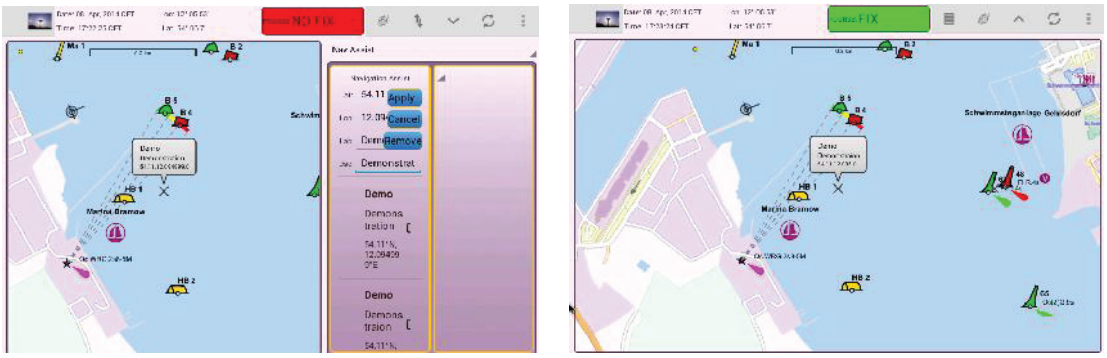


ABBILDUNG 42: KARTE UND DIALOGBEREICH (LINKS), KARTE IM VOLLBILDMODUS (RECHTS)

### 5.1.3 ACTIONBAR MIT MENÜLEISTE UND TITELBEREICH

Abbildung 43 zeigt die Menüleiste mit Titel als Teil der Android-*ActionBar*, wie es in 4.1.1 beschrieben ist. Entsprechend 3.2.3 stellt der Titelbereich Informationen dar, die permanent verfügbar sein müssen und das Menü beinhaltet Funktionen, auf die immer Zugriff möglich sein muss. Die permanenten Informationen sind Datum (Date), Uhrzeit (Time), geographische Länge (Lon) und Breite (Lat) sowie die Qualität der PDGNSS Positionslösung. Wenn kein Prozessierungskanal (1.3.2) mit PDGNSS für die Schiffsdarstellung genutzt wird, zeigt das Feld einen weißen Hintergrund an, wie es in Abbildung 43 ersichtlich ist. In Abbildung 42 sind auch die Farbkodierungen für „kein Fix“ und „Fix“ der PDGNSS-Lösung entsprechend 3.2.3 zu sehen.

Die Menüleiste befindet sich am rechten Ende der *ActionBar*. Sie zeigt, entsprechend 4.1.2, von links nach rechts Symbole für folgende Funktionen: Zentrieren des Schiffes, Vertauschen der Containerinhalte (4.1.1), Vergrößern des linken Containers zum Vollbild (4.1.1), Verbinden mit der PNT-Unit (4.3.4) und Anzeige zusätzlicher Funktionen, worüber man zu den Einstellungen (4.1.3) gelangt.


ABBILDUNG 43: DIE MENÜLEISTE DER ANWENDUNG

### 5.1.4 DAS EIGENE SCHIFF

Abbildung 44 zeigt das eigene Schiff als maßstabsgetreuen Umriss. Gemäß dem Entwurf in 3.3.5 wird der maßstabsgetreue Umriss verwendet, wenn eines der Kriterien zutrifft. Dies ist der Fall, wenn der *MapView*-Zoomlevel größer oder gleich 18 ist. Das rechte der beiden Bilder zeigt das in 3.3.5 vorgeschlagene Popup-Fenster zur Darstellung der wichtigsten Schiffsparameter. In dem Popup-Fenster werden, neben den aus den Einstellungen

hinterlegten Schiffsparametern, die aktuellen Koordinaten sowie der aus den PNT-Daten berechnete Steuerkurs und die Geschwindigkeit in Knoten angezeigt. Es lässt sich durch Berühren des Schiffssymbols ein- und wieder ausblenden.

