



# **Physikalische Testfeld-Architektur für die Unterstützung der Entwicklung von automatisierten Schiffsführungssystemen**

Von der Fakultät II für Informatik, Wirtschafts- und Rechtswissenschaften der Carl von  
Ossietzky Universität Oldenburg zur Erlangung des Grades und Titels eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

angenommene Dissertation vorgelegt von

**Herrn Marius Brinkmann,**

**M. Sc.,**

geboren am 11. April 1988 in Wilhelmshaven

Gutachter:

Prof. Dr.-Ing. Axel Hahn

Prof. Dr.-Ing. Frank Köster

Tag der Disputation: 24. Oktober 2018

# Danksagung

Zahlreichen Menschen gebührt Dank für die Unterstützung und Begleitung während meiner Zeit an der Universität Oldenburg, in der ich diese Dissertation zustande bringen durfte.

Meinem Doktorvater Herrn Prof. Dr.-Ing. Axel Hahn bringe ich meine starke Dankbarkeit entgegen. Er schaffte stets Freiräume, um eine konstruktive Zusammenarbeit und beharrliche Betreuung zu ermöglichen. Mich faszinieren sein innerer Ansporn und ich schätze seinen Sinn für den Menschen, der neben dem fachlichen und methodischen Austausch zu freudigen Momenten führte. Die Fahrten mit der ZUSE und meine Zeit in seiner Abteilung vergesse ich nicht. Axel verdient den Namen Doktorvater.

Weiter danke ich Herrn Prof. Dr.-Ing. Frank Köster für die Übernahme des Zweitgutachters im Promotionsverfahren. Er fasste den Rahmen nicht zu eng und ermöglichte mir somit mich bei der Erarbeitung der Dissertation entfalten zu können. Für den fachlichen Austausch und das Hinterfragen wichtiger Aspekte bin ich dankbar. Herrn Prof. Dr.-Ing. Wolfgang Nebel danke ich für die Übernahme des Vorsitzes der Prüfungskommission und die angenehme berufliche Zusammenarbeit, die erfreulicherweise in der Endphase meiner Promotion begann. Herrn Dr. Ingo Stierand danke ich für sein Einbringen als Mitglied der Prüfungskommission.

Ebenfalls danken möchte ich meinen Kolleginnen und Kollegen der Abteilung Systemanalyse und -optimierung der Universität Oldenburg und der Gruppe Kooperierende Mobile Systeme des OFFIS e. V. Neben vielen fachlichen Diskussionen zum Voranbringen meiner Arbeiten war das Schließen von Freundschaften möglich. Ich danke Manuela Wüstefeld als eine gute Seele der Abteilung nicht nur wegen der vielen leckeren Kuchen, sondern weil ihr das Wohl ihrer Kolleginnen und Kollegen am Wichtigsten ist. Den Studentinnen und Studenten danke ich ganz herzlich, die durch Abschlussarbeiten, Tätigkeiten als studentische beziehungsweise wissenschaftliche Hilfskräfte und der Teilnahme an den Projektgruppen MATE durch die Implementierungen zu meiner Arbeit beitrugen. Besonders hervorheben möchte ich Lea Bollerslev, Julius Möller und Hilko Wiards.

Die langjährige Arbeit an einer Dissertation ist von produktiven und schwierigen Zeiten geprägt, so auch meine. Eine wichtige Stütze leistete meine Familie mit ihrer Unterstützung während des Anfertigens dieser Dissertation und dem Weg dorthin. Meiner Familie spreche ich besonderen Dank aus, denn meine Mutter Monika und mein Vater Hartmut boten mir ein wunderbares Zuhause und gaben mir immerzu Rückenwind. Ohne den bedingungslosen Rückhalt hätte ich all dies nicht erreicht – wie schön, dass es Euch gibt.

Zu guter Letzt danke ich von Herzen meiner lieben Verlobten Evelyn, die mich während aller Höhen und Tiefen von jeher durch ihre Liebe und helfenden Worte innerlich motivierte und stärkte. Sie gab mir den Freiraum zum Erarbeiten dieser Arbeit und förderte durch ihre liebevolle Fürsorge meine persönliche Entwicklung. Mit Dir an meiner Seite habe ich fort-dauernd Sonnenschein im Gesicht: Herzlichen Dank, ich liebe dich!



# Zusammenfassung

Die Führung eines Schiffes erfolgt zunehmend automatisiert durch die Unterstützung moderner schiffs- und küstenseitiger Systeme, welche bei der sicheren Führungsaufgabe assistieren oder diese eigenständig realisieren. Die Entwicklung dieser automatisierten Schiffsführungssysteme ist geprägt durch vernetzte und lose gekoppelte Systeme von verschiedenen Herstellern. Dies führt zu einem stetig wachsenden Anteil von komplexen Systementwicklungen, die im Rahmen des maritimen Transports sicherheitskritische Funktionen als cyber-physische Systeme erfüllen. Für die Systementwicklung im Hinblick auf die korrekte Funktionalität der automatisierten Schiffsführungssysteme erfolgt die Anwendung intensiver Verifikations- und Validierungsmethoden. Ein Testfeld trägt dazu bei, die Verifikation und Validierung (V+V) während des gesamten Systementwicklungsprozesses zu fördern und ermöglicht die Erprobung des zu testenden Systems in einer simulativen oder physikalischen Umgebung. Die Heterogenität der zu testenden Systeme und umfangreiche Anforderungen durch die Systementwicklungsmethodik stellen die Gestaltung aktueller maritimer Testaufbauten vor Herausforderungen. Diese sind individuell für spezifizierte Testszenarien ausgeprägt und nicht nachhaltig konstruiert, denn die Gestaltung folgt keiner strukturierten Methode.

Diese Forschungsarbeit entwickelt eine Systemarchitektur für ein nachhaltiges und wiederverwendbares physikalisches Testfeld, das sich an den etablierten Methoden der Systementwicklung orientiert. Es unterstützt die Methodik des Systems Engineering, um die Validierung eines automatisierten Schiffsführungssystems und die Verifikation der Umsetzung gemäß den technologischen Entwürfen zu ermöglichen. Der Ansatz unterstützt die V+V-Methoden während verschiedener Phasen des Systementwicklungszyklus und berücksichtigt die normativen Anforderungen der maritimen Domäne. Eine Implementierung des physikalischen Testfeldes LABSKAUS im Rahmen der maritimen Referenzplattform eMIR überführt die in dieser Arbeit entwickelte Systemarchitektur eines physikalischen Testfeldes in die Praxis. Mittels repräsentativer Einsatzszenarien von LABSKAUS für die Überprüfung der Anwendbarkeit und die Verifikation der Anforderungserfüllung erfolgt eine Evaluation des Lösungsansatzes.

Die vorgestellte systematische Systemarchitekturentwicklung für ein physikalisches Testfeld reduziert Integrationsaufwände für zu testende Systeme und fördert die Wiederverwendbarkeit des Testfeldes. Dies führt zu einer effektiven Erprobung der komplexen Systeme und Komponenten mit niedrigeren Systementwicklungskosten.



## Abstract

The navigation of a vessel is increasingly automated by the support of modern technology aboard a vessel and on shore, which assist people in safe navigation or perform the navigation and operation out independently. The development of these automated vessel navigation systems is characterised by interlinked and loosely coupled systems from various manufacturers. This leads to a steadily growing amount of complex systems which fulfil safety-critical functions as a cyber-physical system in the context of maritime transportation. For system development considering the correct functionality of automated vessel navigation systems, intensive verification and validation methods are applied. A test bed helps to promote verification and validation (V+V) throughout the entire system development process and enables testing of the system under test in a simulative or physical environment. The heterogeneity of the systems to be tested and the extensive requirements of the system development methodology for these innovative systems result in challenges for the design of current maritime test setups. These are individually designed for specified test scenarios and not sustainably constructed, because their design does not follow a structured method.

This research work develops a system architecture for a sustainable and reusable physical test bed, which is oriented to the established methods of system development. This supports the methodology of systems engineering to enable validation of an automated ship navigation system and verification of its implementation according to the technological design. The approach supports the V+V methods during different phases of the system development cycle and considers the normative requirements of the maritime domain. An implementation of the physical test bed LABSKAUS within the context of the maritime reference platform eMIR transfers the system architecture of a physical test bed developed in this work into practice. Representative application scenarios of LABSKAUS are carried out for the examination of the applicability and verification of the requirements fulfilment which permits an evaluation of the approach.

The presented systematically developed system architecture for a physical test bed reduces integration efforts for systems to be tested and supports the reusability of the test bed. This leads to effective testing of complex systems and components with lower system development costs.



# Inhaltsverzeichnis

Zusammenfassung .....	I
Abstract .....	III
Inhaltsverzeichnis .....	V
Abkürzungen und Akronyme .....	VII
1 Einleitung .....	1
1.1 Systementwicklung für maritime Technologien .....	1
1.2 Problematik .....	4
1.3 Zielsetzung und wissenschaftlicher Beitrag .....	7
1.4 Vorgehen und Struktur der Arbeit .....	10
2 Verifikation und Validierung von automatisierten Schiffsführungssystemen .....	13
2.1 Automatisierte Schiffsführungssysteme .....	13
2.1.1 Hydrodynamische Umgebung .....	14
2.1.2 Drei-Schichten-Modell eines automatisierten Schiffsführungssystems .....	15
2.1.3 Datenaustausch innerhalb des Drei-Schichten-Modells .....	18
2.1.4 Interoperabilität automatisierter Schiffsführungssysteme .....	21
2.2 Systementwicklungsmethodik .....	24
2.2.1 Entwicklung maritimer Automatisierungssysteme .....	24
2.2.2 V-Modell für die Systementwicklung .....	26
2.2.3 Statische und dynamische Verifikation und Validierung .....	28
2.2.4 Testen im Open-Loop- und Closed-Loop-Verfahren .....	31
2.2.5 Unterstützung der V+V durch Testfelder .....	33
2.2.6 Technologien und Konzepte eines Testfeldes .....	34
2.3 Anforderungen an ein physikalisches Testfeld .....	36
2.3.1 Anforderungen des zu testenden Systems .....	36
2.3.2 Anforderungen der Systementwicklungsmethodik .....	38
2.3.3 Anforderungen der Gestaltung .....	41
2.4 Zusammenfassung der Grundlagen .....	43
3 Verwandte Arbeiten .....	45
3.1 Maritime Testaufbauten .....	45
3.2 Luftfahrt- und Kraftfahrt-Testfelder .....	51
3.3 Untersuchungsergebnisse .....	54
3.4 Zusammenfassung und Handlungsbedarf .....	60
4 Systemarchitektur eines physikalischen Testfeldes .....	62
4.1 Anwendungsfall: Führung eines hochautomatisierten Schiffes .....	62
4.2 Systematisierung des Testfeldes für automatisierte Schiffsführungssysteme .....	65
4.3 Testfeld-Architektur .....	68
4.3.1 Einheitliches Datenmodell gemäß S-100 .....	69
4.3.2 Publish-Subscribe nachrichtenorientierte Middleware .....	72
4.3.3 Integration von Testfeld-Komponenten .....	77
4.3.4 Monitoring .....	79
4.3.5 Physikalische Verteilung .....	80
4.4 Steuerung des physikalischen Testfeldes .....	82
4.4.1 Fehlerinjektion durch Manipulation .....	83
4.4.2 Integration des virtuellen Testfeldes .....	85
4.5 Polymorphe Schnittstelle .....	89
4.5.1 Transformationsmodul für Datenströme .....	91

4.5.2	Semantische Annotation mit dem S-100-Referenzmodell .....	94
4.5.3	Transformationsoperatoren des Kommunikationsverhaltens .....	97
4.6	Zusammenfassung der Testfeld-Architektur .....	99
5	LABSKAUS – Labor für sicherheitskritische Analysen auf See .....	101
5.1	Testfeld-Komponenten .....	101
5.2	Nachrichtenorientierte Middleware mit RabbitMQ .....	104
5.3	Einheitliches S-100 Datenmodell mit EMF .....	105
5.4	Polymorphe Schnittstelle .....	106
5.5	Simulations-Adapter.....	110
5.6	Zusammenfassung der Implementierung .....	111
6	Evaluation der entwickelten physikalischen Testfeld-Architektur.....	113
6.1	Leistungsdatenerhebung von LABSKAUS .....	114
6.1.1	Latenz.....	115
6.1.2	Datenübertragungsrate.....	119
6.2	Explorative Datenerhebung für Szenarien-Generierung .....	123
6.2.1	Evaluationsziel.....	123
6.2.2	Durchführung .....	125
6.2.3	Ergebnisse des Einsatzes für explorative Datenerhebung .....	128
6.3	Open-Loop-Test eines Kollisionsvermeidungssystems .....	130
6.3.1	Zielsetzung.....	131
6.3.2	Durchführung .....	132
6.3.3	Ergebnisse des Einsatzes für Open-Loop-Tests .....	137
6.4	Closed-Loop-Test eines hochautomatisierten Schiffsführungssystems .....	138
6.4.1	Zielsetzung.....	140
6.4.2	Durchführung .....	142
6.4.3	Ergebnisse des Einsatzes für Closed-Loop-Tests.....	151
6.5	Anforderungserfüllung und Zielabdeckung.....	154
7	Zusammenfassung und Ausblick .....	157
7.1	Ergebnisse und Forschungsbeitrag .....	157
7.2	Anknüpfender Forschungsbedarf.....	160
	Literatur.....	163
	Abbildungen.....	179
	Tabellen.....	182
	Anhang.....	183

# Abkürzungen und Akronyme

AIS	Automatic Identification System
ASTERIX	All Purpose Structured Eurocontrol Surveillance Information Exchange
AMQP	Advanced Message Queuing Protocol
ASCII	American Standard Code for Information Interchange
BPEL	Business Process Execution Language
CAN	Controller Area Network
CPS	Cyber-physisches System
CORBA	Common Object Request Broker Architecture
DGPS	Differential Global Positioning Systems
DLL	Dynamic Link Library
DOF	Degree of Freedom
ECDIS	Electronic Chart Display and Information System
EMF	Eclipse Modeling Framework
EPD	e-Navigation Prototype Display
FARM	Fault, Activation, Readout, Measure
FMI	Functional Mock-Up Interface
HIL	Hardware in the Loop
IALA	International Association of Marine Aids to Navigation and Lighthouse Authorities
IEC	Internationale Elektrotechnische Kommission
IEEE	Institute of Electrical and Electronics Engineers
IHO	International Hydrographic Organisation
IKT	Informations- und Kommunikationstechnik
IMO	Internationalen Meeresorganisation
IMU	Inertial Measurement Unit
ISO	Internationale Organisation für Normung
IVEF	Inter VTS Exchange Format
MAF	Maritime Architecture Framework
MIL	Model in the Loop
MMSI	Maritime Mobile Service Identity
MTCAS	Maritime Traffic Collision Avoidance System
NMEA	National Marine Electronics Association
NTP	Network Time Protocol
OMG	Object Management Group
PGN	Parameter Gruppennummer
PNT	Positionierung, Navigation und Timing
REST	Representational State Transfer
RPC	Remote Procedure Call
SIL	Software in the Loop
SIP	Strategy Implementation Plan

SOAP	Simple Object Access Protocol
SOLAS	International Convention for the Safety of Life at Sea
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UKW	Ultrakurzwelle
UML	Unified Modeling Language
V+V	Verifikation und Validierung
V2X	Vehicle-to-Everything
VTs	Vessel Traffic Service
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformation

# 1 Einleitung

Strenge Umweltauflagen wie das im Juni 2016 von 195 Ländern geschlossene Pariser Abkommen zum Klimaschutz [Euro16] oder Anforderungen an die Systementwicklung und funktionale Sicherheit durch neue Technologien [HuBH00] verändern die maritime Industrie in politischen und technologischen Aspekten seit Jahren. Daher spielt die maritime Industrie eine Schlüsselrolle für die Automatisierung des zukünftigen Transportierens, um die Schifffahrt effizienter, sicherer und nachhaltiger zu gestalten [HBFF16]. Das Deutsche Institut für Normung definiert Automatisierung als „das Ausrüsten einer Einrichtung, sodass sie ganz oder teilweise ohne Mitwirkung des Menschen bestimmungsgemäß arbeitet“ [Din17]. Diese fortschreitende Digitalisierung resultiert in einer zunehmenden Vernetzung heterogener maritimer Technologien und Systemverbände unterschiedlicher Hersteller, um die komplexen und häufig sicherheitskritischen Transportaufgaben in automatisierter Art und Weise durchzuführen [FoPN17, S.5]. Bezugnehmend auf diese technologischen Entwicklungen wandelt sich die Entwicklung dieser Systeme zunehmend von der Gestaltung elektronischer und mechanischer Komponenten zu einem stetig wachsenden Anteil von Softwareentwicklungen [BrKr08].

## 1.1 Systementwicklung für maritime Technologien

Das von der Bundesregierung initiierte Forschungsprogramm „Maritime Technologien der nächsten Generation“ identifizierte die Förderungsnotwendigkeit der Entwicklung maritimer Technologien als strategisch bedeutsam für die gesamte maritime Wirtschaft. Insbesondere die Entwicklung von Zukunftstechnologien der maritimen Sicherheit und Echtzeitdienste für automatisierte Schiffsführungssysteme adressieren die zentrale Herausforderung der Unterstützung entsprechender Systementwicklungen [Bmwi17, S.25f.]. Automatisierungs- beziehungsweise Assistenzsysteme für die Schiffsführung existieren beispielsweise in Form neuartiger Autopiloten, welche die automatische Steuerung eines Schiffes unterstützen oder vollständig übernehmen. Diese Automatisierungssysteme erfordern die Fähigkeit komplexe Umweltsituationen zu erfassen, zu interpretieren und in Aktionen zu überführen, um dynamischen und statischen Hindernissen, häufig in sicherheitskritischen Situationen, auszuweichen und möglichen Störungen des Transports entgegenzuwirken [AbFH15]. Derartige Systeme setzen neuartige Lösungsansätze ein wie Verfahren des Maschinenlernens aus dem Bereich der künstlichen Intelligenz. Mit diesen aussichtsreichen Verfahren gehen Herausforderungen einher wie das nichtdeterministische Verhalten aufgrund der großen Menge möglicher Eingaben und der Komplexität des Datenverarbeitungsprozesses [DaKa17].

**Systementwicklung.** Die Internationale Organisation für Normung (ISO) bietet Prinzipien für die Entwicklung dieser programmierbaren elektronischen Systeme, welche unter ande-

rem Sicherheitsfunktionen erfüllen. Die ISO 17894 (engl. „General principles for the development and use of programmable electronic systems in marine applications“) ist eine Adaption der internationalen Norm IEC 61508, welche die Internationale Elektrotechnische Kommission (IEC) für die Entwicklung von sicherheitskritischen Systemen bereitstellt. Diese Norm unterstützt die Bewältigung der komplexen Entwicklung neuartiger Automatisierungssysteme und die Risikominimierung eines möglichen funktionalen Fehlverhaltens. Um die Entwicklungs- und Testherausforderungen dieser Technologien während der Gestaltung und Entwicklung zu lösen, schlagen sowohl die ISO als auch die IEC die Verwendung der etablierten Methodik des Systems Engineering vor [Bell09], [Din05].

Systems Engineering ist ein interdisziplinärer Ansatz, welcher die Bewältigung der beschriebenen Komplexität der Systementwicklung von der Anforderungserhebung bis zur Inbetriebnahme einer Lösung unterstützt. Mit der Hilfe der Methoden des Systems Engineering erfolgt die Analyse, Gestaltung, Entwicklung und das Testen eines komplexen Systems [Wass05, S.1ff.]. Für diese Arbeit sind insbesondere die Methoden der Verifikation und Validierung (V+V) im Kontext des Testens als Teilprozesse der Systemgestaltung und Systementwicklung relevant. Validierung bezeichnet die Untersuchung inwiefern die Lösung die Bedarfe der Akteure deckt beziehungsweise ob die richtige Lösung ausgewählt wurde. Verifikation betrifft die Untersuchung hinsichtlich der korrekten Umsetzung der Funktionalität gemäß den Anforderungen und technischen Spezifikationen [BIFa13, S.108ff.], [Enge10, S.16]. Für die Erforschung und Entwicklung bedarf es entsprechender Testmöglichkeiten entlang des Systementstehungsprozesses [Wass05, S.1f].

Abbildung 1 belegt durch eine Studie über die Entwicklung intelligenter und kooperierender Automatisierungssysteme für den Flugtransport wie stark die Aufwände der V+V mit zunehmender Komplexität der zu bewältigenden Szenarien steigen. Die Aussagen treffen wegen der ähnlichen Führungsaufgaben eines Schiffes ebenso auf diese Forschungsarbeit zu.

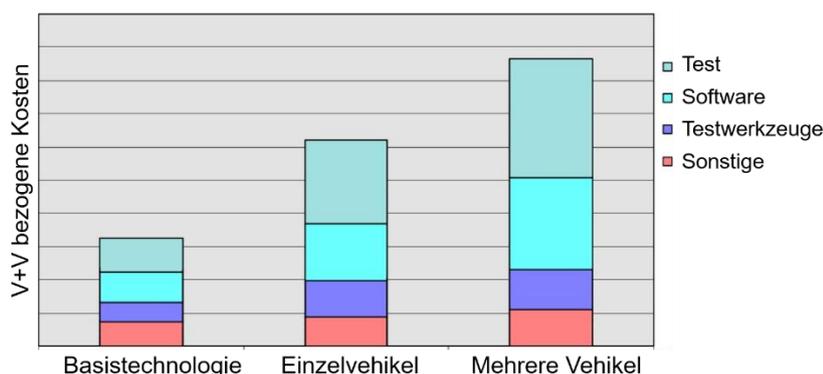


Abbildung 1: Kosten der V+V in der Entwicklung von Luftfahrtführungssystemen nach [TBSS06]

Die Entwicklung eines Systems für die automatisierte Führung eines Einzelvehikels erhöht den Aufwand für Testausführung (TEST), Softwareentwicklung (SW) und Entwicklung der Testwerkzeuge (TTD) gegenüber der Basistechnologie (zum Beispiel Beschleunigen und Steuern) um nahezu das Doppelte. Sofern mehrere automatisierte Vehikel die Führungsaufgabe in kooperierender Art und Weise erfüllen, erhöhen sich diese Aufwände insbesondere für die Testdurchführung signifikant [TBSS06].

[BaGi11] benennen als Ursache die unzureichenden Test- und Entwicklungsmethoden für neuartige Technologien und deren Herausforderungen während der Erprobung. Die gängige Praxis der Systementwicklung von intelligenten automatisierten Systeme sei das Testen einer Neuentwicklung ohne zielgerichtete Methodik, bis ein vorgegebenes Budget aufgebraucht und eine korrekte Funktionalität wahrscheinlich sei [BaGi11, S.3]. Die notwendigen Aufwendungen für einen erfolgreichen Test neuer sicherheitskritischer Automatisierungssysteme für Luftfahrt, Automobil oder Schifffahrt umfassen laut [BaGi11] mehr als 50% der Entwicklungsressourcen und benötigen dringend eine Senkung. Hier inbegriffen sind sämtliche dem Testen zuzuordnende Aufwendungen eines neuen Systems, insbesondere Integrationsaufwände, welche durch die heterogenen Systemlandschaft und die Einrichtung eines virtuellen und physikalischen Testfeldes resultieren [BaGi11, S.2f].

**Testfelder.** Die Verwendung von Testfeldern ist eine etablierte Möglichkeit, den Entwicklungs- und Testherausforderungen der zuvor beschriebenen komplexen Systeme zu begegnen [Hahn15]. Testfelder (engl. „Test bed“) bergen das Potenzial, das Testen durch die Erstellung von Modellen in frühen Systementwicklungsphasen über frühe Tests von Prototypen während der Gestaltung bis hin zu lieferbaren Systemen zu unterstützen. Der Duden definiert ein Modell als eine „Form, Beschaffenheit, Maßverhältnisse veranschaulichende Ausführung eines vorhandenen oder noch zu schaffenden Gegenstandes in bestimmtem besonders verkleinerndem Maßstab“. Allgemein repräsentiert ein Modell die physischen Eigenschaften und Informationen über ein System und dessen Umfeld als Gesamtheit, um die Funktionalität und das Verhalten im späteren Einsatzkontext nachzubilden [HaHG13, S.11f].

Ein Testfeld ist definiert als eine Testplattform, welche simulierten, emulierten oder stimulierten Komponenten die Integration in eine Testumgebung ermöglicht, um ein zu testendes System auszuführen und dieses während der Laufzeit zu erproben. Diese Testplattform besteht aus der für die Testdurchführung notwendigen Hardware, den Simulatoren, Testwerkzeugen und weiteren Elementen [Wass05, S.664ff.]. Das Industrial Internet Consortium beschreibt ein Testfeld als eine kontrollierte Experimentierplattform mit einer technologisch und V+V methodisch generischen Ausprägung, die verschiedene Anwendungsfälle und Testszenarien implementiert. Die Implementierung ermöglicht die Beobachtung, Analyse und Bewertung erwarteter und tatsächlicher Resultate [Iic17]. Unter Verwendung eines Testfeldes erfolgt ein Erkenntnisgewinn durch die Erforschung und Beobachtung des maritimen Verkehrssystems und der Erhebung von Daten zur Modellbildung und Modellverifikation [Hahn15]. Des weiteren dienen Testfelder für die Generierung neuer Systeme und Dienstleistungen, insbesondere für die Demonstration und das Testen dieser in der Zielumgebung [Iic17].

Ein Testfeld ist in der Lage komplexe maritime Umgebungssituationen bereitzustellen, um das Verhalten des gesamten Systems „Schiff“ unter Berücksichtigung der Komponenten und Subsysteme zu analysieren. Die International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) stellt an eine Systemarchitektur für ein Testfeld die Anforderung weder beschränkt noch durch gegenwärtige Technologie-Architekturen, Daten-

strukturen oder Prozesse limitiert zu sein. Das Testfeld befindet sich in virtuellen oder physikalischen Laboratorien beziehungsweise Einrichtungen oder auf mobilen Testträgern wie Schiffen. Somit ermöglicht ein Testfeld die nahtlose Überführung eines sich in der Entwicklung befindlichen Systems von einer virtuellen in eine physikalische Testumgebung [Iala17]. Durch Testfelder werden Fehler oder unerwünschte Ergebnisse bereits in frühen Entwicklungsphasen identifiziert und somit spätere, sehr aufwändige Fehlerbehebungen vermieden [Wass05, S. 664ff.]. Die Bereitstellung einer Testumgebung unterstützt die Systementwicklung durch die Möglichkeit zur Erprobung einzelner Komponenten bis zu ganzen Systemen unter möglichst geringen Anstrengungen [VeBK10]. Ein physikalisches Testfeld bietet in der maritimen Industrie und Wissenschaft das Potenzial, die Entwicklung und Erforschung durch die zuvor beschriebenen Einsatzmöglichkeiten zu beschleunigen und effizienter zu gestalten.

## 1.2 Problematik

In der Schifffahrt kommunizieren und interagieren seeseitige sowie küstenseitige Systeme als Systemverbund über kabellose Kommunikationskanäle. Diese Systeme sind unter anderem Schiffe, Häfen oder Flugobjekte über Wasser, die durch den Einsatz von Technologien verschiedene automatisierte Führungsaufgaben übernehmen [RLSS10]. Automatisierte Schiffsführungssysteme bestehen aus Teilsystemen und Komponenten wie Sensoren, Aktoren, Software und Kommunikationsinfrastruktur, deren integrierte rechnerische und physikalische Komponenten sowie Prozesse zu einem Systemverhalten führen. Daher ist ein automatisiertes Schiffsführungssystem ein cyber-phisches System (CPS). Hieraus resultiert das nötige Verständnis sowohl der physikalischen sowie rechnerischen Komponenten und deren Schnittstellen [LeSe17, S. 1f]. Baheti und Gill definieren den Begriff des cyber-phisches Systems als eine neue Generation von Systemen mit integrierten rechnerischen und technischen Möglichkeiten, die untereinander oder mit Menschen in neuer Art und Weisen interagieren [BaGi11]. Die im Rahmen eines CPS über eine Kommunikationsinfrastruktur verbundenen informations- sowie softwaretechnischen und mechanischen Komponenten erfüllen im Zuge des Transports eine zentrale Funktion. Dies ermöglicht die Realisierung neuartiger Automatisierungssysteme [Bend17]. Jedoch resultieren hierdurch ebenso neue Herausforderungen an die V+V dieser Systeme, beispielsweise durch Systemkomplexität, Funktionssicherheit, Kontrollierbarkeit insbesondere im Kontext offener Systeme und wechselnder Systemverbünde und Integration intelligenter Algorithmen mit nichtdeterministischem Systemverhalten [DaHe18], [HBFF16].

Ein integriertes Brückensystem für die Schiffsführung führt die in Kapitel 1.1 dargestellten Herausforderungen am Beispiel des intelligenten Autopiloten weiter. Wie Abbildung 2 exemplarisch illustriert, interoperieren diverse Sensorkomponenten (z. B. ein differentielles globales Positionierungssystem (DGPS), Wettersensorik, Rudersensorik, Radarsystem) mit wiederum selbst vernetzten Komponenten (z. B. das automatische Identifikationssystem (AIS) (engl. „Automatic Identification System“), ein elektronisches Kartendarstellungs-

und Informationssystem (engl. „ECDIS“), eine Schiffsführungsanzeige (engl. „Conning Display“). Erschwerend nehmen automatisierte Steuerungskomponenten beziehungsweise Aktoren am Systemverbund teil, wie ein Autopilot oder diverse Ruderanlagen. Die Komplexität erhöhend kommunizieren mehrere seeseitige Schiffe und küstenseitige Einrichtungen mit ähnlichen Systemen, die für unterschiedlichste Aufgaben kooperativ eine Vielzahl an Daten wie Routeninformationen erfassen und über eine Kommunikationsinfrastruktur austauschen [PaLP16]. Zukünftig erfolgt die Integration intelligenter und wechselnder partizipierender Systeme, wie beispielsweise küstenseitige Verkehrsführungssysteme bei der Einfahrt in einen Hafen. Diese Einrichtungen der maritimen Infrastruktur werden temporär ein Bestandteil des Systemverbunds für die automatisierte Schiffsführung [HaSS18].

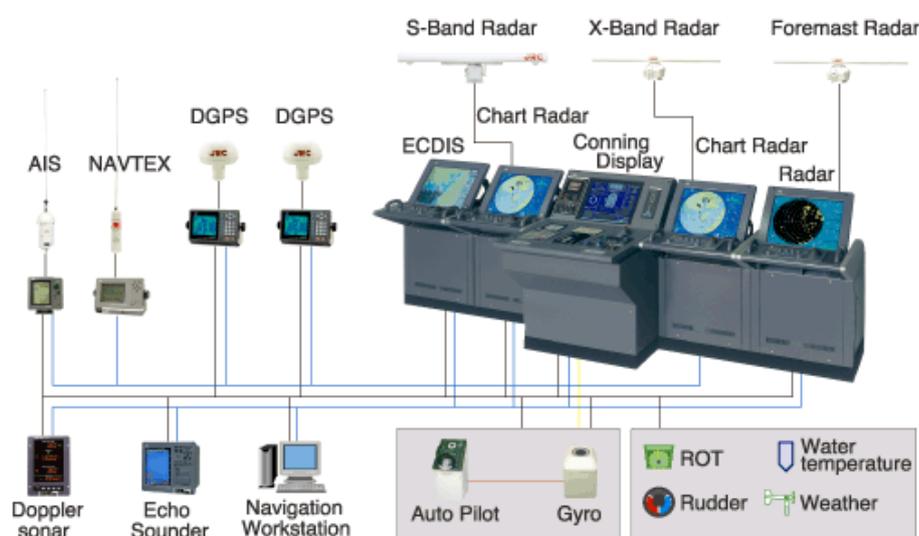


Abbildung 2: Integriertes Brückensystem [Sing15]

Die Aufgaben eines integrierten Brückensystems für die vernetzte und intelligente Navigation sind ausschließlich lösbar, sofern die Komponenten der Hardware, Software und Kommunikationstechnologie funktional korrekt und robust als Gesamtsystem kooperieren [GroV13, S.1ff.]. Es existieren diverse durch verschiedene Hersteller bereitgestellte Systemkomponenten, deren Implementierung auf der Basis unterschiedlicher, teilweise proprietärer Software- und Datenmodelle erfolgt. Für die Datenübertragung dienen weitreichende Standards, wie NMEA 0183 (National Marine Electronics Association) [Iec16], NMEA 2000 [Iec08], S-100 [WAGP08], IVEF (Inter VTS Exchange Format) oder proprietäre [LiHu06]. Die nähere Betrachtung dieser Standards für die Datenübertragung findet in Kapitel 2.1.3 statt.

Eine Herausforderung ist die Integration dieser heterogenen Komponenten, da diese komplexen Systemlandschaften mit zunehmend fortgeschrittenen Technologien zu hohen Integrationsaufwänden führen [BaGi11, S.1f]. Die maritime Domäne erkannte, dass „[...] das Schiff zu einem Gesamtsystem von miteinander verbundenen programmierbaren elektronischen Systemen und Besatzung geworden ist, die zusammenarbeiten, um die Geschäftsziele des Betreibers zu erfüllen“ [Din05, S.5]. Aktuelle und zukünftige Bemühungen führen diese Entwicklung weiter, den Faktor Mensch aus dem Gesamtsystem teilweise oder völlig durch

Automatisierungssysteme zu ersetzen [HaSS18]. Diese technologischen Entwicklungen der modernen sicherheitskritischen Systeme benötigen neuartige Entwicklungsmethoden.

**Validierung und Verifikation.** Das in Kapitel 1.1 vorgestellte Systems Engineering adressiert durch einen spezifizierten Entwicklungsprozess die Bewältigung der komplexen Systementwicklung und Risikominimierung eines funktionalen Fehlverhaltens. Es verankert den Einsatz von Methoden der V+V bereits in frühen Phasen der Anforderungsgestaltung durchgängig bis zur Systemabnahme. Die Entwicklung und das Testen dieser komplexen Systeme benötigen interdisziplinäres Fachwissen, beispielsweise über Kommunikationstechnologie, Regelungstechnik, Software, Elektrotechnik, Mechanik und Testmethoden. In der industriellen Praxis verläuft das Testen der entwickelten Komponenten und Teilsysteme meist lange entkoppelt und isoliert durch dedizierte Fachexperten, unter anderem wegen der Vielzahl von methodischen Herausforderungen der V+V. Dies führt zu hohen Aufwänden späterer Integrationstests des Gesamtsystems in der operativen Umgebung. Das durch aktuelle Regularien und Normen beschriebene isolierte Testen einzelner Teilsysteme ist für diese Bemühung unzulänglich [StHM08].

Beispielsweise liefert die Internationale Meeresorganisation (IMO) durch das Internationale Übereinkommen von 1974 zum Schutz des menschlichen Lebens auf See (SOLAS) verschiedene Konformitätsnormen, Leistungsstandards und Testmethoden für die Erprobung eines integrierten Navigationssystems (MSC.252(83)/IEC 61924-2) oder korrekte Daten eines AIS, das eine auf Funk basierende Technologie für den Austausch von Navigationsinformationen und Schiffsdaten für die Kollisionsverhütung ist [Imo17a]. Zwar werden sämtliche Einzelkomponenten gemäß den Leistungsstandards der IMO auf Funktionsfähigkeit geprüft, ein erfolgreiches Zertifikat der Einzelkomponenten führt jedoch nicht zwangsläufig zu einem funktionierenden Gesamtsystem. Systeme mit nichtdeterministischem Verhalten, beispielsweise in Form hochautomatisierter und autonomer Führungssysteme unter Einbezug dynamischer Systemverbünde und Methoden der künstlichen Intelligenz, stellen größere Herausforderungen an das Testen als die konventionellen Technologien für die Steuerung eines Schiffes [HuBH00]. Die kooperierenden Systeme müssen sowohl isolierten Teilkomponenten-Tests unterzogen werden, weitergehend die Funktionsfähigkeit des dynamischen Gesamtsystems erfüllen und im Test beweisen [Hahn15]. Ein Testfeld gilt es gemäß diesen Konzepten des Zielsystems und den aktuellen Methoden der V+V zu gestalten, um die Systementwicklung effizient unterstützen zu können.

**Maritime Testaufbauten.** Wie in Kapitel 1.1 motiviert, lassen sich die Gestaltung und Entwicklung von automatisierten Schiffsführungssystemen mithilfe von Testfeldern während des kompletten Systementwicklungszyklus unterstützen. Aktuelle maritime Testaufbauten verfügen über begrenzende Systemarchitekturen, Überprüfungsmethoden und Konzepte für die Gestaltung und Absicherung in speziellen Einsatzszenarien und vereinzelt Systementwicklungsstadien [Hahn15]. Da ein Testfeld die Umgebung für ein zu testendes System während des gesamten Systementwicklungsprozesses als generische Plattform bereitstellen muss, beinhalten physikalische Testaufbauten eine Vielzahl heterogener Komponenten. Diese bestehen aus Hardware, Software und Kommunikationsinfrastruktur. Die Komplexität

erhöhend erfolgt die Integration von Simulatoren für das Ersetzen der physikalischen Komponenten, um die real vorherrschende Situation zu ergänzen oder eine von dem Testszenario abweichende beziehungsweise ergänzende Umgebung zu simulieren [KKDH05]. Dies führt zu der Herausforderung existierende Komponenten, proprietäre Werkzeuge und verteilte Infrastrukturen in ein Testfeld integrieren zu müssen.