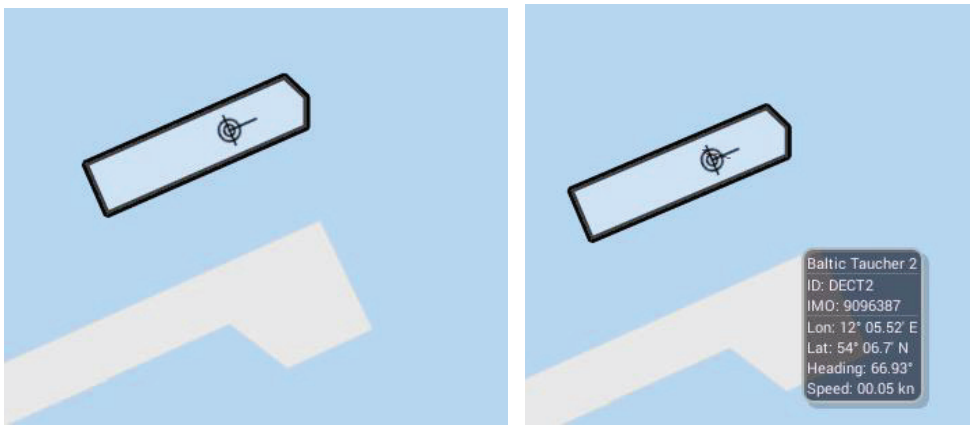


ABBILDUNG 44: DAS EIGENE SCHIFF ALS UMRISS, RECHTS MIT INFORMATIONSFENSTER

### 5.1.5 DIALOGE

Abbildung 45 zeigt weitere Dialoge, die entsprechend 4.3.6 beim jetzigen Stand der Entwicklung im Dialogbereich eingeblendet werden können. In der Abbildung sind die Inhalte beider Fragment-Container, wie in 4.1.1 beschrieben, vertauscht. Die Karte befindet sich im rechten Fenster und die Dialoge im linken.

Der Satelliten-Dialog zeigt links einen typischen Satelliten-Skyplot und rechts ein Diagramm, das die Innovation jedes Satelliten darstellt. Die Überschreitung eines definierten Grenzwertes ist farblich rot kodiert. Der Grenzwert ist in der *SatellitePaintView* derzeit noch fest kodiert, da der Satelliten-Dialog für den Anwendungsfall Baltic Taucher II nicht relevant ist und ein vorerst niedriger Grenzwert zum Testen der Funktionalität sinnvoll ist.

Der Alarm-Dialog visualisiert, sowohl graphisch als auch alphanumerisch, verschiedene Alarmwerte, die Bestandteil der PNT-Informationen sind. Am Alarm-Dialog lässt sich die Schachtelung des Dialogbereichs, bestehend aus Layout-Containern, Layout-Elementen und *PaintViews*, wie in 4.3.6 beschrieben, gut demonstrieren. Der linke der drei Einzeldialoge zeigt in der *PaintView* den RAIM-Wert (7,59075 m) des PNT-Datensatzes alphanumerisch und graphisch als Säule in einem Gefäß, das doppelt so hoch wie ein gesetzter Grenzwert (10,0 m) ist. Wird der Grenzwert überschritten, ändert sich die Farbkodierung der Säule von grün zu rot. Der in 4.3.6 diskutierte Grenzwert lässt sich in einem herunterklappbaren Menü, oberhalb der *PaintView*, einstellen (Abbildung 45 rechts).