In der Praxis begegnen die Entwickler von Testumgebungen diesen Herausforderungen durch die Gestaltung der Test-Infrastruktur für einen eingeschränkten Umfang möglicher Anwendungen, um die Aufwände und Komplexität zu verringern [VeBK10]. Die IALA hingegen definiert als Zielvorgabe die notwendigen Testfelder für die Erprobung neuer Automatisierungstechnologie unter nachhaltigen und generischen Aspekten zu gestalten und zu nutzen. Neben Richtlinien für die Gestaltung und den Betrieb maritimer Testfelder listet die IALA gegenwärtig 46 Projekte auf, die sich mit dem Aufbau von Test-Infrastrukturen beschäftigen (Stand: 25.05.2018). Eine nähere Betrachtung der Auflistung stützt die Aussage von [VeBK10], denn nahezu sämtliche Test-Infrastrukturen dienen speziell eingegrenzten Anwendungsszenarien als Testaufbau und nicht als ein wiederverwendbares Testfeld im Sinne dieser Forschungsarbeit. Beispielsweise ermöglicht der physikalische Testaufbau des Projekts ACCSEAS die Erprobung einer modernen Navigation auf der Basis von Sensordatenfusion, taktischem Routenaustausch und weiteren speziellen Navigationsanwendungen in der Nordsee [Pora13]. EfficienSea erprobt in der Ostsee ebenfalls den Routenaustausch [Imo11]. Der Testaufbau des Projekts MUNIN validierte das Anwendungsszenario eines unbemannten Schiffes [Burm16].

Die exemplarisch aufgeführten maritimen Testaufbauten bieten keine offene und nachhaltige Systemarchitektur des Testfeldes für eine mehrfache Wiederverwendung in verschiedenen Testszenarien und werden meist nach erfolgreichem Test aufgelöst [Hahn15, S.3]. Dies ist allerdings aus Sicht der ingenieurwissenschaftlichen Entwicklungsmethodik des Systems Engineering und der Technologieentwicklung unüblich und resultiert in hohen Testaufwänden, wie in Kapitel 1.1 dargestellt. Die maritime Domäne erkannte die Ineffizienz des gegenwärtigen Vorgehens und beschloss die Zusammenarbeit zur Etablierung eines globalen e-Navigation Testfeldes, welches den vollständigen Entwicklungsprozess durch physikalische oder virtuelle Komponenten uneingeschränkt durch ein im Vorfeld definiertes Testszenario unterstützt und unabhängig von zu testenden Systemen als eine generische Testplattform dient [Imo14].

### **1.3 Zielsetzung und wissenschaftlicher Beitrag**

Der in dieser Arbeit zu entwickelnde Ansatz, ausgehend von der zuvor beschriebenen Problematik, erfüllt den Bedarf nach der Gestaltung eines generischen physikalischen Testfeldes zur Unterstützung der Entwicklung von automatisierten Schiffsführungssystemen. Für diesen Zweck erfolgt die Entwicklung einer technischen Lösung für die integrierte und wiederverwendbare Systemarchitektur eines physikalischen Testfeldes (im Folgenden „Testfeld-Architektur“) für automatisierte Schiffsführungssysteme, die sich an den etablierten

Systems Engineering Methoden orientiert. Die Testfeld-Architektur ist unter den Aspekten der Wiederverwendbarkeit und Erweiterbarkeit gestaltet, um für eine große Menge von physikalischen Testszenarien flexibel nutzbar zu sein. Ein gemäß der in dieser Arbeit vorgestellten Systemarchitektur implementiertes physikalisches Testfeld ermöglicht das Testen von automatisierten Schiffsführungssystemen unter realitätsnahen Bedingungen nahtlos während des Systementwicklungsprozesses. Der Schwerpunkt ist das Finden einer Antwort auf die wissenschaftliche Fragestellung:

*Wie muss ein physikalisches Testfeld gestaltet sein, um die Verifikation und Validierung in der Systementwicklung von automatisierten Schiffsführungssystemen effizient zu unterstützen?*

In dieser Arbeit erfolgt neben der Konzeption der Systemarchitektur eines offen und generisch gestalteten physikalischen Testfeldes die Implementierung im Rahmen der e-Maritime Integrated Reference Platform (eMIR). eMIR ist eine 2013 gestartete Initiative der deutschen maritimen Industrie und verschiedenen Forschungseinrichtungen mit dem Ziel, ein maritimes Testfeld aufzubauen [Hahn15]. Auf der Basis der oben beschriebenen Forschungsfrage ergeben sich im Rahmen der vorliegenden Arbeit die folgenden Teilfragestellungen.

**Teilfrage 1:** Welche Technologiekonzepte und Systementwicklungsmethoden müssen in der Gestaltung des physikalischen Testfeldes berücksichtigt werden?

**Ziel 1**

Berücksichtigung von Technologien und Konzepten des automatisierten Schiffsführungssystems als das Zielsystem

Da ein Testfeld die Umgebung für ein zu testendes System oder Einzelkomponenten bereitstellt, müssen die hieraus resultierenden Anforderungen des zu testenden Systems an das Testfeld identifiziert werden. Insbesondere deren spezifischen Eigenschaften müssen berücksichtigt werden, die aus vernetzten und heterogenen Systemen und Systemverbänden mit dynamischer kooperativer Aufgabenlösung resultieren. Ebenso deren Einfluss auf das Testfeld aus einer systemischen Perspektive. Die Herausforderungen der komplexen Systemlandschaft, deren lose gekoppelte Systeme und Komponenten unterschiedlicher Hersteller sich dynamisch verändernd kooperieren, gilt es als Zielsetzung dieser Arbeit hinreichend zu untersuchen. Dies zielt auf die Gestaltung einer an das Zielsystem gerichteten Systemarchitektur eines physikalischen Testfeldes ab, das für die V+V von automatisierten Schiffsführungssysteme nutzbar ist.

**Ziel 2**

Unterstützung der Systementwicklungsmethodik durch V+V

Auf der Basis von Ziel 1 ergeben sich neue Herausforderungen an die Systementwicklungsmethodik und Methoden der V+V. Komplexe Automatisierungssysteme werden entsprechend etablierter Systementwicklungsmethoden unter Nutzung von Methoden der V+V entwickelt, für deren Durchführung physikalische Testfelder herangezogen werden. Um die unwirtschaftlichen Aufwände zu senken, die aus einer physikalischen Erprobung auf See resultieren, muss das Konzept des physikalischen Testfeldes bereits in frühen Systementwick-

lungsphasen und weitergehend möglichst nahtlos während des gesamten Systementwicklungsprozesses unterstützen. Ein Ziel ist daher die Gestaltung des physikalischen Testfeldes in der Art und Weise, dass die Erprobung des zu testenden Systems gemäß den etablierten Methoden des Systems Engineering unter realitätsnahen Bedingungen erfolgt. Die entsprechenden Standards, Zulassungs- und Klassifizierungsverfahren unter anderem der IMO, bieten gegenwärtig keine Lösung für die ausreichende und ergebnisorientierte V+V dieser neuartigen Systeme. Daher erfolgt die Untersuchung von Standards wie der ISO 26262 der Automobilbranche, die einen größeren Fortschritt in diesen Fragestellungen aufweisen. Dort werden seit Jahren Teilkomponenten oder ganze Systeme für die Automatisierung von Fahrerfunktionalitäten mithilfe von Testfeldern getestet, validiert und verifiziert. Diese Erfahrungen und Methoden werden auf die maritime Branche übertragen, um den Einsatz eines physikalischen Testfeldes zur Unterstützung parallel zum Entwicklungsprozess herzustellen.

**Teilfrage 2:** Wie stellt sich eine wiederverwendbare Gestaltung durch die Systemarchitektur des physikalischen Testfeldes dar?

### **Ziel 3**

Nahtlose V+V durch die Integration virtueller und physikalischer Komponenten

Ein Testfeld ermöglicht die Erprobung eines Systems entlang des Systementwicklungsprozesses, um bereits möglichst in frühen Entwicklungsstadien vorhandene Fehler sowie Risiken und deren Auswirkungen beurteilen zu können. Um einen möglichst geringen Aufwand für die Durchführung der V+V in der Simulationsumgebung (virtuelles Testfeld) bis zur Erprobung in der realen Umgebung (physikalisches Testfeld) zu erreichen, muss die Gestaltung der Testfeld-Architektur einen nahtlosen Übergang zwischen virtuellen und physikalischen Komponenten des Testfeldes ermöglichen. Diese Arbeit zielt daher auf die Entwicklung der Systemarchitektur des physikalischen Testfeldes dahingehend ab, dass die V+V während des Systementwicklungsprozesses durchgehend und nahtlos ermöglicht wird.

### **Ziel 4**

Flexible Integration des zu testenden Systems in die Infrastruktur des Testfeldes

Die Integration eines zu testenden Systems gemäß dessen inhärenten Schnittstelle in die Infrastruktur des physikalischen Testfeldes muss effizient und ohne Modifikationen entweder des zu testenden Systems oder der Testfeld-Infrastruktur erfolgen. Während die verschiedenen Testfeld-Komponenten wiederverwendet oder gegebenenfalls während des Implementierungszeitpunkts unbekannt sind, stellt ein zu testendes System andere Herausforderungen an die Integration in die Testfeld-Infrastruktur. Die Schnittstelle des zu testenden Systems ist entweder gemäß variierenden Standards definiert oder gemäß einer proprietären Schnittstellendefinition des jeweiligen Systementwicklers. Hieraus resultiert der Bedarf nach einem technologischen Konzept, das die Implementierung eines zu testenden Systems und die Infrastruktur des Testfeldes voneinander entkoppelt und Interoperabilität zwischen diesen erreicht. Dies muss durch eine dynamische Integration des zu testenden Systems zur Laufzeit der Testausführung erfolgen.

**Ziel 5**

Anpassung/Erweiterung an zukünftige Anwendungen für Wiederverwendbarkeit

Wegen der beschriebenen sich rasch weiterentwickelnden Technologien für die Automatisierung von Schiffsführungssystemen, muss das physikalische Testfeld funktional an neue Anforderungen angepasst werden können. Insbesondere zielt diese Arbeit auf die Gestaltung und Bereitstellung eines nachhaltigen und generischen Testfeldes ab, das zukünftige Automatisierungsanwendungen unterstützt. Diese Zielsetzung folgt aus der Betrachtung des Testfeldes als eine generische Testumgebung zur Durchführung von Methoden der V+V und nicht die Durchführung einer einzelnen Erprobung eines im Vorfeld spezifizierten Systems. Um diese Lücke gegenwärtiger maritimer Testaufbauten zu schließen und eine Senkung der Aufwände für V+V zu erreichen ist ein Ziel dieser Arbeit die Gestaltung des physikalischen Testfeldes unter anpassbaren und erweiterbaren Aspekten.

**Wissenschaftlicher Beitrag.** Diese Arbeit leistet den Erkenntnisgewinn einer Lösung für die in Kapitel 1.2 beschriebene Problematik in der Form einer Systemarchitektur für die Implementierung eines generischen physikalischen Testfeldes. Die Testfeld-Architektur orientiert sich an der Systementwicklungsmethodik für maritime Systeme, um die mit modernen Technologien und der notwendigen Sicherstellung der funktionalen Korrektheit einhergehenden Herausforderungen zu bewältigen. Der Lösungsbeitrag dieser Arbeit ermöglicht die Umsetzung neuer Verfahren der V+V von automatisierten Schiffsführungssystemen. Das gemäß der entwickelten Testfeld-Architektur realisierte physikalische Testfeld stellt eine integrierte und offene Testumgebung für die Erprobung, Verifikation und Validierung dar.

Durch diese Gestaltung ist die Unterstützung der Systementwicklung von unterschiedlichen automatisierten Schiffsführungssystemen unter nachhaltigen Gesichtspunkten möglich. Ein besonderer Fokus liegt auf der dynamischen Integration von beliebigen Systemen in die von dem zu testenden System entkoppelte Testfeld-Infrastruktur. Die maritime Industrie und Forschung profitieren vom Einsatz des physikalischen Testfeldes für die Entwicklung neuer Technologien, beispielsweise durch verkürzte Entwicklungszeiten und die Einsatzfähigkeit neuer Methoden der V+V zur Sicherstellung korrekter Funktionalität. Weitergehend ermöglicht das physikalische Testfeld die Erforschung neuer wissenschaftlicher Fragestellungen in einer realistischen Umgebung. Dieses Ergebnis erweitert das Feld der V+V als Bestandteil der interdisziplinären Methodik des Systems Engineering. Daher lautet der wissenschaftliche Beitrag dieser Arbeit:

*Entwicklung und Verwendung einer offen und flexibel gestalteten Systemarchitektur eines physikalischen Testfeldes für die Unterstützung der Entwicklung von automatisierten Schiffsführungssystemen.*

## 1.4 Vorgehen und Struktur der Arbeit

Diese anwendungsorientierte Forschungsarbeit strebt die Konzeption und Realisierung einer Systemarchitektur eines physikalischen Testfeldes für die Unterstützung der Entwicklung

von automatisierten Schiffsführungssystemen an. Daher folgen das Vorgehen und die Struktur dieser Arbeit den Phasen eines ingenieurwissenschaftlichen Systementwicklungsprozesses, wie in Abbildung 3 dargestellt und im folgenden erläutert [Burg12].

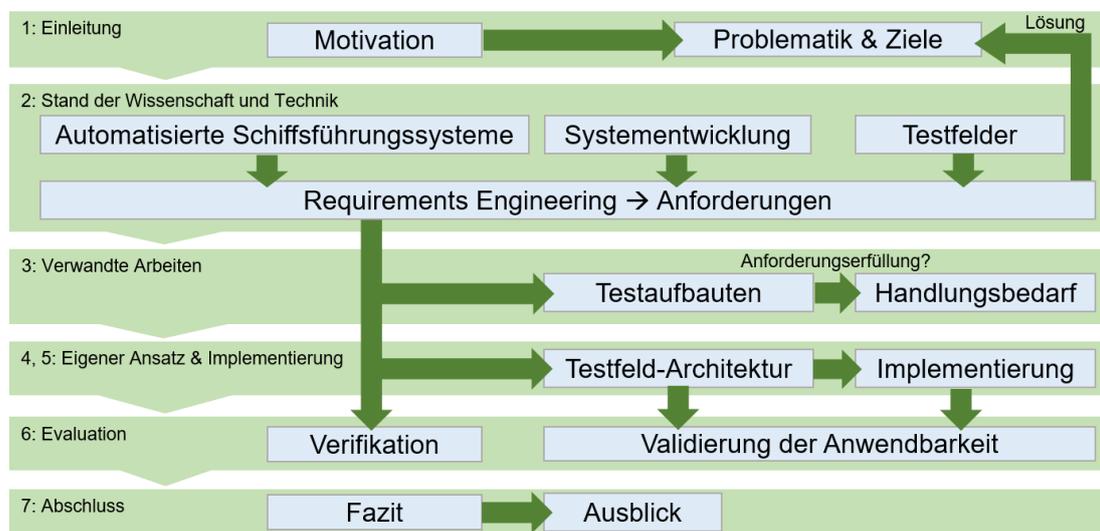


Abbildung 3: Ingenieurwissenschaftliche Methodik der Arbeit

Kapitel 1 leitet ausgehend von der Motivation und Problematik dieser Arbeit die technischen und wissenschaftlichen Fragestellungen und Ziele der aktuellen Systementwicklung her, insbesondere der hierin verankerten V+V von automatisierten Schiffsführungssystemen und den Einsatz von Testfeldern.

Als Methode der qualitativ systematischen Erfassung und Deutung von Tatbeständen dient in Kapitel 2 eine Dokumentenanalyse der Literatur, Projektanträge, Projektdokumentationen und Veröffentlichungen. In diesem Rahmen wird der aktuelle Stand der Wissenschaft und Technik als Ableitungsgrundlage für das Requirements Engineering zur Identifikation der Systemanforderungen erörtert. Ein Requirements Engineering dient für die Erhebung, die Analyse, das Verständnis und die Dokumentation von zu erfüllenden Anforderungen durch ein System [Burg12], [VWFG13, S.9]. Der aktuelle Stand der Wissenschaft und Technik resultiert zum einen durch das zu testende System in Form des automatisierten Schiffsführungssystems, zum anderen durch die Systementwicklungsmethodik des Systems Engineering und in diesem Zusammenhang die Unterstützung der notwendigen V+V durch Testfelder.

In Kapitel 3 werden die verwandten Arbeiten in Form maritimer Testaufbauten und Testfelder der Luft- und Raumfahrt hinsichtlich der Anforderungserfüllung untersucht, woraufhin die Beschreibung der identifizierten Lücken und des Handlungsbedarfes stattfindet. Hierauf basierend folgt in Kapitel 4 die Gestaltung der Testfeld-Architektur und der hierin verankerten Konzepte und Methoden. Das Kapitel 5 beschreibt die Implementierung der Testfeld-Architektur durch das physikalische Testfeld LABSKAUS (Labor für sicherheitskritische Analysen auf See) im Kontext von eMIR. Im Rahmen der Evaluation in Kapitel 6 erfolgt die Validierung der Testfeld-Architektur durch die Überprüfung der Anwendbarkeit gemäß repräsentativer Einsatzszenarien von LABSKAUS und die hierauf basierende Verifikation der

Anforderungserfüllung [FeLo04, S.7f]. Den Abschluss dieser Arbeit bilden eine Zusammenfassung und der Ausblick auf weitere Forschungs- und Einsatzmöglichkeiten der Testfeld-Architektur in Kapitel 7.

## 2 Verifikation und Validierung von automatisierten Schiffsführungssystemen

Das Ziel dieser Arbeit ist die Unterstützung der Systementwicklung für automatisierte Schiffsführungssysteme durch die Konzeption einer Systemarchitektur für ein generisches physikalisches Testfeld. Dieses Kapitel untersucht relevante Technologien und Ansätze des aktuellen Stands der Technik, um Anforderungen an einen potenziellen Lösungsansatz abzuleiten. Für die Gestaltung einer Systemarchitektur erfolgt in dieser ingenieurwissenschaftlichen Arbeit die Definition von Systemanforderungen im Rahmen des Requirements Engineering auf der Basis der folgenden drei Bereiche [Burg12], [VWFG13, S.9].

Zunächst stellt Kapitel 2.1 die Systemkonzepte des mit der Hilfe eines physikalischen Testfeldes zu erprobenden Zielsystems vor. Hierfür erörtert diese Arbeit die relevanten Charakteristika der Schifffahrt und die konzeptionellen Grundlagen eines automatisierten Schiffsführungssystems. Anschließend findet die Betrachtung der Methoden der Systementwicklung gemäß dem Systems Engineering in Kapitel 2.2 statt. Darauf aufbauend fokussiert dieses Kapitel die Unterstützung der Systementwicklung durch Methoden der V+V und insbesondere den Einsatz von physikalischen Testfeldern. Die aus diesen drei Bereichen abgeleiteten Systemanforderungen dienen in der weiteren Arbeit der Untersuchung verwandter Arbeiten in Form aktueller maritimer Testaufbauten und Testfelder der Luft- und Raumfahrt als Untersuchungskriterien und Gestaltungsziele einer Systemarchitektur [Burg12].

### 2.1 Automatisierte Schiffsführungssysteme

Gegenwärtige Schiffsführungssysteme bedienen sich zwar nur limitierter Automatisierung und Intelligenz, dennoch existieren technologische Konzepte und Technologien durch die zukünftige Schiffe zunehmend automatisiert werden. Mit der Hilfe von Autopiloten zum Folgen einer Trajektorie wird die Besatzung eines Schiffes unterstützt oder durch die Integration von Kollisionsverhütungsfunktionen in die Automatisierungssysteme das intelligente Schiff ermöglicht. Diese Funktionalitäten setzen ein Wahrnehmungs-, Planungs- und Kontrollsystem voraus, das eine Vielzahl komplexer Fähigkeiten benötigt, wie beispielsweise die akkurate Positionsbestimmung für das Folgen einer vordefinierten Route. Dies findet unter Berücksichtigung diverser zu erfassender und verarbeitender Parameter statt, wie die Geschwindigkeit, Umweltbeschaffenheit, der interne Zustand des Schiffes oder die Verkehrsteilnehmer im relevanten Umkreis [AbFH15].