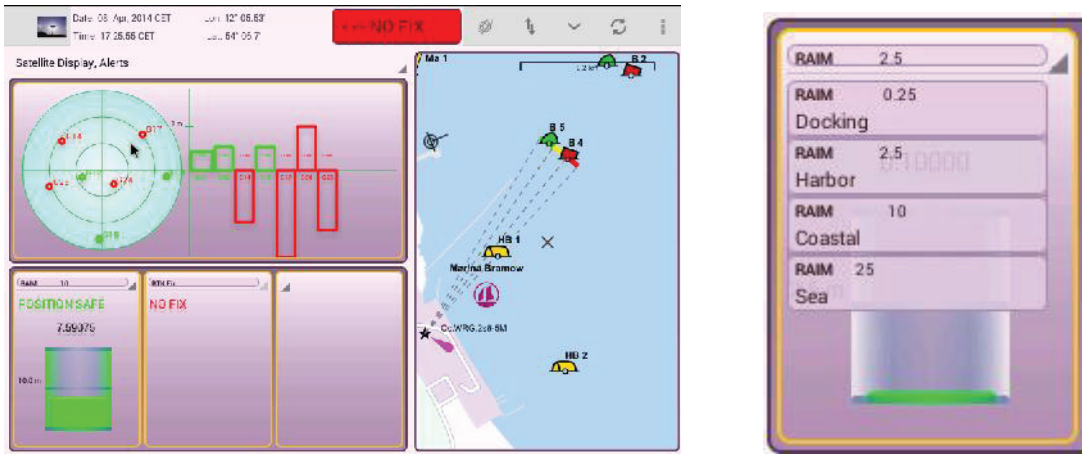


ABBILDUNG 45: WEITERE DIALOGE (SATELLITEN OBEN UND ALARME UNTEN), RECHTS: RAIM

Dem modularen Ansatz folgend, ist es möglich, sowohl dem Dialogbereich vertikal weitere Dialoge, als auch horizontal einem Dialog weitere Einzeldialoge hinzuzufügen. Wie Abbildung 46 links zu entnehmen ist, besteht der gesamte Dialogbereich zum jetzigen Zeitpunkt aus vier thematisch abgrenzbaren Dialogen: (von oben nach unten) Schiffsbewegung, Satelliten, Alarme und Integritätswerte sowie das Dialogmenü zur Eingabe von Zielkoordinaten für den Baltic-Taucher-Anwendungsfall. Die vorhandenen Dialoge lassen sich über ein Auswahlmenü am oberen Rand ein- und ausblenden (rechtes Bild). Die Darstellung des Dialogs für Schiffsbewegungen ist als XML-Ressource schon vorhanden, konnte aber aus Zeitgründen noch nicht um eine *PaintView* ergänzt werden. Es ist vorgesehen, dass in dem Dialog die Roll-, Nick- und Gierwinkel aus den PNT-Daten visualisiert werden.


ABBILDUNG 46: KOMPLETTER DIALOGBEREICH (LINKS) UND DIALOGAUSWAHL (RECHTS)

### 5.1.6 BAHNDARSTELLUNG

Abbildung 47 zeigt die Darstellung von Bahnen, die aus den zurückliegenden Positionen jeweils einer Prozessierungskette (1.3.2) bestehen - bei kleinem (links) und großem Maßstab (Mitte), so wie es in 3.3.5 bzw. 4.3.5 beschrieben wurde. Das rechte Bild zeigt das Kontextmenü aus den Einstellungen der Anwendung, in dem der Nutzer einstellen kann, wo das eigene Schiff positioniert werden soll. Unter einem weiteren Einstellungspunkt lässt sich festlegen, welche Bahnen dargestellt werden sollen.


ABBILDUNG 47: DARSTELLUNG VERSCHIEDENER BAHNEN AUS PROZESSIERTEN POSITIONEN

### 5.1.7 DAS EINSTELLUNGSMENÜ

Abbildung 48 zeigt die in Kategorien zusammengefassten Einstellungen, so wie es in Kapitel 4.1.3 beschrieben wurde. Der sich unter der Kategorie „Settings File“ befindliche Punkt „Import…“ erlaubt das Importieren einer Datei mit Einstellungen, die sich auf dem Anzeigegerät befindet und einem Format gemäß Abbildung 28 entspricht. Die Zahlenwerte, die dem Schlüsselwert „SHIP_CONTOUR“ folgen, sind dreidimensionale Koordinaten des Schiffes, die relativ zum Referenzpunkt aufgemessen wurden und aus denen der maßstabsgetreue Umriss des Schiffes in dem entsprechenden *Overlay* der Kartendarstellung konstruiert wird (vgl. 4.3.5). Diese Koordinaten werden als JSON-String unter der Kategorie „Ship“ gespeichert, da Android in *SharedPreference*-Objekten nur die Speicherung von wenigen Datentypen[19] erlaubt. Dies macht es erforderlich, die Koordinaten des Umriss von (zum Darstellen des Umriss) und nach JSON (beim Importieren) zu transformieren. Beides ist vertretbar, da das Importieren von Einstellungen selten ist und das Transformieren des JSON-Objekts in einen darstellbaren Pfad im *OwnShipOverlay* (vgl. 4.3.5 unten bzw. Abbildung 37) nur einmal im Konstruktor der Klasse durchgeführt wird.