Die Basis für diese Funktionalitäten bildet ein Verbund aus informatischen und softwaretechnischen Komponenten mit mechanischen und elektronischen Teilen im Sinne des in Kapitel 1.2 vorgestellten CPS für kooperierende Schiffe und weiterer maritimer Infrastruktureinrichtungen. Diese Systeme erfassen und verarbeiten Daten aus der Umwelt und erzeugen in automatisierter Weise die hierauf reagierenden mechanischen Bewegungen und Reaktio-

nen [FoPN17, S.5]. Ein automatisiertes Schiffsführungssystem kategorisiert sich in verschiedene Stufen der Automatisierung. Diese reichen von der vollständig manuellen Schiffsführung ohne Automatisierung bis zur Vollautomatisierung dieser. Tabelle 1 stellt diese Stufen als Ableitung der seit Januar 2014 gültigen Norm SAE J3016 für die Klassifizierung und Definition von Automatisierungssystemen in straßengebundenen Kraftfahrzeugen dar. Diese Klassifizierung lässt Rückschlüsse auf die Anforderungen des zu testenden Systems an das Testfeld zu [HaSS18].

Stufe	Automatisierung	Beschreibung
0	Keine	Mensch ist zuständig für das Steuern, Überwachen der Umwelt und die Kollisionsvermeidung.
1	Assistenzsysteme	Mensch wird beim Steuern vom Autopiloten unterstützt. Assistenzsysteme helfen bei der Wahrnehmung der Umwelt und der Kollisionsvermeidung.
2	Teilautomatisierung	Stufe 1 und zusätzlich werden Vorschläge für Manöver zur Kollisionsvermeidung ausgegeben.
3	Hochautomatisierung	Das Schiff fährt völlig autonom, muss jedoch nach wie vor vom Menschen überwacht werden. An- und Ablegemanöver müssen manuell durchgeführt werden.
4	Vollautomatisierung	Das Schiff fährt voll autonom inklusive An- und Ablegemanöver. Es muss kein Mensch an Bord sein.

Tabelle 1: Stufen für automatisierte Schiffsführungsfunktionen [HaSS18]

Die folgenden Unterkapitel bilden die systemperspektivische Grundlage für ein physikalisches Testfeld, welches eine Testumgebung für die hier beschriebenen Automatisierungskomponenten und -systeme bietet. Ein automatisiertes Schiffsführungssystem ist durch die notwendige Berücksichtigung spezifischer Eigenschaften von Umwelt und Technologie der maritimen Domäne gekennzeichnet. Diese Merkmale werden im Folgenden beschrieben und dienen als Verständnisgrundlage und Ableitungsgrundlage der Anforderungen für die Gestaltung einer physikalischen Testfeld-Architektur aus der Systemperspektive.

### 2.1.1 Hydrodynamische Umgebung

Laut Duden bezeichnet die Hydrodynamik „(als Teilgebiet der Strömungslehre) die Wissenschaft und Lehre von den Bewegungen der Flüssigkeiten und den dabei wirksamen Kräften.“ Die hydrodynamischen Kräfte wirken auf den Körper eines Schiffes, den es durch entsprechende Regelungstechnologie für die Kontrolle zu beeinflussen gilt. Diese Einflüsse auf die Schiffsdynamik führen zu umfangreichen von Sensoren erfassten Parametern, resultierend in einem komplexen digital abzubildenden Umweltszenario [ÄsKä76, S.9ff.]. Die hydrodynamische Umgebung betrifft Parameter der Umwelt wie Strömung, Wellen und Wind. Die genaue Modellierung dieser Umweltkräfte ist für das simulierte Testen wichtig, wohingegen im physikalischen Testen durch die reale Umgebung korrekte Umweltdaten vorliegen. Die Herausforderungen hierin liegen in der Wahrnehmung dieser Umweltdaten durch Sensoren

sowie die Übertragung der entstehenden großen Datenströme in vielen Fällen über eine drahtlose Kommunikationsinfrastruktur.

Im Entwurf von automatisierten Schiffsführungssystemen stellen sich die hydrodynamischen Faktoren herausfordernder dar als die Fahroberfläche von Kraftfahrzeugen. Die Seeverhältnisse in Form von Wind, Strömung und Wellen nehmen erheblichen Einfluss sowohl auf ein zu testendes System sowie auf das physikalische Testfeld, beispielsweise in Form eines sicher zu steuernden Schiffes als Testträger. In der maritimen Domäne wird eine Schiffsdynamik entsprechend des Modells bis zu sechs Freiheitsgrade (engl. „Six Degrees of Freedom“, 6DOF) für erweiterte Manöver beschrieben. 6DOF bedeutet eine mögliche Schiffsbewegung in drei Richtungen und eine Schiffsrotation um drei Achsen [ÅsKä76]. In vielen Funktionen der automatisierten Schiffsführungssysteme reicht jedoch ein 3DOF Modell, welches durch eine entsprechende Testfeld-Komponente repräsentiert wird, wie ein Schiff als Testträger [SKKR11].

Eine automatisierte Steuerung und ein physikalisches Testfeld zur Erprobung dieser muss diese hydrodynamischen Kräfte insbesondere in einem Extremszenario auf rauer See kontrollieren und die funktionale Sicherheit aller Verkehrsbeteiligten gewährleisten [Foss11, S.187ff.]. Somit ist die hydrodynamische Umgebung in dem physikalischen Testfeld zu berücksichtigen, beispielsweise in Form entsprechend wahrnehmender Sensorik oder der Testmethode, die vom Testfeld ein spezielles Umweltszenario gemäß der Spezifikation des Tests erfordert [FoPN17, S.409]. Weitergehend muss die Infrastruktur des physikalischen Testfeldes ebenfalls die Umweltverhältnisse des Tests beobachten, während der Testdurchführung dokumentieren und gegebenenfalls Gebrauch von Sicherheits- und Kontrollmechanismen machen.

Das Führen eines Schiffes in der hydrodynamischen Umgebung erfolgt durch die Wahrnehmung der Umwelt und die hierauf reagierende mechanische Bewegung durch ein automatisiertes Schiffsführungssystem auf Basis unterschiedlicher funktionaler Ebenen. Diese funktionalen Ebenen eines automatisierten Schiffsführungssystems beschreibt das folgende Kapitel.

### **2.1.2 Drei-Schichten-Modell eines automatisierten Schiffsführungssystems**

Ein Automatisierungssystem für die Führungsaufgaben eines Schiffes teilt sich funktional in die Teilaspekte der Führung, Navigation und Kontrolle (engl. „Guidance, navigation and control“). Diese durch Systemkomponenten adressierten Führungsfunktionen interagieren durch Daten- sowie Signalübertragung und variieren bezüglich des zeitlichen Horizontes und der notwendigen Informationen [Foss11, S.232f]. Dieser Ansatz findet ebenso in der Kraftfahrt Verwendung, denn er bildet sämtliche Aspekte zur Führung eines Vehikels und deren Beziehung zu der Umwelt ab. In der Kraftfahrt erfolgt seit Jahrzehnten die Erforschung und Entwicklung der Automatisierung von Fahrfunktionen. So wurde seit 1982 ein Modell

entwickelt, welches die Aspekte des Fahrer-Vehikel-Systems in einer dreischichtigen Architektur abbildet [Dong12]. Abbildung 4 illustriert das Drei-Schichten-Modell der Fahreraufgaben nach Donges und vereint dies mit dem maritimen Konzept von Fossen aus dem Jahr 2011.

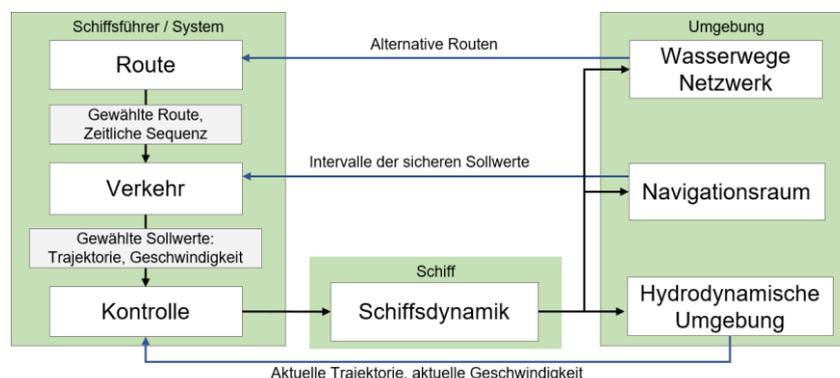


Abbildung 4: Drei-Schichten-Modell zur Schiffsführung nach [Dong12] und [Foss11, S.232]

**Route.** Auf der obersten Schicht finden die Routenplanung und Routenüberwachung statt, welche die Aufgabe der strategischen Navigation durch das Finden und Modifizieren möglicher Routen realisieren. Die Basis hierfür bildet das zur Verfügung stehende Netzwerk der Wasserwege unter Berücksichtigung der verschiedenen Parameter, welche eine gute, bessere oder die beste Route bestimmen. Eine Route bezeichnet einen vordefinierten Weg, um ein Schiff von einem Startpunkt zu einem Endpunkt zu führen. Aktuelle Störeinflüsse machen gegebenenfalls das Ändern der Route notwendig, wie das Wetter oder regulatorische Beschränkungen wie Verkehrstrennungsgebiete. Die Routenkomponente benötigt unter anderem die Informationen über das Netzwerk der Wasserwege eines Seegebietes, wie Seewege, statische Hindernisse oder die Tiefenlinien in einer digitalen Seekarte.

Die Routenkomponente sendet die berechnete und gewählte Route inklusive der zeitlichen Aspekte an die Schicht der Verkehrskomponenten. Die notwendige Latenz für die Schiffsführungsaufgaben zwischen der Wahrnehmung und Aktionsausführung auf der Schicht der Routenplanung und Routenüberwachung basiert auf der navigatorischen Entscheidungszeit und den Manöviereigenschaften des jeweiligen Schiffes. In diesem Zusammenhang beziffert sich die Latenz auf einen Wert von sechs Minuten [Szl12].

**Verkehr.** Das Drei-Schichten-Modell absteigend folgt in der mittleren Schicht die Aufgabe der Verkehrsführung (engl. „Guidance“) unter Berücksichtigung des vorliegenden Navigationsraumes und der Verkehrssituation, wie durch Gezeiten oder dynamische Hindernisse beeinflusst. Die Verkehrskomponente überwacht taktisch das Abfahren der strategisch gewählten Route auf örtlich punktueller beziehungsweise zeitlich diskreter Basis. Die Verkehrsführung berechnet kontinuierlich die erwünschte Position, Geschwindigkeit und Beschleunigung eines Schiffes. Für diesen Zweck erfassen Sensoren die vorherrschende Situation auf einem Schiff, beispielsweise die Beschleunigung, Rotationsgeschwindigkeit oder dynamische Objekterkennung. Komplementiert wird diese Situation durch eine landseitige Lagebilderfassung, wie beispielsweise die Verkehrslenkung durch den Schiffsverkehrs-

dienst (engl. „Vessel Traffic Service“, kurz VTS). Mithilfe dieser Informationen wird ein Lagebild erzeugt und verschiedene Anwendungen durchgeführt, wie die Kollisionsverhütung oder Kursoptimierung. Die Verkehrskomponenten benötigen hierfür Daten der gegenwärtigen Verkehrssituation und Umgebung, wie Schiffe in unmittelbarer Nähe oder ein durch Objekte begrenzter Navigationsraum.

Die Sensoren für die Lagebilderzeugung über Wasser sind beispielsweise das auf Funkwellenortung basierende Radarsystem, auf Laserstrahlen basierende Lidarsystem und die Erkennung mithilfe von optischen Kameras und einer hierauf operierenden Bilderkennung. Für die Unterwasser Erkennung existiert das Echolot, welches die Wassertiefe mithilfe einer elektroakustischen Messung erfasst. Ergänzend hierzu ermöglichen Mikrofone die Auswertung beispielsweise durch andere Schiffe erzeugte akustische Schallsignale. Die Windrichtung sowie die Windgeschwindigkeit misst ein Anemometer, während ein Barometer den Luftdruck feststellt. Die Luft- und Wassertemperaturmessung findet mithilfe eines Thermometers statt [Foss11, S.233]. Die Verkehrskomponenten berechnen die Intervalle der sicheren Sollwerte in Form von Trajektorien und Geschwindigkeiten. Diese Informationen übergeben die Verkehrskomponenten den Kontrollkomponenten, um auf die aktuelle Situation des Navigationsraumes durch ein Manöver reagieren zu können, wie beispielsweise durch die internationalen Kollisionsverhütungsregeln oder nationalen Regularien bestimmt [JoPC16]. Ähnlich wie bei den Führungsaufgaben auf der Schicht der Route zuvor orientiert sich die Latenz für die Aufgaben der Schicht des Verkehrs an der navigatorischen Entscheidungszeit, die einen Wert von sechs Minuten zwischen der Wahrnehmung und resultierenden Aktion vorgibt [Szła12].

**Kontrolle.** Die unterste Schicht der Kontrolle repräsentiert die direkte Reaktion auf Abweichungen des Ist-Zustands des Schiffskörpers vom erwarteten Soll-Zustand, beispielsweise zur Steuerung oder Stabilisierung auf der Grundlage der hydrodynamischen Umgebung. Auf der Basis der Schiffsdynamik, unmittelbaren Umgebung und hydrodynamischen Oberfläche erfolgt die dynamische Bewegungskontrolle des Schiffes. Die Stabilisierung der Eigenbewegung des Schiffes basiert auf den sich stetig verändernden Schiffsbewegungen und resultierenden Sensorinformationen. Die Verkehrskomponenten geben beispielsweise in Form einer Trajektorie und Geschwindigkeit als sinnvoll erachtete Sollwerte vor, während die Kontrollkomponenten die zur Umsetzung notwendigen Steuerbefehle berechnen und diese an die Aktorik weiterleiten, um den Schiffskörper zu bewegen [Dong12, S.15–23], [Foss11, S.232ff.]. Die Bewegungskontrolle bestimmt ob die aktuellen Kontrollkräfte und das Drehmoment die Kontrollziele der Verkehrskomponenten erfüllen. Insbesondere die Stabilität und Manövrierfähigkeit gemäß den Zielvorgaben des Schiffes wird auf der Schicht der Kontrolle betrachtet.

Für die automatisierte Führung eines Schiffes sind die drei Fähigkeiten einer sicheren und zuverlässigen Positionierung, Navigation und des Timings (PNT) von entscheidender Bedeutung, auf deren Basis die vorgegebene Route beziehungsweise Trajektorie angepasst wird [Imo17b]. Positionierung beschreibt die Fähigkeit akkurat und präzise den Ort und die Orientierung in Relation zu einem etablierten Standard zu bestimmen, wie das weltweite geodätische Referenzsystem (engl. „World Geodetic System“) mit der Bezeichnung

WGS 1984. Die Navigation bestimmt die gegenwärtige und gewünschte Position sowie die Korrektur der Abweichungen des Kurses, der Orientierung und Geschwindigkeit gemäß dem Soll-Kurs. Das Timing bezeichnet die Fähigkeit akkurate und präzise Zeitbestimmung gemäß einem Standard durchzuführen, wie der koordinierten Weltzeit. Um diese Fähigkeiten zu ermöglichen liefern unterschiedliche Datenquellen die nötigen Daten. Dies sind beispielsweise ein Kompass beziehungsweise Gyrokompass oder ein globales Navigationssatellitensystem (GNSS). Die Geschwindigkeit und Beschleunigung eines Schiffes lässt sich ebenfalls mithilfe eines GNSS oder eines Logs bestimmen [Imo17b], [DEGN12, S.119ff.]. In der Regel erfolgt eine Sensordatenfusion mehrerer Datenquellen, um die Robustheit und Genauigkeit zu erhöhen [LiHL09, S.1f].

Auf dieser Schicht führen die Aktoren zur Bewegung des Schiffes die mechanischen Kräfte aus, wie der Propeller, das Strahlruder oder Tiefenruder [Foss11, S.233]. Essenziell notwendig für die Signale an die Aktoren eines automatisierten Schiffes ist das Wissen über den Zustand dieser Aktoren. Hierfür liefert die Motorensteuerung (engl. „Motor Control Unit“) die Informationen über die Motorfunktionalitäten und ein Ruderlagenanzeiger den Ausschlag des Ruders in Grad [StHM08]. Die Lage eines Schiffes im Wasser (Stampf- und Rollwinkel sowie Hub) misst eine inertielle Messeinheit (IMU). Eine Komponente auf der Schicht der Kontrolle sendet Befehle an das Schiff und benötigt daher für das Setzen der Sollwerte in Echtzeit die Informationen über die hydrodynamische Umgebung, wie die Geschwindigkeit oder Ruderlage. Relevante Informationen sind beispielsweise Wellen oder die Wasserströmung [HBFF16], [SKKR11].

Die Latenz der Schiffsführungsaufgaben auf der Schicht der Kontrolle sollte kleiner als eine Sekunde für informationsdarstellende Assistenzsysteme sein. Im Falle einer kritischen Informationsdarstellung beziehungsweise der automatisierten Kontrolle des Schiffes ist eine Latenz von kleiner als 100 Millisekunden zwischen der Wahrnehmung und resultierenden Ausführung vorgegeben [Dnvg11].

Das in diesem Kapitel beschriebene automatisierte Schiffsführungssystem ist ein Systemverbund im Sinne eines CPS. Den resultierenden Austausch von Daten zwischen den Schichten der Route, des Verkehrs und der Kontrolle beschreibt das folgende Kapitel.

### **2.1.3 Datenaustausch innerhalb des Drei-Schichten-Modells**

Die zu automatisierenden Schiffsführungsfunktionen werden gemäß dem zuvor beschriebenen Drei-Schichten-Modell realisiert, das die Datenflüsse zum Informationsaustausch zwischen den verschiedenen Systemen und Komponenten beschreibt. In diesem Zusammenhang existieren verschiedene Ebenen des Datenaustauschs, wie im Folgenden aus der Perspektive eines Testfeldes beschrieben.

- Daten zwischen den Systemkomponenten der Schiffsführung innerhalb einer Schicht. Insbesondere wird ein automatisiertes Schiffsführungssystem nur partiell die Funktionalitäten des Schiffsführers umsetzen oder unterstützen und benötigt entsprechende Daten und Funktionselemente der anderen Komponenten durch das Testfeld.

- Daten aus der Umwelt und des Eigenschiffes für die drei Schichten der Route, des Verkehrs und der Kontrolle, die durch entsprechende Sensoren gemessen werden. Ein Testfeld erfüllt diese Funktion, verarbeitet diese Daten und leitet sie an das zu testende System weiter.
- Befehle zur Steuerung des Testträger Schiffes für die notwendigen Aktoren, um die Ergebnisse der Kontrollalgorithmen in mechanische Kräfte zu überführen.

Diese Informationsflüsse (Pfeile) zwischen den Komponenten (Boxen) wurden bereits in Abbildung 4 in dem Kapitel 2.1.2 dargestellt. Zur Verbesserung der Algorithmen werden die Daten oftmals nicht isoliert interpretiert und benötigen intelligente Datenfusionsmechanismen, um die Genauigkeit und Robustheit des Ergebnisses zu erhöhen. Beispielsweise werden für die Positionsbestimmung der detektierten Schiffe in der unmittelbaren Nähe sowohl die Beobachtungen einer optischen Kamera, eines AIS und Radarsystems fusioniert, um zu einem validen und genauen Ergebnis des erfassten Objektes zu gelangen [LâVK15]. Diese zunehmende Datenverarbeitung im Rahmen automatisierter Schiffsführungssysteme führt zu steigenden Herausforderungen bezüglich des seeseitigen und küstenseitigen Datenaustauschs und der dezentralen Datenverarbeitung [SKKR11]. Für den Datenaustausch dieser Systeme existieren verschiedene die Schnittstellen spezifizierende Standards. Einige für diese Arbeit relevante Standards werden im Folgenden erläutert.

**Digitale Schnittstellen für Schiffsführungssysteme.** Die IEC 61162 als eine Sammlung von Standards für „Digitale Schnittstellen für Navigationsausrüstung innerhalb eines Schiffes“ in der aktuellsten Version aus dem Jahr 2018 umfasst relevante Schnittstellenspezifikationen und ist in vier Teile untergliedert [Iec18a]. Diese stellen die Anforderungen an die Datenkommunikation zwischen den Komponenten für elektronische Instrumente, Navigationseinrichtungen und Funkkommunikationsgeräte für die Schifffahrt dar.

- IEC-61162-1 beschreibt die serielle Ein-Weg-Datenübertragung von einem Datensender zu einem oder mehreren Datenempfängern. Die Daten liegen in dem NMEA 0183 konformen Datenformat vor, das durch den American Standard Code for Information Interchange (ASCII) repräsentiert ist. Die Länge einer Datennachricht (engl. „Sentence“) umfasst elf bis höchstens 79 Zeichen und ist durch die niedrige Übertragungsgeschwindigkeit mit einer Symbolrate von 4800 Baud nur begrenzt für datenverarbeitungsintensive Anwendungen einsetzbar. Es erfolgt keine sichergestellte Datenübertragung und eine eingeschränkte Fehlerüberprüfung [Iec16].
- IEC 61162-2: Dieser Standard basiert auf dem IEC 61162-1, erweitert diesen jedoch um eine höhere Symbolrate von 38.400 Baud für die Datenübertragung [Iec98].
- IEC 61162-3: Beschreibt die auf dem Controller Area Network (CAN) basierende serielle Datenübertragung zwischen mehreren Datensendern und Datenempfängern in einem Netzwerk. Die Erweiterung des CAN basiert auf dem NMEA 2000 Standard. Die einer Parameter Gruppennummer (PGN) zugeordneten Datennachrichten werden als eine Serie von Datenframes übertragen, durch die eine robuste Fehlerbehandlung, bestätigte

Datenzustellung und garantierte Latenzzeiten ermöglicht wird. Der IEC 61162-3 definiert Datenformate, Netzwerkprotokolle und die physikalische Verbindung. Die Datenübertragungsrate beträgt 250 Kbit/s [Iec08].

- IEC 61162-4: Spezifiziert ein auf Ethernet basierendes Kommunikationsprotokoll, eine Schnittstellenbeschreibungssprache und eine Menge von Standardschnittstellenspezifikationen für die Integration verschiedener Systeme, wie ein Radarsystem, ECDIS oder die Schiffsführungsanzeige. Hierdurch erfolgt die Datenübertragung mit hoher Datenübertragungsrate zwischen mehreren Datensendern und mehreren Datenempfängern sowohl schiffsintern und zwischen anderen Systemen sowie Schiffen. Das Datenformat basiert auf NMEA 0183 und unterstützt eine Datenübertragungsrate von bis zu 100 Mbit/s. Der IEC 61162-4 liegt in sukzessive aufeinander aufbauenden Varianten vor, welche die Spezifikation dieses Standards erweitern. Die aktuellste Version ist der IEC 61162-460, der durch die vorherigen Standards der IEC 61162-4 ergänzt wird [Iec18a].

Neben diesen weit verbreiteten Standards existieren weitere proprietäre oder für spezielle Anwendungszwecke einzusetzende. Beispielsweise dient das von der Europäischen Organisation zur Sicherung der Luftfahrt veröffentlichte Protokoll mit der Bezeichnung All Purpose Structured Eurocontrol Surveillance Information Exchange (ASTERIX) dem Austausch von Radar-, Zeitsynchronisations- und Wetterdaten zur Überwachung eines Raumes [Enge16]. Für den Informationsaustausch zwischen landseitigen VTS leitet ein Standardisierungsgremium gegenwärtig die Entwicklung des Inter VTS Exchange Format (IVEF). Dieser zukünftige Standard wird den Austausch von Echtzeitinformationen hinsichtlich eines Lagebilds des Schiffsverkehrs zwischen internationalen VTS und weiteren Anwendern ermöglichen [Iala00]. Die notwendige Beachtung vielfältiger Standards ist hinsichtlich der Schnittstellen für die Datenübertragung erkennbar.

Für die Kopplung der zuvor vorgestellten Schnittstellen der Komponenten eines physikalischen Testfeldes ist ein weiterer anspruchsvoller Umstand die im physikalischen Testfeld eingesetzte Kommunikationstechnologie, welche Systeme auf See und an Land sowie untereinander auf See verbindet. Während für landseitige Transportsysteme eine zuverlässige und flächendeckende Breitbandverbindung zur Verfügung steht, sind die seeseitigen Bereiche teilweise schwach oder gar nicht durch IP-basierte Kommunikationsinfrastruktur abgedeckt [HBFF16]. Obligatorisch ist eine kontinuierliche landseitige Überwachung der Testfeld-Infrastruktur, welche für die Erprobung eines seeseitigen zu testenden Systems beispielsweise in Form eines selbstfahrenden Schiffes genutzt wird. Dennoch kann eine stabile Kommunikationsverbindung während der Fahrt nicht in allen Gewässern mit der vorhandenen Kommunikationsinfrastruktur an Bord, Land und über Satelliten garantiert werden. Ebenfalls nicht mit einer Reduktion der Nutzlast der Verbindung. So sollten Sicherheitsmechanismen im physikalischen Testfeld errichtet werden, die im Falle eines Fehlens der Verbindung das Schiff in einen sicheren Zustand bringen. Diese Sicherheitsmechanismen können im Falle eines Testfeldes beispielsweise ein Notataster oder die direkte Fernsteuerung

des Schiffes auf der Basis einer alternativen Kommunikationsverbindung sein, beispielsweise Ultrakurzwelle (UKW) [StHM08].

Das in der Schifffahrt verbreitetste Datenformat ist das durch den Standard IEC-61162-1 spezifizierte NMEA 0183. Ebenfalls Einzug hält zunehmend das durch den Standard IEC 61162-3 spezifizierte NMEA 2000. Da NMEA 2000 über eine wesentlich höhere Datenübertragungsrate verfügt, dient der Standard in dieser Arbeit als Annahmebedingung für die Schätzung der durch die Testfeld-Infrastruktur bereitzustellende Datenübertragungsrate. Für einen Datenaustausch zwischen mehreren Testfeld-Komponenten ist somit je Sensorsystemknoten eine Datenübertragungsrate von 250 Kbit/s notwendig, welche mit zunehmender Menge von Sensorsystemknoten mit der Anzahl dieser multipliziert wird.

Das physikalische Testfeld dient wie in der Zielsetzung in Kapitel 1.3 beschrieben nicht für den Test von Kommunikationstechnologie. Daher erfolgt keine detaillierte Betrachtung verschiedener etablierter Kommunikationstechnologien und deren Auswirkungen aus der Perspektive des zu testenden Systems auf die Testfeld-Architektur. Die vorgestellten digitalen Schnittstellen eines automatisierten Schiffsführungssystems standardisieren vollständig den Datenaustausch von Komponenten der in Kapitel 2.1.2 vorgestellten Schichten zur Schiffsführung. Die in diesem Kapitel aufgeführten Standards und Formate der heterogenen Systeme eines automatisierten Schiffsführungssystems führen zu Herausforderungen hinsichtlich der Interoperabilität.

## 2.1.4 Interoperabilität automatisierter Schiffsführungssysteme

Aus der in Kapitel 1.2 beschriebenen heterogenen Systemlandschaft für die Schiffsautomatisierung resultieren Herausforderungen und Anforderungen der System- und Komponentenkopplung an ein physikalisches Testfeld. Unterschiedliche Schnittstellen führen zu Interoperabilitäts Herausforderungen hinsichtlich der eingesetzten Kommunikationsprotokolle, Daten und Funktionalitäten sowohl der internen Komponenten der Infrastruktur des Testfeldes als auch bezüglich der inhärenten Schnittstelle des zu testenden Systems. Der Duden bezeichnet als Interoperabilität die Fähigkeit unterschiedlicher Systeme möglichst nahtlos zusammenzuarbeiten. Somit stellt Interoperabilität eine wesentliche Anforderung moderner Technologien dar. Die technologischen Herausforderungen der internen Testfeld-Infrastruktur sind abstrahiert von den zu testenden Systemen lösbar indem die Komponenten des Testfeldes für die Erprobung des Testgegenstands kooperieren. Im Hinblick auf wechselnde zu testende Systeme gilt es spezifische und nicht vorhersehbare Schnittstellen in interoperabler Weise durch das Testfeld zu bedienen.

Für die Bewältigung dieser Herausforderung existieren verschiedene Konzepte wie der Standard IEC 61850, die im Wesentlichen die Interoperabilität insbesondere von verteilten Automatisierungssystemen anhand verschiedener Ebenen beschreiben [Iec18b]. Interoperabilität ist auf verschiedenen Ebenen zu betrachten, wie die syntaktische oder semantische [Shet99]. Für die Harmonisierung der heterogenen Systeme der maritimen Domäne und die Integration neuer Komponenten in existierende Strukturen existiert das domänenspezifische Maritime Architecture Framework (MAF). Das MAF basiert wie das etablierte Smart Grid

Architekturmodell (engl. „Smart Grid Architecture Model“) auf den Ebenen der konzeptionellen Interoperabilität (engl. „Levels of Conceptual Interoperability“), die unter anderem der GridWise Architekturrat basierend auf dem Standard IEC 61850 beschrieb [Sgcg12, S.24ff.], [WHAD07]. Da in dem Ansatz des MAF zur Beschreibung maritimer Systeme ein etabliertes Konzept für die Betrachtung der Interoperabilität abgeleitet, erprobt und verwendet wurde folgt diese Arbeit dieser Empfehlung [Benj16], [ASRU13]. Der folgende Abschnitt beschreibt die Ebenen der Interoperabilität nach [WeHN16], [WHAD07] und [USDT13, S.19ff.].

- **Regularien und Governance** beeinflussen in Form von internationalen, nationalen und regionalen Regeln, Bestimmungen sowie Strukturen grundlegend die Systementwicklung von automatisierten Schiffsführungssystemen und somit ein Testfeld. Beispielsweise erfordert die Gestaltung eines Testfeldes die Unterstützung heutiger Zertifizierungstests wie die der DNV GL, in denen funktionelle Eigenschaften gemäß Normerfüllung verifiziert werden. Dies beeinflusst auf dieser Ebene die Kollaboration zwischen Systemen und Komponenten über Systemgrenzen hinweg unter Berücksichtigung der Strategien und taktischen Absichten zur Zielerfüllung.
- **Funktion** als eine Ebene der Interoperabilität bildet die zur Zielerfüllung durch das jeweilige System auszuführenden Tätigkeiten und Prozesse ab. Insbesondere werden auf dieser Ebene die Schnittstellen auf technologisch abstrahierter Ebene untersucht. Wegen der zuvor beschriebenen Kooperation der heterogenen Systeme erfolgt die systemübergreifende Identifikation der Funktionen und Prozesse der verschiedenen Komponenten und deren Zusammenspiel auf dieser Ebene.
- **Information** beschreibt die Durchführung der technologischen Bedarfsidentifizierung einer zuvor konzeptionell definierten Funktion. Ein spezifisches Informationsmodell legt das semantische Verständnis dar, beispielsweise durch eine Ontologie oder ein Referenzmodell. Somit stellt die Ebene der Information eine wesentliche Perspektive in der Verbindung zweier unterschiedlicher Systeme sowie der Erlangung von Interoperabilität auf der semantischen Datenebene dar. Durch die kodifizierte Semantik über die Daten erfolgt die Verwendung und Übertragung der Daten zwischen den Komponenten.
- **Kommunikation** beschreibt das syntaktische Verständnis von Daten und Mechanismen der Datenübertragung, die zwischen Anwendungen und Systemen ausgetauscht werden. An dieser Stelle wird das ISO/OSI Referenzmodell ergänzend hinzugezogen, welches durch die detailliertere Untergliederung eine ausreichende Sicht auf die technische Interoperabilität bietet. Die Ebene der Kommunikation entspricht der Anwendungs- und Darstellungsschicht des ISO/OSI Referenzmodells. Artefakte von Relevanz auf dieser Ebene sind Inhaltsstrukturen und Muster des Nachrichtenaustausches, wie Extensible Markup Language (XML), Representational State Transfer (REST) oder das Simple Object Access Protocol (SOAP). Weiterführend repräsentiert die Ebene der Kommunikation die Interoperabilität der Netzwerkkommunikation durch die Vermittlungsebene und Sitzungsebene des ISO/OSI Referenzmodells und ermöglicht somit den Nach-

richtenaustausch zwischen verschiedenen Systemen. Die Ebene der Kommunikation betrachtet neben anwendungsspezifischen Aspekten ferner die Analyse der Interoperabilität bezüglich der Transportprotokolle, wie das Transmission Control Protocol (TCP), User Datagram Protocol (UDP), IP oder IPSec.

- **Komponente** stellt als niedrigste Ebene die grundlegenden physikalischen Komponenten und deren interoperable physikalische Verbindung dar. Um wieder Bezug auf die Schichten des ISO/OSI Referenzmodells zu nehmen sind physikalische Schnittstellen der Sicherungs- und Bitübertragungsschicht ein Betrachtungsgegenstand dieser Ebene. In diesem Kontext sind die elektrischen Verbindungen, die Kodierung, Datenflusskontrolle oder Fehlerkorrektur zu betrachten, die weitläufig durch Standards wie Ethernet, WiFi oder PPP spezifiziert sind.

Diese Betrachtungsweise der Interoperabilität zweier unterschiedlicher Systeme basierend auf den Ebenen des MAF ermöglicht die Identifikation der notwendigen Maßnahmen zur technologischen Problemlösung durch die Infrastruktur des physikalischen Testfeldes.

**Interoperabilität im Kontext der e-Navigation.** Auf diese Interoperabilitäts Herausforderungen der heterogenen Systeme in der Schifffahrt reagierend veröffentlichte die IMO den strategischen Implementierungsplan für e-Navigation für den Zeitraum 2015-2019 als Ergebnis einer vorherigen Lückenanalyse. „e-Navigation ist das harmonisierte Sammeln, Integrieren, Austauschen, Darstellen und Analysieren mariner Information schiffseitig und küstenseitig mit elektronischen Mitteln zur Verbesserung der Navigation vom Anlegeplatz zum Anlegeplatz und damit verbundene Dienste für den Schutz und die Sicherheit auf See und den Schutz der marinen Umwelt“ [Imo16]. In diesem Zusammenhang strebt die IMO eine harmonisierte Erhebung und Integration maritimer Informationen und Systeme an, um einen domänenweiten Informationsaustausch und die Interoperabilität zwischen existierenden und neuen Navigationssystemen zu erlangen. Insbesondere wird in der von der IMO vorgeschlagenen datengetriebenen Architektur der e-Navigation deutlich, dass eine Anforderung an den interoperablen Datenaustausch die Verwendung des gemeinsamen maritimen Datenformates (engl. „Common Maritime Data Structure“) in Form des S-100 Framework Standards ist.

**S-100 Framework.** Als ein universelles Rahmenwerk für hydrographische Daten entwickelte die International Hydrographic Organisation (IHO) im Jahr 2010 das S-100 Framework. Dieser neue Standard ersetzt den bisherigen S-57 Standard wegen dessen fehlender Flexibilität [JHHK10]. Das S-100 Framework definiert Spezifikationen für maritime Daten sowie für die Kommunikation und ist nicht nur ein Datenmodell sondern ein ganzheitliches Framework. Die Spezifikation bietet verschiedene Richtlinien zur Datenerstellung und zum Datenaustausch zwischen Anwendungen und Systemen. In diesem Zusammenhang bietet das S-100 Framework einen domänenweit vereinheitlichten Modellierungsrahmen, der aus

dem Metamodell mit der Bezeichnung General Feature Model, den einheitlichen Datenstrukturen, einer festgelegten Terminologie, den Kodierungsformaten und Interoperabilitätsmechanismen besteht.

Durch die Kompatibilität mit webbasierten Diensten ist das S-100 Framework für den interoperablen Datenaustausch für automatisierte Schiffsführungssysteme prädestiniert und stellt eine Lösung für die Herausforderung der isolierten Komponenten dar [PaPa16, S.302]. Um mit den internationalen Geoinformationsstandards kompatibel zu sein basiert S-100 auf der ISO 19100 für die Normung von Geoinformationen. Daher unterstützt S-100 die syntaktische und semantische Interoperabilität zwischen Systemen mit maritimen und maritim-bezogenen geografischen Informationen, wie automatisierten Schiffsführungssystemen. Derzeit wird das S-100 Framework als Grundlage für verschiedene Zwecke verwendet, wie Produktspezifikationen für elektrische Navigationskarten, bathymetrische Daten oder Routeninformationen [HBFF16].

## 2.2 Systementwicklungsmethodik

Die folgenden Kapitel beschreiben ausgewählte Systementwicklungsmethoden für automatisierte Schiffsführungssysteme. Hieraus lassen sich Anforderungen an ein physikalisches Testfeld ableiten, welches den in Kapitel 2.1 beschriebenen Systemen eine Testumgebung bietet und den Systementwicklungsprozess unterstützt.

### 2.2.1 Entwicklung maritimer Automatisierungssysteme

Die Internationale Organisation für Normung (ISO) stellt in Zusammenarbeit mit der Internationalen Elektrotechnischen Kommission (IEC) und dem weltweiten Berufsverband von Ingenieuren aus den Bereichen Elektrotechnik und Informatik (engl. „Institute of Electrical and Electronics Engineers“, kurz IEEE) die internationale Norm ISO/IEC/IEEE 15288:2015 über den Lebenszyklus von Systemen bereit. Mit der Bezeichnung „System- und Software Engineering – Systemlebenszyklus-Prozesse“ (engl. „Systems and Software Engineering – System Life Cycle Processes“) definiert diese Norm die Prozesse und Terminologie für das Gestalten eines komplexen Systems. Das technische Komitee ISO/IEC JTC 1/SC7 gab diesen Standard ursprünglich im Jahr 2008 frei. Die aktuelle Version erschien in dem Jahr 2015 und bildet die Grundlage für die Entwicklung von komplexen Systemen [IsII15].