[19] Erlaubte *SharedPreference* Datentypen sind Boolean, Float, Int, Long und String (http://developer.android.com/guide/topics/ui/settings.html#Overview )


ABBILDUNG 48: EINSTELLUNGEN (LINKS SCHIFF, MITTE KARTE, RECHTS IMPORTIEREN UND VERBINDUNG)

## 5.2 TESTSZENARIEN UND ERKENNTNISSE

Die Klassen und Komponenten der mobilen Visualisierung wurden bereits während der Implementierung sukzessive im Labor getestet[20]. Die Visualisierung, so wie sie in 5.1 beschrieben wurde, entspricht weitestgehend den festgelegten Anforderungen. Da der Prototyp auf einem proprietären Gerät implementiert wurde, ist das Anzeigegerät nicht robust gegenüber äußeren Einflüssen wie Wasser oder einem Herunterfallen. Es wurde aus naheliegenden Gründen darauf verzichtet, diese Kriterien zu testen. Durch den kapazitiven Bildschirm des Anzeigegerätes, ist die Anwendung für den Einsatz auf See eigentlich ungeeignet – da sie, wie im Folgenden noch dargelegt wird, nur als Brückenanzeige zum Einsatz kommt, fällt dieser kritische Punkt nicht in Gewicht.

Im Labor wurde die Anforderung getestet, dass das Anzeigegerät während der angedachten Aufgabe nicht ausfallen darf. Diese Tests geschahen bereits in Vorbereitung auf eine anstehende Messkampagne auf See. Als mögliche Fehler wurden ein Ausfall der Datenübertragung, eine Benutzeroberfläche, die nicht mehr auf Nutzereingaben reagiert und ein Stromausfall auf Seiten des Anzeigegerätes ausgemacht. Ein Eintreten dieser Fehler würde beim Benutzer jeweils den Eindruck erwecken, dass die Anwendung ausgefallen ist. Für diesen Testfall wurden aufgezeichnete PNT-Daten von der PNT-Unit abgespielt und das Anzeigegerät, das eine Netzverbindung über Kabel besaß, hat die Daten dargestellt. Im Anschluss sollte überprüft werden, ob die Datenverbindung noch bestand und ob die dargestellten Daten noch mit denen der PNT-Unit synchron waren, ob die Benutzeroberfläche noch auf Eingaben reagiert und wie sich der Energiehaushalt darstellt.

Im Vorfeld zeigte sich bereits, dass die Stromversorgung des Gerätes sehr kritisch ist. Mit gedimmtem Bildschirm und bei dauerhafter Netzverbindung über USB-Kabel war es möglich, Energieverlust und –gewinn in Waage zu halten. Bei mobiler Nutzung und Akkubetrieb war kein durchgängiger Betrieb der Anwendung möglich. Dies führte zu der Notwendigkeit, den Prototypen nur mit kabelgebundener Stromversorgung in einem realen Umfeld zu testen. Die

[20] JUNIT-basierte Testschnittstelle von Android:
http://developer.android.com/tools/testing/testing_android.html

Anwendung konnte auch nach mehrtägiger Datenverbindung in einem Hintergrundthread und bei permanenter Aktualisierung des „UI-Threads“ noch auf Nutzereingaben reagieren. Das bei Android-Anwendungen kritische ANR [21] (ability not responding) ist ausgeblieben. Die dargestellten Daten entsprachen auch noch den momentan prozessierten. Der unter Dauerbetrieb permanent beleuchtete Bildschirm konnte den Akkustand nicht senken.