Die Systementwicklung von kooperierenden automatisierten Schiffsführungssystemen setzt fachliches Wissen in verschiedensten Disziplinen voraus und resultiert unter anderem deswegen in einer hohen Komplexität und einem umfangreichen Koordinationsaufwand. Hierfür dient das Systems Engineering als übergeordnete Methodik, welche die einzelnen Disziplinen wie Softwareentwicklung oder Regelungstechnik nicht ersetzt sondern ganzheitlich vereint. Der übergeordnete Fokus betrachtet das korrekte Zusammenspiel der Komponenten zur Realisierung der vom Benutzer gewünschten Anforderungen an das System so-

wie deren Implementierung, Erprobung und Betrieb [Weil07, S.15], [Wass05, S.1]. Das Systems Engineering unterscheidet zwischen Verifikation, Validierung und Testen. Ergänzend zu der in Kapitel 1.1 erfolgten Definition von Verifikation und Validierung bezeichnet das Testen als Untermenge der V+V die dynamische Ausführung des realisierten Systems und entsprechender Erprobung mit der Hilfe von V+V-Methoden [Enge10, S.16].

**Systementwicklung maritimer Automatisierungssysteme.** Ein domänenunspezifischer Standard für die Entwicklung von Systemen mit sicherheitskritischen Funktionen bildet die erstmals im Jahr 1998 veröffentlichte internationale Norm IEC 61508. Die Veröffentlichung der neuesten Version der IEC 61508 „Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme“ fand in der Edition 2.0 im Jahr 2010 statt. Das Ziel ist die Definition von Methoden und Verfahrensweisen für die Entwicklung von Systemen, die keine unverhältnismäßigen oder fahrlässigen Gefahren für Anwender und Umwelt bedeuten und daher eine Zertifizierung ermöglichen. Ebenso wie die zuvor vorgestellte ISO 15288 schlägt die IEC 61508 die etablierten Systementwicklungsmethoden gemäß dem Systems Engineering vor und erweitert diese um die Aspekte der funktionalen Sicherheit [Din05]. Kooperierende Automatisierungssysteme erfüllen automatisierte Funktionen in einem sicherheitskritischen Kontext, deren korrekte Funktionalität von höchster Bedeutung für das Leben eines Menschen ist und durch Methoden der Betriebssicherheit zur Risikominimierung beitragen [SmSi11]. Die normativen Standards geben hier insbesondere Anforderungen an die Prozesse für die Entwicklung dieser Systeme vor, wie näher in Kapitel 2.2.2 erörtert. Eine umfangreiche Analyse und Einteilung der Funktionen in sicherheitskritische Stufen und die Durchführung entsprechender Testmaßnahmen wird durch aktuelle Standards gefordert, insbesondere im Hinblick auf Zuverlässigkeit und Robustheit. Gemäß der jeweiligen Funktionseinstufung empfehlen die vorgestellten Standards intensive Methoden für die Entwicklung unter Zuverlässigkeitsaspekten und den umfangreichen Test der Systeme, die näher in Kapitel 2.2.3 beschrieben werden [DaHe18].

Der Standard IEC 61508 dient als Grundlage für viele branchenspezifischen Adaptionen wie die ISO 17894 der Schifffahrtdomäne oder die ISO 26262 der Kraftfahrtdomäne. Die maritime Ausprägung ISO 17894 (engl. „General Principles for the Development and Use of Programmable Electronic Systems in Marine Applications“) beinhaltet zwanzig grundlegende Prinzipien für die Entwicklung und das Testen von maritimen, programmierbaren, elektronischen Systemen. Eine Richtlinie über die relevanten Eingaben und Ausgaben eines jeden Prozesses und deren Ziel wird in Anhang C der ISO 17894 vorgeschlagen [Din05]. Im Rahmen der V+V gilt es das folgende einzuhalten:

- Jede Lebenszyklusphase sollte durch eine Verifikationsaktivität abgeschlossen werden.
- Das für die V+V zuständige Personal sollte von den für die Systementwicklung eines Lebenszyklus verantwortliche Personal unabhängig sein.
- Das integrierte System sollte gegen die Anforderungen verifiziert werden.
- Die Kriterien, Techniken und Werkzeuge welche für die V+V verwendet werden, sollten spezifiziert werden.

- Alle die funktionale Sicherheit betreffenden Funktionen sollten durch zu den Sicherheitsanforderungen nachvollziehbare Testfälle getestet werden.
- Alle Veränderungen des Systems/der Komponenten sollten einer V+V unterzogen werden [Din05, S.10].

Als Vorgehensmodell verwenden die ISO 15288, IEC 61508, ISO 26262 und ISO 17894 das etablierte V-Modell, das im Systems Engineering weit verbreitet ist [Iso16]. Das V-Modell beschreibt ein Vorgehen, das den Entwicklungsprozess gemäß der ISO 15288 und somit das Systems Engineering unterstützt [Weil07, S.19]. Das V-Modell (Wortlaut in der ISO 17894 ist „Lebenszyklus“) wird daher im folgenden Kapitel ausführlich beschrieben und dient dieser Arbeit als dem durch ein physikalisches Testfeld zu unterstützenden Systementwicklungsprozess.

## 2.2.2 V-Modell für die Systementwicklung

Das V-Modell ist ein Vorgehensmodell für die Systementwicklung, welches eine modifizierte Version des Wasserfallmodells ist. Im Gegensatz zu dem Wasserfallmodell sind die frühen Phasen des Systementwicklungsprozesses des V-Modells mit den späteren Phasen über Testfälle verbunden. Ein wichtiger Aspekt des V-Modells in diesem Kontext ist die intensive V+V der Teilprodukte gemäß den in den Definitions- und Gestaltungsphasen spezifizierten Testfällen [BaMu12]. Das V-Modell bietet sich insbesondere wegen der tiefen Einbindung einer Qualitätssicherung und eines Konfigurationsmanagements der verschiedenen Versionen im Rahmen der modellbasierten Entwicklung an, um unter anderem sicherheitskritische Systeme zu entwickeln [GaEK01, S.268], [HaHG13, S.15f].

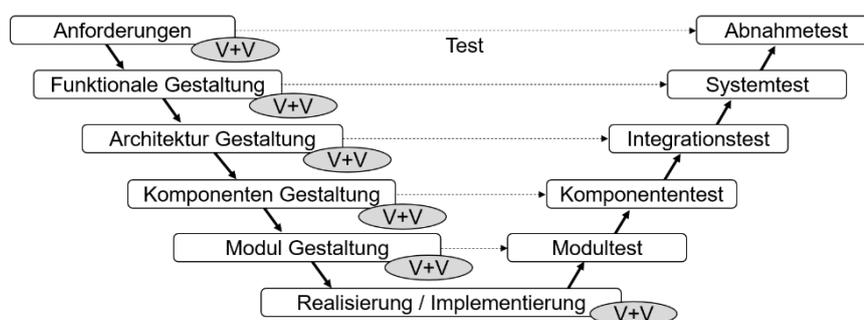


Abbildung 5: V-Modell mit Front-Loading der V+V nach [KoWa16]

Wie Abbildung 5 darstellt erfolgt zunächst die Erhebung von funktionalen und nicht-funktionalen Anforderungen. Im Kontext sicherheitskritischer Systeme liegt besonderer Fokus auf der Gefahrenabwehr (engl. „Security“) und funktionalen Sicherheit (engl. „Safety“) durch das System als Erfüllungsziel des späteren Produkts, wie in Kapitel 2.2.1 erläutert. Weiterführend wird das System auf der Basis der zuvor erhobenen Systemanforderungen in einzelne Segmente zerlegt, aus denen technische spezifischere Anforderungen resultieren. Hierauf basierend erfolgt der Entwurf der Hardware und Software, der als Lösungskonzept die Anforderungen konkretisiert und technisch spezifiziert. Auf Basis dieser konzeptionellen

Grundlage werden die Komponenten implementiert, woraufhin die linke Phase der Planung und des Entwurfs über die Implementierung in der unteren Spitze in die rechte Phase der Integration und Inbetriebnahme durch das das Testen übergeht [Wass05, S.693].

**Front-Loading der V+V.** Der Aspekt der V+V war in dem ursprünglichen V-Modell erst zu spät und nicht ausreichend integriert, weshalb diese Methoden der V+V bereits in frühen Phasen der Anforderungsdefinition und Gestaltung integriert wurden. Das Front-Loading der V+V bezeichnet Methoden, um möglichst früh die Fehler zu entdecken und diese aufwandsarm zu beheben. Methoden wie die Modellprüfung (engl. „Model Checking“) im Rahmen der modellbasierten Analyse oder das Rapid-Prototyping verbessern die Systementwicklungsleistung durch das Verschieben der Identifikation und Lösung von Problemen und Fehlern in die frühen Phasen der Systementwicklung [ThFu00], [MSRH10, S.435ff.]. Abbildung 5 illustriert die Anpassung des V-Modells mit kontinuierlicher V+V der Anforderungen und Spezifikationen bereits in der ersten Phase der Gestaltung, welche durch die in Kapitel 2.2.1 dargestellten Standards vorgeschlagen wird [KoWa16]. Neuartige Systementwicklungsansätze wie die vertragsbasierte Entwicklung (engl. „Contract based Design“) ermöglichen es, die V+V der funktionalen und teilweise nicht-funktionalen (z. B. Echtzeit) Systemeigenschaften in strukturierter oder formaler Form in die Definition und Gestaltung eines Systems zu integrieren. Die eingesetzten Verträge sind formalisierte Bedingungen für die Richtigkeit der Elementintegration, damit die nächst niedrigere Abstraktionsebene mit der folgenden übereinstimmt (horizontale Verträge). Des Weiteren werden Verträge für die Abstraktion der verfügbaren Komponenten identifiziert, damit die Realisierungen mit den Verträgen übereinstimmen (vertikale Verträge) [SaDP12], [Fire14, S.6].

**Konformitätsüberprüfung.** Die spezifikationsorientierte Überprüfung der realisierten Komponenten und Systeme dient zur Beurteilung, inwieweit die vorgegebene Umsetzung konform zu der Gestaltung ist, beispielsweise die Sicherheitsanforderungen oder Schnittstellenspezifikationen. Als Grundlage der Konformitätsüberprüfung dienen verschiedene Spezifikationsdokumente. Für einen Modultest werden die Modulspezifikationen verwendet, bei einem Integrationstest die Schnittstellenspezifikationen und bei dem Abnahmetest die beispielsweise in einem Pflichtenheft spezifizierten fachlichen Anforderungen [PCVB06]. Das Resultat des Durchlaufens dieses Systementwicklungsprozesses ist ein Gesamtsystem, das konform zu den Anforderungen und Spezifikationen ist und im Rahmen einer Typenprüfung beziehungsweise Zertifizierung für die Erfüllung der Regularien auf einem Prüfstand beziehungsweise Testfeld getestet wird (engl. „Type Approval“) [HöHR08, S.17ff.], [GrVE09, S.37ff.].

Die Zertifizierung eines Systems bedeutet die Überprüfung der Resultatdokumentation der V+V für die Sicherstellung der korrekten Funktionalität und der Test auf einem Prüfstand. Dieser Test erfolgt gemäß den in Kapitel 1.2 beschriebenen Leistungsstandards durch eine als Klassifikationsgesellschaft bezeichnete autorisierte Stelle, die bei positivem Entscheid zu einem Zertifikat und einer resultierenden Typgenehmigung führt. Dieses Zertifikat erklärt die Freigabe des Systems für eine spezifische Anwendung unter bestimmten Bedingungen [Wass05, S.655f], [WRFH15, S.275]. Klassifikationsgesellschaften (engl. „Notified

Bodies“) bezeichnen autorisierte Stellen der Schifffahrtsindustrie, die technische Standards (beispielsweise Leistungsstandards, wie den in Kapitel 1.2 vorgestellten MSC.252(83)/IEC 61924-2 für integrierte Navigationssysteme) in Zusammenarbeit mit der IMO aufbauen, pflegen und die Zertifizierung und daraus folgende Typgenehmigungen für Schiffe und Schiffssysteme durchführen. Daher verwenden die etablierten Klassifikationsgesellschaften wie DNV GL, ABS oder TÜV SÜD die Standards, welche die ISO, IEC oder die IMO in Form der Beschlüsse des Schiffssicherheitsausschusses (engl. „Maritime Safety Committee“) entwickelten. Diese legen unter anderem Anweisungen für das Testen maritimer Systeme fest. Beispielsweise spezifiziert die IEC 61174:2015 „Maritime Navigations- und Radiokommunikationsausrüstung und Systeme – Elektronisches Kartendarstellungs- und Informationssystem (ECDIS) – Betriebsbedingte und Leistungsstandards, Methoden des Testens und erforderliche Testergebnisse“ die Struktur, Leistung, Inspektion und Prüfung eines ECDIS [Iec15]. Diese Norm liefert neben anderen Informationen die funktionalen Anforderungen und Test-szenarien, die von einem Kontrollsystem im Rahmen einer Zertifizierung in einem physikalischen Testfeld erfüllt und dokumentiert werden. Das Internationale Übereinkommen von 1974 zum Schutz des menschlichen Lebens auf See (SOLAS) bietet in Anlage 5 eine Übersicht der zu erfüllenden Leistungsstandards der IMO für die Navigationsausrüstung eines Schiffes, wie ECDIS, Kompass, GNSS, Radare, AIS und weitere [Imo17a].

### 2.2.3 Statische und dynamische Verifikation und Validierung

Der Anstieg von allgegenwärtigen, eingebetteten, sicherheitskritischen Systemen unter anderem für die Schiffführung zieht hohe Anforderungen gegenüber dem Vertrauen in einen zuverlässigen Einsatz nach sich. Um die Konformität und als Ziel eine Zertifizierung dieser Systeme zu ermöglichen erfolgt während des gesamten Systementwicklungsprozesses die Anwendung von statischen und dynamischen Methoden der V+V, alternativ kombiniert als hybride Methoden [YIKG13].

**Statische Verifikation und Validierung.** Die statischen Methoden verifizieren formal das Systemverhalten früh vor der Ausführung eines realisierten Systems, um durch Fehlerprävention die Aufwände für das Finden und Beheben dieser Fehler zu verringern. Dies erfolgt durch die Prüfung des Betrachtungsgegenstands mit der Hilfe von formalen statischen Analysen im Hinblick auf die Erfüllung vorgegebener Kriterien, beispielsweise durch einen mathematischen Beweis. Die statischen Modellprüfungen dienen für die Inspektion einer Systembeschreibung in Form eines Modells gegenüber der Spezifikation, welche beispielsweise durch eine Formel repräsentiert ist [LeWi17, S.12ff.]. Wegen des sicherheitskritischen Kontextes und den zunehmend hochkomplexen automatisierten Systemen werden den statischen Methoden der V+V bereits lange in der Kraftfahrt- und Luftfahrtbranche eine besondere Bedeutung beigemessen. Eine frühe Risikoanalyse und Gefahrenabschätzung des zu entwickelnden Systems durch statische Methoden, wie die Fehlermöglichkeits- und Fehlerereignisanalyse (engl. „Failure Mode and Effects Analysis“) oder Fehlerbaumanalyse, ermöglichen die Identifikation und Vorbeugung von Fehlern im Rahmen der modellbasierten

Entwicklung [PCVB06]. Ein Testfeld unterstützt statische V+V durch die Beobachtungsfähigkeit und hierdurch Identifikation von Annahmen und Bedingungen, die in der zu erfolgenden Überprüfung genutzt werden. Dies betrifft insbesondere das empirische Sammeln und Verifizieren von Anforderungen an das zu entwickelnde System [Hahn14].

**Dynamische Verifikation und Validierung.** Bei der Anwendung der dynamischen Methoden der V+V wird eine Umsetzung des Zielsystems beziehungsweise einzelner Komponenten ausgeführt und das Verhalten in Modultests, Komponententests, Integrationstests und Systemtests in der virtuellen oder physikalischen Umgebung analysiert. Das Testen bedeutet die Ausführung eines Systems mit der Absicht Fehler zu identifizieren [LeWi17, S.11]. Die dynamischen V+V-Methoden dienen unter anderem der Eigenschaftenabsicherung auf der Basis funktionaler oder nicht-funktionaler Anforderungen, wie in Kapitel 2.2.1 beschrieben. Das dynamische Testen ist im Allgemeinen in White-Box-Tests und Black-Box-Tests unterteilt, welche die für den Test zur Verfügung stehenden Kenntnisse über das zu testende System beschreiben. Ein White-Box-Test nutzt die interne und strukturelle Ansicht des zu testenden Systems. Dies bedeutet die Überprüfung erfolgt unmittelbar am Programmcode. Ein Black-Box-Test stellt eine funktionale beziehungsweise systemexterne Ansicht dar, bei der mit der Hilfe von Spezifikationen und Anforderungen des zu testenden Systems ohne die Kenntnisse der Implementierung überprüft wird. Ein Grey-Box-Test umfasst die kombinierten Fähigkeiten des Black-Box-Tests und White-Box-Tests. Eine gängige Ebene der Durchführung von White-Box-Tests ist gemäß dem in Abbildung 6 dargestellten V-Modell der Modul- und Komponententest, der oftmals bereits von den Systementwicklern durchgeführt wird. Der Black-Box-Tests findet überwiegend auf Subsystemebene und Systemebene oftmals durch dedizierte Testingenieure statt [Enge10, S.354], [LeWi17, S.11ff.].

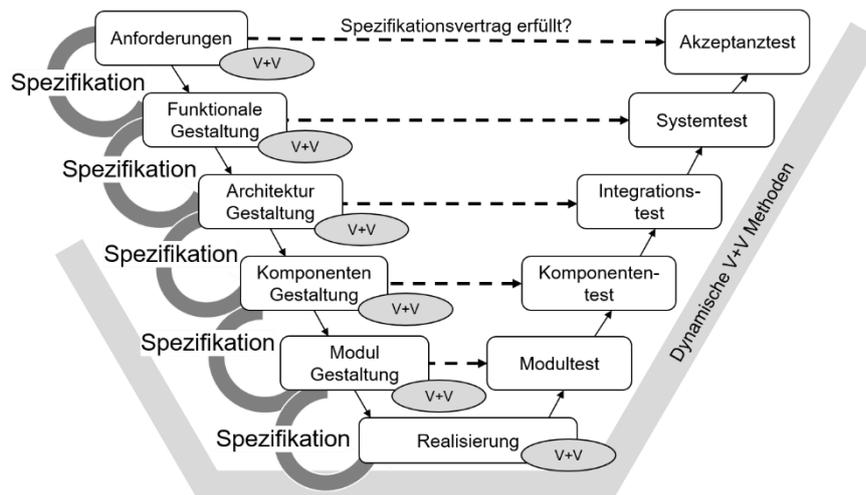


Abbildung 6: Dynamische V+V in der Gestaltungsphase und vertragsbasierte Gestaltung

Um die Systemkomplexität und hohen Aufwände für die V+V zu beherrschen erfolgt bereits im Sinne der modellbasierten Entwicklung und Analyse eine dynamische V+V während der Phasen der Definition und Gestaltung auf der linken Seite des V-Modells. Hierbei ist insbesondere durch Konzepte wie der vertragsbasierten Gestaltung eine virtuelle Integration der prototypischen und teilweise modellierten Systemkomponenten möglich, um Fehler in der

Gestaltung und Umsetzung nicht erst während der Erprobung in dem physikalischen Testfeld festzustellen, wie in Abbildung 6 dargestellt. Formale Spezifikationen und Anforderungen, deren Korrektheit durch statische V+V bewiesen wurden, werden im Rahmen der frühen virtuellen Integration auf Erfüllung bereits in der Gestaltungsphase geprüft [SaDP12].