## 5.3 Messkampagne mit der Baltic Taucher II

Während einer mehrtätigen Messkampagne, deren Ziel es war, PNT-Daten aufzuzeichnen, wurde die mobile Visualisierung auf der Baltic Taucher II (Abbildung 49) in einem realen Umfeld getestet und den Händen der Besatzung überlassen. Die PNT-Unit (die Datenquelle) war während der Messkampagne auf der Brücke unterhalb der bestehenden Anzeigen (siehe Abbildung 50) installiert. Da zuvor bereits bekannt war, dass der Prototyp der Anwendung zu diesem Zeitpunkt eine dauerhafte Stromverbindung benötigt, wurde die Anwendung mit einem Ständer auf der Brücke so befestigt, dass die bestehenden Geräte nicht beeinträchtigt wurden. Die Anforderung, dass das Anzeigegerät mobil verwendbar sein soll, konnte so nicht mehr erfüllt werden, dafür konnte der Anwendungsfall getestet werden.

ABBILDUNG 49: DIE BALTIC TAUCHER II

ABBILDUNG 50: INSTALLATION DER VISUALISIERUNG AUF DER BRÜCKE (MITTE UND RECHTS; LINKS: VORHER)

[21] http://developer.android.com/training/articles/perf-anr.html

ABBILDUNG 51: DIE VISUALISIERUNG AM HECK DES SCHIFFES

Die Reichweite der Datenverbindung wurde - unabhängig der Messkampagne - als „Begehung des Schiffes“ getestet. Dabei zeigte sich, dass die Datenverbindung zur Datenquelle auf der Brücke sowohl unter Deck als auch am Heck des Schiffes stabil blieb. Abbildung 51 zeigt die Anwendung, abgelegt am Heck der Baltic Taucher II, voll funktionsfähig.

Eine Umfrage zur Praxistauglichkeit der Anwendung war für den vorliegenden Anwendungsfall nicht sinnvoll. Als Brückengerät hatte der Kapitän der Baltic Taucher II die Visualisierung aber verwendet und wurde auch in die Verwendung eingewiesen und konnte im Anschluss der Messkampagne eine Einschätzung dazu abgeben. Wie im Anwendungsfall vorgesehen, wurde die Visualisierung zur hochgenauen Positionierung des Schiffes verwendet. Die Eingabe von Zielkoordinaten und der Vergleich der prozessierten und in der Titelleiste angezeigten Position mit der Position im ECDIS des Schiffes gestalteten sich als schwierig, da die Formatierung der Koordinaten unterschiedlich war. Das gewünschte Format ist [ 00° 00.00‘ ]. Dieser Verbesserungswunsch wurde bei einem späteren Besuch auf dem Schiff behoben.

Der fehlende Nachtmodus, der aus Zeitgründen nicht umgesetzt werden konnte, hatte auf den Anwendungsfall keine Auswirkungen, es wurde aber seitens der Besatzung als sehr störend beschrieben. Für die zukünftige Umsetzung des Nachtmodus ist vorgesehen, dass ein neues XML-basiertes „*rendertheme*“, das helle Informationen auf dunklem Hintergrund erzeugt, für die Kartendarstellung (vgl. 4.3.5) erstellt wird und für die Benutzeroberfläche auch neue entsprechend dunkle XML-Ressourcen erstellt werden. Ebenfalls nicht umgesetzt wurden Mittel, die den Nutzer darauf hinweisen, dass die dargestellten Informationen nicht den prozessierten PNT-Daten entsprechen. Die Idee hierzu ist, die verstrichene Zeit der Uhr des Anzeigegerätes seit der letzten Aktualisierung des PNT-Datensatzes zu überwachen und ab einem gewissen Grenzwert einen Alarmhinweis für den Nutzer auszugeben. Zeitbedingt war es auch nicht mehr möglich, Seezeichen in der Kartendarstellung als lokale Quelle umzusetzen, wodurch das Online-basierte Overlay für die Seezeichendarstellung leer blieb.

Positiv war, dass die Anwendung die PNT-Daten korrekt dargestellt hat (soweit der Kapitän zugegen war und das auch beurteilen konnte) und dass sie bei gelegentlichen Ausfällen seitens der PNT-Unit immer eine neue Verbindung versucht und erzeugt hat und den Verbindungsversuch auch visualisiert hat. Auch die Kombination einer Android-Anwendung

mit der Anlehnung der Darstellungsform an bekannte Brückenanzeigen wurde hinsichtlich der intuitiven Informationsdarstellung und -bedienung als positiv bewertet.

# 6 FAZIT UND AUSBLICK

Das Ziel der Arbeit war es, hochpräzise Navigationsinformationen, die von einem neuartigen Navigationssystem stammen, in einer mobilen Anwendung zu visualisieren; mit dem Ziel, dass eine Nutzerin oder ein Nutzer an Bord eines Schiffes mobil mit diesen Informationen arbeiten kann. Dieses Ziel konnte anhand der Entwicklung und Implementierung eines prototypischen mobilen Systems erreicht und an Bord eines Schiffes getestet werden – mit der gegebenen Einschränkung, dass das Endgerät, auf dem die mobile Anwendung implementiert wurde, vorerst nur als stationärer Bildschirm auf der Brücke verwendet werden konnte. Das Endgerät war nicht hochseetauglich und auf Grund von Beschränkungen des Akkus nur bedingt mobil verwendbar. Für eine mobile Anwendung, die unter Hochseebedingungen für die Dauer einer Aufgabe robust und ausfallsicher arbeiten soll, ist es essentiell, die Fragen nach der Energieversorgung und des Schutzes des Anzeigegerätes gegenüber den äußeren Bedingungen, im Vorfeld zu klären. Die Wahl von Android als Entwicklungsplattform hat sich im Nachhinein nicht als nachteilig erwiesen, konnte doch durch das Zurückgreifen auf Bibliotheken von Dritten eine Kartendarstellung auf Basis einer lokalen Vektordatei umgesetzt werden. Dies ist angesichts des Einsatzes auf einem Schiff ebenfalls essentiell und die Umsetzung der lokalen Kartenerzeugung (inklusive Seezeichen) stellt den dritten Punkt dar, der im Vorfeld geklärt werden sollte, wenn die Entscheidung für eine Entwicklungsplattform getroffen werden muss.

Der Umstand, dass keine Standards für derartige mobile Anwendungen vorhanden sind und vorhandene Standards nur bedingt auf eine mobile Anwendung in der vorgestellten Form zutreffen, machte es notwendig, Kriterien entsprechend der Nutzungsanforderungen zu entwickeln. Die Visualisierung wurde weitestgehend so entwickelt, wie es ein möglicher Standard vorschreiben könnte. Anstatt mit Android wäre die Implementierung auch auf einer der anderen mobilen Plattformen oder auch auf komplett anderem Wege möglich gewesen. Ein Urteil dazu kann an dieser Stelle nicht getroffen werden und die Anwendung soll auch nicht in einem der „Stores“ vertrieben werden. Der Vorteil von Android war letztendlich, dass sich das entwickelte Konzept unproblematisch, kostengünstig und in vertretbarer Zeit als Prototyp implementieren ließ und die prototypische Anwendung die Navigationsinformationen mobil darstellen kann. Die Anwendung läuft auch, was in der Arbeit nicht dokumentiert ist, auf kleinen Geräten (Handys). Auch wenn ein(e) Nutzer(in) Abstriche in der Qualität und Erkennbarkeit der dargestellten Informationen hinnehmen muss, ist die angedachte Funktionalität komplett vorhanden. Zukünftig wäre es interessant, die Anwendung auf übergroßen Touchscreens zu testen.

Als Schlussbemerkung sei nochmal zusammengefasst, dass es, Anwendungsfall-bezogen, einen Bedarf an der Verfügbarkeit von Navigationsinformationen außerhalb der Brücke eines Schiffes gibt und (auch wenn die vorliegende Anwendung nur als Brückengerät zum Einsatz kam) eine mobile Anwendung zur Darstellung dieser Informationen diesen Bedarf erfüllen kann.

# LITERATURVERZEICHNIS


[1] DLR: Institut für Kommunikation und Navigation, „Projekt Maritime Verkehrstechnik: E-Navigation Integrität," Neustrelitz, 2014.

[2] H. Kahmen, Angewandte Geodäsie: Vermessungskunde, Berlin: de Gruyter, 2006.

[3] H. Hecht, B. Berking, G. B. Büttgenbach und M. Jonas, Die elektronische Seekarte: Grundlagen, Möglichkeiten und Grenzen eines neues Navigationssystems, Herbert Wichmann, 1999.

[4] G. Hake, D. Grünreich und L. Meng, Kartographie: Visualisierung raum-zeitlicher Informationen, Berlin: de Gruyter, 2002.