Abbildung 7 stellt die von der ISO 26262 vorgeschlagenen Methoden der V+V für die jeweiligen Systementwicklungsphasen dar. Diese etablierten statischen und dynamischen Methoden der V+V lassen sich ebenso wegen der ähnlichen Herausforderungen in der Systementwicklung von automatisierten Schiffsführungssystemen anwenden. Hierbei finden frühe statische Methoden der V+V für die Gefahrenanalyse und Risikobewertung sowie die Verifikation des Systementwurfs statt. Weitergehend schlägt die ISO 26262 die dynamischen Methoden der V+V für die Hardwareintegration, Softwareintegration und Softwaremodultests vor, um weitergehend den umgesetzten Systementwurf durch die Verifikation der Implementierung zu testen. Abschließend erfolgt die Zertifizierung auf der Basis regulatorischer Vorgaben. Die Umsetzung der hier gelisteten Methoden der V+V gilt es durch ein physikalisches Testfeld zu unterstützen.

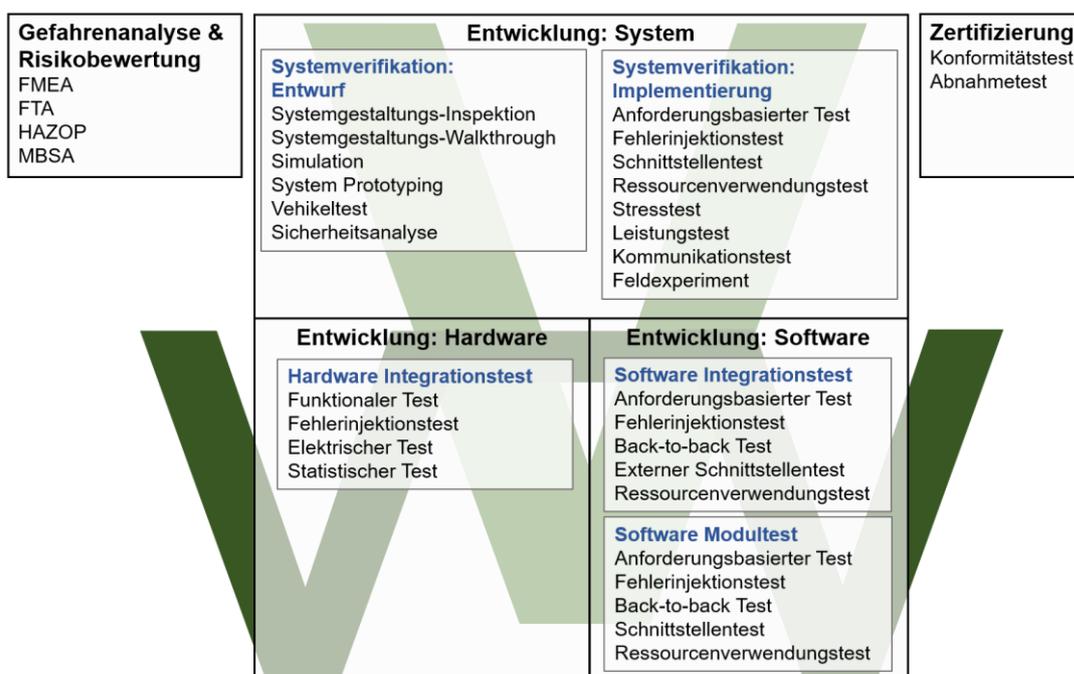


Abbildung 7: Vorgeschlagene Methoden der V+V für die Entwicklung von sicherheitskritischen Systemen der Vehikel-Führung nach [Iso16]

Um die von den Standards wie der ISO 26262 oder IEC 61508 geforderte einhundertprozentig dokumentierte Anweisungüberdeckung zu erhalten, muss ein Testfeld insbesondere Testmethoden für Randbedingungen hinsichtlich der Zuverlässigkeitsüberprüfung unterstützen, wie den Fehlerinjektionstest oder Stresstest. Besonders treten während der Testsituationen gegebenenfalls nicht alle in den Anforderungen definierten Fehlerszenarien auf, so dass die Testsituation entsprechend in den Fehlerraum gesteuert werden muss [ULRC17]. In einem virtuellen Testfeld ist das Herbeiführen der Fehlersituationen ohne weiteres möglich, während ein physikalisches Testfeld besondere Testfeld-Steuerungsausrüstung benötigt. Hierfür

injiziert die Testfeld-Infrastruktur die Fehler auf Daten-, Kommunikations- und Funktionsebene [DJMT96]. Insbesondere die Kombination verschiedener Ausnahmebedingungen gilt es im Rahmen der Fehlerinjektionstests eines automatisierten CPS zu testen. Grundsätzlich unterteilen sich Fehlerinjektionsmethoden in invasive und nicht-invasive Methoden auf, wobei durch ein physikalisches Testfeld besonders die nicht-invasiven adressiert werden. Dies erfolgt auf der Basis von softwarebasierten Verfahren, in denen die Schnittstellen des zu testenden Systems mit entsprechenden fehlerhaften Teststimuli bedient werden [ZiAV04].

## 2.2.4 Testen im Open-Loop- und Closed-Loop-Verfahren

Im Rahmen der dynamischen V+V-Verfahren werden Komponenten oder Systeme ausgeführt und im Regelkreis (engl. „Loop“) getestet. Hierfür wird das zu testende System beziehungsweise die Komponente über dessen inhärente Schnittstellen in ein simuliertes oder physikalisches Testfeld zur Bereitstellung der Testumgebung integriert. Die verschiedenen Verfahren unterscheiden sich dadurch, dass das zu testende System und die Regelstrecke jeweils real oder simuliert sind. Dieses Testen im Regelkreis ist etabliert, beispielsweise in der Entwicklung von eingebetteten Systemen zur Automatisierung von Fahrzeugen oder Schiffen und der Verwendung von sorgfältig konstruierten Testumgebungen für die jeweilige Regelkreis-Methode [KlAl11]. Abbildung 8 illustriert die Unterscheidung in Open-Loop-Verfahren und Closed-Loop-Verfahren. Beim Open-Loop-Verfahren muss ein Testfeld ein zu testendes System direkt durch Signalvektoren stimulieren und ausgehende Signalvektoren empfangen, um diese entsprechend des Testreferenzwertes zu vergleichen. Im Gegensatz hierzu verwendet ein Closed-Loop-Verfahren ein Simulationsmodell der Regelstrecke, welches die stimulierenden Signalvektoren des Testfeldes empfängt. Das Simulationsmodell erhält zudem die Ausgangssignale des Automatisierungssystems. Durch die Rückkopplung ergibt sich ein geschlossener Regelkreis, der in den Vergleich mit dem Testreferenzwert einfließt [Baum06].

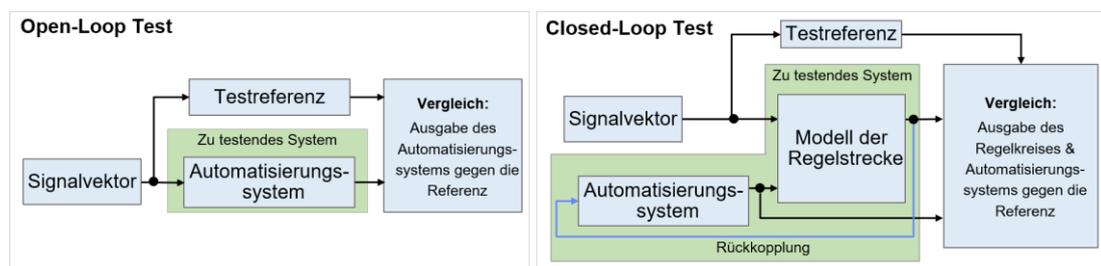


Abbildung 8: Abgrenzung zwischen Open-Loop-Test und Closed-Loop-Test nach [Baum06]

Im Kontext der modellbasierten Entwicklung erfolgt das Testen gemäß der fortschreitenden Entwicklungsstadien durch die dynamische V+V im Regelkreis, wie Model in the Loop (MIL), Software in the Loop (SIL), Hardware in the Loop (HIL) bis hin zu einer Realwelterprobung, in der die Komponenten beziehungsweise Systeme integriert und als Gesamtsystem in einem

physikalischen Testfeld erprobt werden. Mithilfe dieser Regelkreis-Methoden werden zu jeder Entwicklungsphase dynamische Tests des jeweiligen zu testenden Systems in der simulierten, emulierten oder physikalischen Umwelt durchgeführt.

Abbildung 9 stellt das beschriebene modellbasierte Vorgehen der dynamischen V+V während der Systementwicklung am Beispiel eines Autopiloten dar. Die erste Version ist ein funktionales Modell, welches durch MATLAB Simulink ausgeführt und mit der Hilfe einer simulierten Umgebung erprobt wird. Nachdem die Spezifikation erfolgte wird das funktionale Modell in eine Programmiersprache überführt und die implementierte Software durch eine simulierte Umgebung einer V+V unterzogen [FPGD16]. Der nächste Schritt des Testens installiert die Implementierung der Software auf der jeweiligen Zielhardware, die im Regelkreis in eine Simulation integriert wird. Abschließend erfolgt eine Realwelterprobung des gesamten Systems in einem physikalischen Testfeld, beispielsweise auf einem Schiff als Testträger. Die Überführung des zu testenden Systems in verschiedene Testumgebungen erfolgt nahtlos [Hahn15], [ScZu10, S.279].

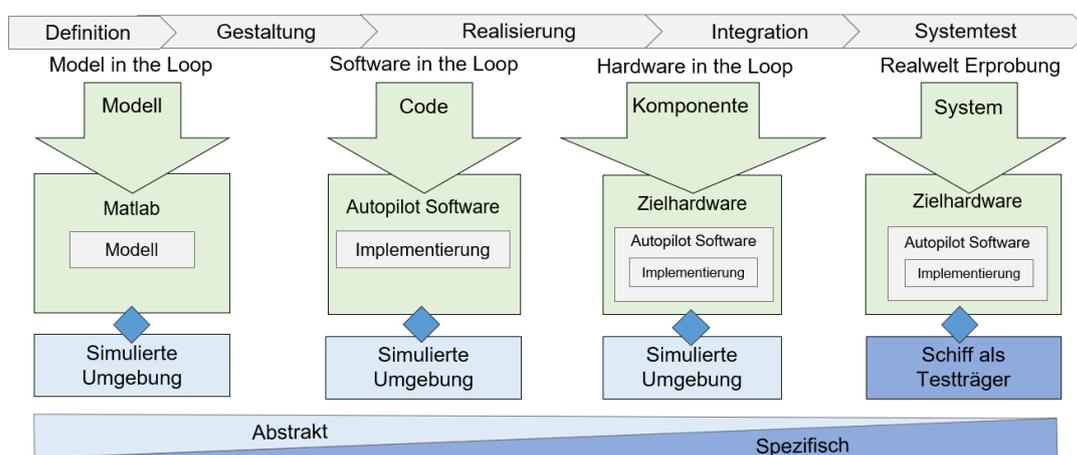


Abbildung 9: Dynamische V+V der modellbasierten Systementwicklung im Regelkreis

Das Testen im Closed-Loop-Verfahren eignet sich insbesondere für mehrere reaktive Komponenten, die als Gesamtsystem komplexe und sicherheitskritische Anforderungen in einem Szenario erfüllen müssen [ScZu10, S.277 ff.], [JHHK10].

**Szenariobasiertes Testen.** Die Komplexität der kooperierenden Teilsysteme führt dazu, dass ein isolierter Einzeltest der ein System bildenden Komponenten durch entsprechende Stimulation der zu testenden Komponente nicht ausreicht, wie in Kapitel 1.2 beschrieben. Viele Systemfehler treten erst während des Gesamtsystemtests auf. Daher bietet sich für die V+V von hochautomatisierten Systemen ein szenariobasierter Ansatz an, wie in den Projekten PEGASUS [WiWJ16] und ENABLE-S3 [NHMG17] unter anderem im Kraftfahrt- und Luftfahrtbereich erforscht. Insbesondere die Kraftfahrtdomäne sammelte positive Erfahrungen für die Erprobung des kooperativen Fahrens, in dem ähnlich wie in der Schifffahrt ein dynamischer Systemverbund aus Fahrzeugen und der Verkehrsinfrastrukturkomponenten interagiert [CoFS05]. Die wesentliche Einflussgröße der Qualität einer szenariobasierten V+V stellt die Identifikation und Auswahl der relevanten Testszenarien in Form einer Kette von Ereignissen dar, welche die Überprüfung der Anforderungen und eine Aussage über die

repräsentative Testabdeckung zulässt. Neben der Ausführung dieser Testszenarien dient ein physikalisches Testfeld insbesondere dazu die Testszenarien durch empirische Beobachtung zu identifizieren und die Eintrittswahrscheinlichkeit zu bewerten [NHMG17].

## 2.2.5 Unterstützung der V+V durch Testfelder

Wie in Kapitel 2.2.2 dargelegt erfolgt in der modellbasierten Systementwicklung bereits in frühen Entwicklungsphasen die Anwendung von Methoden der V+V zur Überprüfung der sich in der Entwicklung befindenden Gestaltungsartefakte. Hierfür dienen virtuelle und physikalische Testfelder als Simulations- oder Realweltumgebung. Wesentlicher Ansatz einer dieser Definition entsprechenden Lösung ist ein Testfeld, dass

- dem sich in der Entwicklung befindenden System eine nahtlose Testumgebung inklusive der notwendigen Schnittstellen bietet und
- die V+V der Systemelemente gemäß der etablierten Methoden ermöglicht [Wass05, S.665].

In diesem Kapitel wird die nahtlose Unterstützung der V+V unter Berücksichtigung sowohl virtueller als auch physikalischer Testmethoden für den gesamten Entwicklungslebenszyklus eines Systems vorgestellt. Da die physikalische Prüfung auf der See aufwändig ist wird der Hauptteil des Tests in virtuellen Simulationsumgebungen durchgeführt. Die Simulation ermöglicht die Replikation von nahezu Realbedingungen für automatisierte Schiffsführungssysteme, wie mit der Hilfe einer maritimen Verkehrssimulation oder einer Umweltsimulation. Zwar lässt sich durch eine Simulation ein Fehlverhalten identifizieren, jedoch garantiert eine simulative Überprüfung nicht die korrekte Funktionsfähigkeit eines Systems in der realen Umgebung.

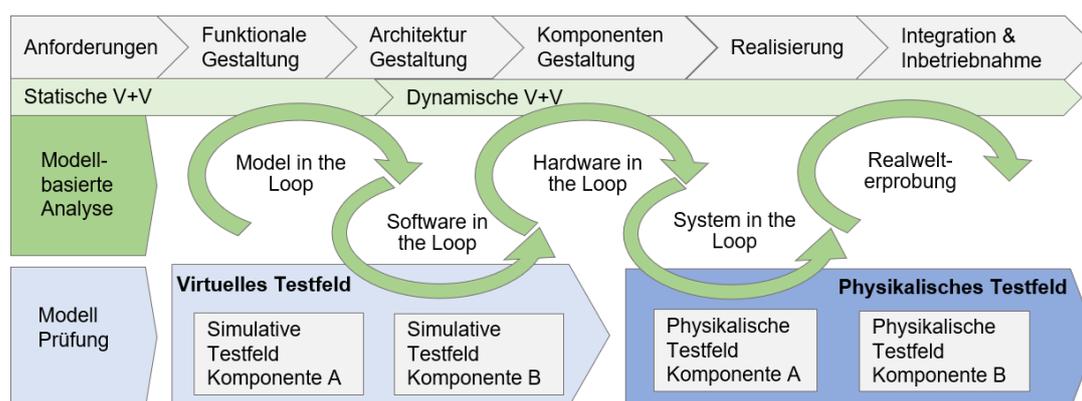


Abbildung 10: Nahtlose Unterstützung der V+V während der Systementwicklung durch Testfelder

Daher muss ein System weitergehend in der operativen Umgebung erprobt werden, um die korrekte Funktionalität zu gewährleisten und eine Zertifizierung zu ermöglichen [Hahn15]. Wie in Abbildung 10 dargestellt bieten Testfeld-Komponenten hierfür die nötige Testumgebung, wie Sensorsimulatoren oder eine Sensorbox in einer Teststrecke oder auf einem Testträger.

Die Anforderungen werden zunächst mit der Hilfe statischer V+V-Methoden überprüft, beispielsweise in dem mit der Hilfe von Beobachtungsdaten aus einem physikalischen Testfeld die Anwendungsszenarien, Annahmen und Bedingungen empirisch erhoben werden. In dieser frühen Phase finden gemäß der in Kapitel 2.2.1 vorgestellten Standards insbesondere Sicherheits- und Risikoanalysen statt, wie die FMEA. Das Ziel dieser Testmethoden ist die Identifizierung von potenziellen Sicherheitsrisiken der Systemkomponenten und deren Auswirkung auf die Funktionalität des Systems [Iec06]. Darauf folgend wird eine Umsetzung des Systems durch Prototypen unter virtuellen Bedingungen mit Simulatoren erprobt, welche als Testfeld-Komponenten durch ein virtuelles Testfeld bereitgestellt werden. Anschließend ermöglichen physikalische Testfeld-Komponenten die Realwelterprobung während eines Systemtests.

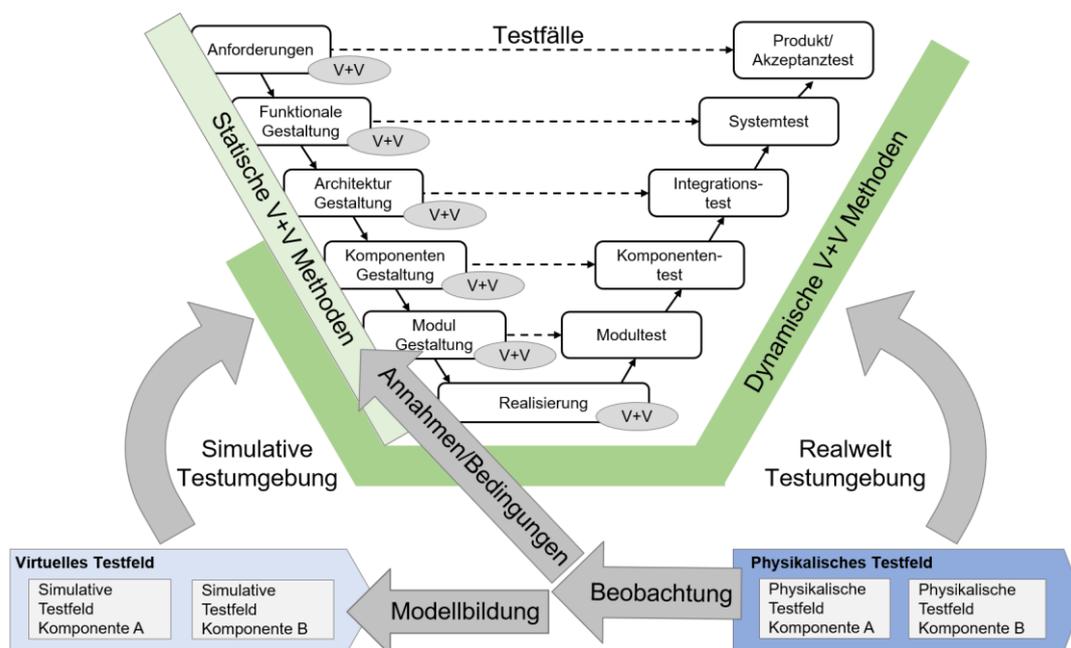


Abbildung 11: Einsatz des physikalischen Testfeldes in der Systementwicklung

Wie Abbildung 11 darstellt dient die explorative Datenerhebung der Realwelt des physikalischen Testfeldes der Modellbildung für das virtuelle Testfeld, beispielsweise für die Schiffsdynamik oder Umwelt. Mit fortschreitender Systementwicklung erfolgt das nahtlose Ersetzen der virtuellen Komponenten durch physikalische. Weiterführend erfolgt die Konformitätsüberprüfung während der Phasen der Integration und Inbetriebnahme auf der rechten Seite des V-Modells, die durch ein physikalisches Testfeld unterstützt werden [Hahn14], [Kris05].