[5] N. Bartelme, Geoinformatik: Modelle Strukturen Funktionen, Berlin Heidelberg: Springer, 2005.

[6] Wikipedia, „Mobilität," 08 09 2014. [Online]. Available: http://de.wikipedia.org/wiki/Mobilität. [Zugriff am 21 10 2014].

[7] D. Dr. Ing. Krannich, Mobile System Design, Norderstedt: Books on Demand GmbH, 2010.

[8] Bundesamt für Seeschifffahrt und Hydrographie, „ECDIS Faltblatt," 2012. [Online]. Available: http://www.bsh.de/de/Produkte/Infomaterial/Elektronische_Seekarten/ECDIS-Faltblatt.pdf. [Zugriff am 09 2014].

[9] Bundesamt für Seeschifffahrt und Hydrographie, „ECDIS," Bundesamt für Seeschifffahrt und Hydrographie, 06 01 2011. [Online]. Available: http://www.bsh.de/de/Produkte/Karten/Elektronische_Seekarten/356.jsp. [Zugriff am August 2014].

[10] International Maritime Organization (IMO), „E-navigation," IMO, 2014. [Online]. Available: http://www.imo.org/OurWork/Safety/Navigation/Pages/eNavigation.aspx. [Zugriff am 18 10 2014].

[11] IALA AISM, „e-Navigation Frequently Asked Questions," IALA AISM, 2014. [Online]. Available: http://www.iala-aism.org/about/faqs/enav.html. [Zugriff am 18 10 2014].

[12] DLR Institute of Communications and Navigation, „Technical Note: PNT Unit Softwareentwicklung," Neustrelitz, 2013.

[13] S. Gleason und D. Gebre-Egziabher, GNSS: Applications and methods, Norwood, MA: Artech House, 2009.

[14] V. Teoh, „HDTVtest," 22 11 2012. [Online]. Available: http://www.hdtvtest.co.uk/news/ipad-mini-nexus-7-201211222374.htm. [Zugriff am 10 10

2014].

[15] SAM Electronics, „NACOS Platinum: Operating Instructions,“ Hamburg, 2011.

[16] Crow, David, Zeichen, München: Stiebner Verlag, 2012 (gefunden bei: http://books.google.de/books?id=3psIxEY3RAQC&printsec=frontcover).

[17] Android Developer, „Android Developers Dashboard,“ 09 2014. [Online]. Available: https://developer.android.com/about/dashboards/index.html. [Zugriff am 09 2014].

[18] salsoft, „How To Render MapsForge tiles in OSMDroid,“ Salida Software, 2014. [Online]. Available: http://www.salidasoftware.com/how-to-render-mapsforge-tiles-in-osmdroid/. [Zugriff am 07 2014].

[19] Bundesamt für Seeschifffahrt und Hydrographie, „Elektronische Seekarten,“ 2012. [Online]. Available: http://www.bsh.de/de/Produkte/Infomaterial/Elektronische_Seekarten/ECDIS-Faltblatt.pdf. [Zugriff am August 2014].

[20] Golem, „golem.de,“ 01 08 2014. [Online]. Available: http://www.golem.de/news/mobile-betriebssysteme-android-laeuft-auf-fast-85-prozent-aller-smartphones-1408-108290.html. [Zugriff am 20 10 2014].

# VERWENDETE NORMEN UND STANDARDS

[MSC.191(79)] Performance standards for the presentation of navigation-related information on shipborne navigational displays (adopted on 6 December 2004)

[SN/Circ.243] Guidelines for the presentation of navigation-related symbols, terms and abbreviations (15 December 2004)

[IEC 62288] Maritime navigation and radiocommunication equipment and systems

[IEC 62388] Navigations- und Funkkommunikationsgeräte und -systeme für die Seeschifffahrt – Radar für Schiffe – Leistungsanforderungen, Prüfverfahren und geforderte Prüfergebnisse (IEC 62388:2007)

[IMO A.915(22)] Revised maritime policy and requirements for a future global navigation satellite system (GNSS)

[DIN 33402-2:2007] Ergonomie - Körpermaße des Menschen - Teil 2: Werte