## 2.2.6 Technologien und Konzepte eines Testfeldes

Ein Testfeld ist eine Umgebung, in der simulierte, emulierte oder physikalische Komponenten und Systeme mithilfe von Testmethoden in einem simulativen, realen oder integriert virtuell-realen Testszenario erprobt werden. Die identifizierten Testfälle lassen sich mithilfe eines Testfeldes realisieren und die Ergebnisse des Verhaltens eines zu testenden Systems

überprüfen [DWPM11]. Ein Testfeld ermöglicht die Beobachtung eines zu testenden Systems in einer kontrollierten Testumgebung. Diese Testumgebung wird mithilfe verschiedener durch Technologiekomponenten erfasster oder simulativ erzeugter Eingaben stimuliert, woraufhin das zu testende System beobachtet, untersucht und angepasst wird [AkMC08]. Eine weit verbreitete da relativ effiziente Möglichkeit zum Testen neuer Technologien stellen Simulationen dar. Beispielsweise wird der koreanische Brückensimulator in Mokpo oder die hochskalierende Umgebung für Planungs- und Kontrollalgorithmen Maritime Traffic Simulator für die Simulation von automatisierten Schiffsführungssystemen in einem realitätsnahen Test genutzt [DiHa14]. Um die korrekte Funktionalität eines Systems in der Realität zu erproben und demonstrieren werden physikalische Testfelder verwendet. Für diesen Zweck existieren verschiedene maritime Testaufbauten, die im Kontext einiger Projekte für das Testen maritimer Technologien entwickelt wurden, wie weitergehend in Kapitel 3 analysiert [Hahn15].

Gemäß der Referenzarchitektur eines Testfeldes für die Systementwicklung nach [Wass05, S.665] ist ein Testfeld als eine zentrale Rahmenstruktur zu gestalten. Anhand dieser Rahmenstruktur werden die Systemelemente in eine Testfeld-Infrastruktur integriert und die Interaktionen zwischen den Komponenten und Bestandteilen kontrolliert und gesteuert. Die Kommunikationsinfrastruktur des Testfeldes bildet das Rückgrat, das ein zu testendes System über Schnittstellenadapter mit simulativen oder physikalischen Testfeld-Komponenten und Testwerkzeugen verbindet [Wass05, S.665f]. Gemäß dem Systementwicklungsprozess werden Prototypen und frühe Varianten des Systems beziehungsweise Module und Komponenten zunächst simulativ getestet. Erreicht die Entwicklung eine akzeptable Reifestufe ersetzen physikalische Komponenten sukzessive die Modelle, Simulationen oder Emulationen durch eine möglichst nahtlose Überführung des zu testenden Systems. Dies erreicht eine einheitliche beziehungsweise integrierte Kommunikationsinfrastruktur des virtuellen und physikalischen Testfeldes [BIFa13, S.108ff.]. Generell bietet ein Testfeld eine „Einstecken und Einschalten“ artige Test-Infrastruktur, die über entsprechende Schnittstellen ein zu testendes System integriert, um diesem die nötige Testumgebung bereitzustellen [Wass05, S.667].

Im Kontext der automatisierten cyber-physischen Systeme stellte das Forschungsprojekt ENABLE-S3 eine Referenzarchitektur für ein simulatives, physikalisches und integriert virtuell-physikalisches Testfeld vor, deren Bestandteile in Abbildung 12 dargestellt sind. Eine ausführliche Beschreibung dieser Elemente übersteigt den Umfang dieser Arbeit, daher werden die drei wesentlichen Blöcke im Folgenden beschrieben. „V+V Management“ beinhaltet die Elemente zur Unterstützung der V+V entlang des Systementwicklungsprozesses inklusive der Testszenario-unspezifischen Elemente, wie eine repräsentative Realwelt-Datenbank. Diese Komponenten werden in der Regel für unterschiedliche Tests wiederverwendet. Der Block „Test Management“ repräsentiert die Elemente für die Durchführung eines konkreten Tests inklusive der notwendigen Teststeuerungs- und Beobachtungsausrüstung. Die Testplattform stellt die für die Realisierung des Tests notwendigen Komponenten dar, wie

Sensoren, Hardware oder Softwarekomponenten zur Bereitstellung der operativen Umgebung eines zu testenden Systems. Die Testplattform ist durch die speziellen Konzepte des zu testenden Zielsystems ausgeprägt [NHMG17, S.7ff.].

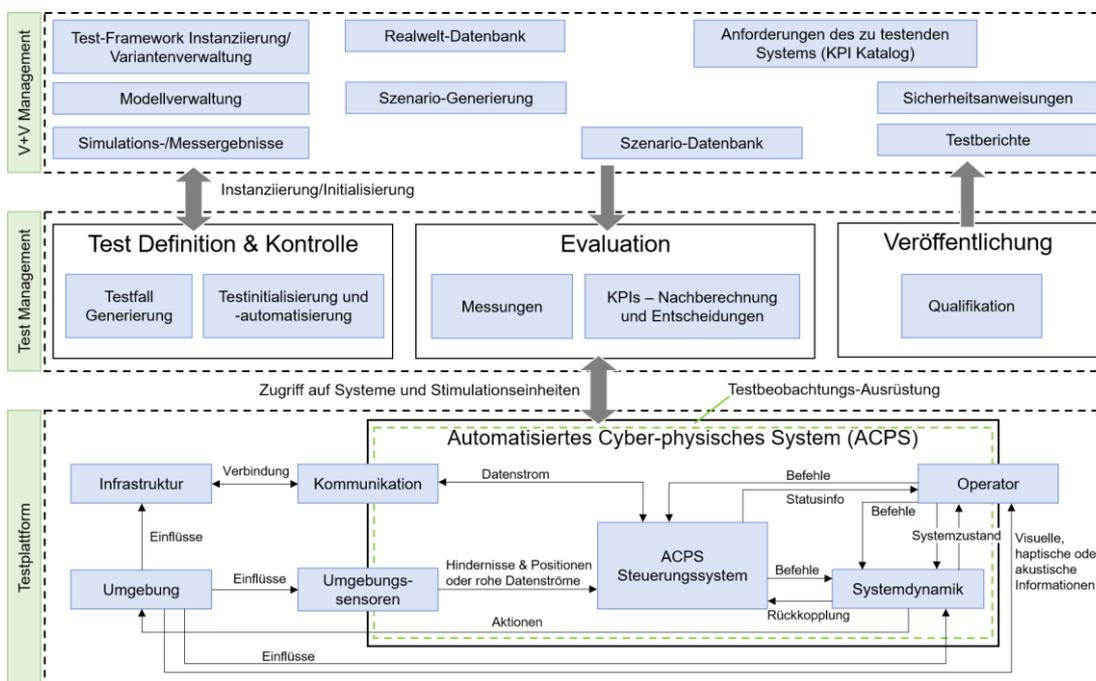


Abbildung 12: Referenzarchitektur eines generischen Testfeldes für automatisierte cyber-physische Systeme nach [NHMG17, S.7ff.]

## 2.3 Anforderungen an ein physikalisches Testfeld

In dem folgenden Kapitel werden Anforderungen an ein physikalisches Testfeld für automatisierte Schiffsführungssysteme auf der Basis der Ausführungen der vorherigen Kapitel abgeleitet. Die internationale Norm ISO 9000 beschreibt den Begriff der Anforderung als die angegebene Notwendigkeit oder Erwartung hinsichtlich der Funktionalität und Systemgüte eines Systems, die durch den Einbezug eines oder mehrerer Zielakteure als notwendig erachtet wird [RoRo06, S.5f.]. Die in diesem Kapitel beschriebenen Systemanforderungen dienen in der weiteren Arbeit für die Untersuchung der verwandten Arbeiten, als Grundlage für die Gestaltung der Testfeld-Architektur und Verifikation im Rahmen der Evaluation. Die Anforderungen leiten sich aus dem zu testenden CPS des automatisierten Schiffsführungssystems, der Systementwicklungsmethodik des Systems Engineering und den gestalterischen Anforderungen eines Testfeldes ab. Auf dieser Basis erfolgt die kategorisierende Zuordnung der identifizierten Anforderungen in den folgenden Kapiteln.

### 2.3.1 Anforderungen des zu testenden Systems

Die Anforderungen des zu testenden Systems leiten sich von den Technologien und Konzepten zur Realisierung eines automatisierten Schiffsführungssystems (Ziel 1) und der flexiblen

Integration des zu testenden Systems in die Infrastruktur des Testfeldes unter Berücksichtigung der technischen Heterogenität ab (Ziel 4). Nähere Informationen bezüglich der Konzepte des automatisierten Schiffsführungssystems als Ableitungsgrundlage der Anforderungen befinden sich in Kapitel 2.1.

**A1 – Datenbereitstellung der Schiffsumwelt.** Ein Testfeld stimuliert ein zu testendes System je nach Testszenario durch die Bereitstellung von Daten über die Schiffsumwelt der hydrodynamischen Umgebung, den Navigationsraum und das Wasserwege-Netzwerk, wie in Kapitel 2.1.2 beschrieben. Das Testfeld muss in der Lage sein mit der Hilfe der spezifischen maritimen Komponenten wie das AIS, die hydrodynamischen Sensoren, optische Kameras und weitere das für das zu testende System notwendige Lagebild der Schiffsumwelt über eine Schnittstelle bereitzustellen [Foss11, S.232ff.], [Imo17b], [JoPC16], [Iec18a].

**A2 – Datenbereitstellung des Testträger-Schiffes.** Neben der Schiffsumwelt ist die Bereitstellung der Eigendaten des Testträgers als Eingangssignale für das zu testende System über eine Schnittstelle notwendig, wie die eigene Position, Ruderlage, Maschinendaten und weitere. Insbesondere die Komponenten der Kontrolle verarbeiten gemäß dem in Kapitel 2.1.2 vorgestellten Drei-Schichten-Modell die Eigendaten des Schiffes im Regelkreis, um die Navigationsziele zu erreichen [Foss11, S.232ff.], [Imo17b], [Iec18a].

**A3 – Entgegennahme und Verarbeitung der Reaktionen des zu testenden Systems.** Das Testfeld muss eine Schnittstelle zum Empfang der Reaktionsdaten des zu testenden Systems bereitstellen. Weitergehend erfolgt die Übermittlung dieser Reaktionsdaten an die jeweilige Testfeld-Komponente sowie die Ausführung durch einen Testträger, beispielsweise Routen als Eingabe für eine Verkehrskomponente oder auszuführende Steuerbefehle für die Aktorik des Testträgers, wie in Kapitel 2.1.2 näher erörtert [Foss11, S.232ff.], [HBFF16], [SKKR11].

**A4 – Bereitstellung der Kommunikationsinfrastruktur.** Das physikalische Testfeld muss im Kontext eines CPS den seeseitigen und landseitigen Datenaustausch zwischen den Komponenten des Testfeldes geographisch verteilt realisieren, wie in Kapitel 2.1.3 beschrieben. Die Infrastruktur des physikalischen Testfeldes verbindet durch eine drahtlose Kommunikationsinfrastruktur die landseitigen und schiffseitigen Testfeld-Komponenten und bietet eine Datenverbindung für die Werkzeuge der V+V und Testdurchführung. Hierbei erfolgt in vielen Fällen die Kommunikation nicht über eine drahtgebundene und somit robuste Kommunikationsverbindung, sondern beispielsweise per WLAN oder LTE [SKKR11], [Iec18a], [StHM08]. Es gilt unterschiedliche Anforderungen an die notwendige Datenübertragungsrate und Latenz zu berücksichtigen, die je nach zu testendem System von den Führungsaufgaben und der Testmethode bestimmt werden, wie Open-Loop und Closed-Loop Test (siehe A5 und A6).

Da die Systemarchitektur des physikalischen Testfeldes die Erprobung von automatisierten Schiffsführungssystemen ermöglicht erfolgt in diesem Zusammenhang die Eingrenzung dahingehend, dass diese Arbeit nicht das Testen verschiedener Kommunikationstechnologien unterstützt. Die Kommunikationstechnologie wird als ein Bestandteil des zu testenden

Systems betrachtet. Eine Unterstützung des Tests von Kommunikationstechnologie würde den notwendigen Ausgleich der durch die notwendigen Testfunktionen resultierenden Interferenz mit der Kommunikationstechnologie bedeuten. Dies ließe sich beispielsweise durch die technologische Realisierung einer Parallelinfrastruktur lösen, um die Ausführung des zu testenden Systems nicht durch das Testen zu beeinflussen.

**A5 – Latenz.** Die durch das physikalische Testfeld bereitgestellte Kommunikationsinfrastruktur muss die durch die Systemkonzepte des Zielsystems entstehenden Anforderungen an die Latenz erfüllen. Wie in Kapitel 2.1.2 beschrieben resultieren aus den jeweiligen Führungsaufgaben unterschiedliche Latenzanforderungen. Komponenten der Routen- und Verkehrsebene kommunizieren oftmals über drahtlose Medien und basieren auf der sechs Minuten betragenden navigatorischen Entscheidungszeit [Szła12]. Das physikalische Testfeld sollte zu Testzwecken eine um den Faktor zehn bessere Leistung hinsichtlich der Latenz der drahtlosen Datenübertragung leisten, also eine Latenz von 36 Sekunden. Komponenten der Kontrollebene besitzen Anforderungen an die Latenz im Fall von Assistenzsystemen von einer Sekunde und von kleiner als 100 Millisekunden für kritische beziehungsweise automatisiert ausführende Kontrollsysteme [Dnvg11].

**A6 – Datenübertragungsrate.** Auf der Basis der in Kapitel 2.1.3 vorgestellten erwarteten Datenübertragungsrate von 250 Kbit/s je Sensorsystemknoten muss die Kommunikationsinfrastruktur des physikalischen Testfeldes unter der Annahme von vier Sensorsystemknoten je Testszenario mindestens eine Datenübertragungsrate von 1 Mbit/s bereitstellen. Die Anforderung der Datenübertragungsrate ist stark variierend und von dem jeweiligen Testszenario abhängig [Iec18a], [Iala00].

**A7 – Fehlertoleranz und Notfallmechanismen.** Da ein physikalisches Testfeld die kostenintensiven und sicherheitskritischen Komponenten wie Schiffe als Testträger verwendet, erfordert die Gestaltung der Testfeld-Architektur die Berücksichtigung der zeitlichen Restriktionen der Schiffsdynamik und des Fehlverhaltens der Testfeld-Komponenten. Die Fehlertoleranz und Notmechanismen beziehen sich ebenso auf klassische Herausforderungen der IT-Infrastruktur, wie die nötige Datenübertragungsrate, Ausfallsicherheit und Sicherheit gegenüber der bewussten Manipulation durch Dritte. Gerade die Schifffahrt ist von schwer berechenbaren und dynamischen physikalischen Einflüssen und nicht flächendeckend vorhandener Kommunikationsinfrastruktur betroffen und benötigt daher Notfallmechanismen [PhNa07].

### 2.3.2 Anforderungen der Systementwicklungsmethodik

Die methodischen Anforderungen leiten sich aus der Unterstützung der Systementwicklungsmethodik durch die V+V (Ziel 2) und dem nahtlos zu unterstützenden Systementwicklungsprozess (Ziel 3) ab. Nähere Informationen zu der Systementwicklungsmethodik als Ableitungsgrundlage der Anforderungen finden sich in Kapitel 2.2.

**A8 – Beobachtbarkeit der Umweltparameter.** Ein Testfeld muss die Umweltparameter der Durchführung eines Testszenarios für die Beobachtung bereitstellen. Sowohl für die in Kapitel 2.2.3 beschriebene Modellbildung sowie die Identifikation von Annahmen und Bedingungen für statische V+V sowie das dynamische Testen in einer Testumgebung ist die Beobachtung des Umweltszenarios notwendig. Hierbei erfolgt nicht nur die Analyse des Verhaltens eines zu testenden Systems in Bezug auf die reagierenden Ausgabedaten in Relation zu den Eingabedaten, sondern ebenfalls die dem zu Grunde liegenden Parameter des Testfalles, wie das Wetter, die hydrodynamische Umgebung, Schiffsdynamik und Objekte des maritimen Verkehrs. Insbesondere darf die Funktionalität zur Beobachtung des zu testenden Systems die Testdurchführung nicht beeinflussen [Wass05, S. 664ff.], [PCVB06], [Enge10, S.354].

**A9 – Beobachtbarkeit der Eingabe- und Ausgabedaten des zu testenden Systems.** Wie durch die ISO 17894 beschrieben (siehe Kapitel 2.2.1) ist ein entwickeltes System einer Konformitätsbewertung durch die Prüfung der Anforderungserfüllung und Eigenschaftenabsicherung gemäß den Spezifikationen zu unterziehen. Wie in Kapitel 2.2.4 vorgestellt erfolgt das Testen im Rahmen der modellbasierten Systementwicklung von eingebetteten Systemen im Regelkreis über die eingehenden und ausgehenden Schnittstellen des zu testenden Systems. Neben der Beobachtung der Umweltparameter stellt das physikalische Testfeld hierfür die Fähigkeit zur Beobachtung der an das zu testende System gesendeten Eingabedaten sowie die hieraus resultierenden Ausgabedaten bereit, um einen Black-Box-Test durchzuführen [AkMC08], [Enge10, S.354], [LeWi17, S.11ff.].

**A10 – Beobachtbarkeit der systeminternen Parameter.** Wie in Kapitel 2.2.3 beschrieben erfolgen frühe dynamische Tests auf der Ebene des Programmcodes durch White-Box-Tests, während für das Testen in späteren Entwicklungsphasen ein Black-Box-Test auf der Basis der Spezifikationen und Anforderungen erfolgt. Für das Testen gemäß einem White-Box-Test findet die Beobachtung der internen Struktur und Logik des zu testenden Systems durch das physikalische Testfeld statt. Diese Fähigkeit in Kombination mit der Beobachtbarkeit der Eingabedaten und Ausgabedaten gemäß den vorgegebenen Schnittstellen führt zu der Fähigkeit des Grey-Box-Tests, wie in Kapitel 2.2.3 gefordert. Hierfür müssen spezielle Messgeräte oder Protokollierungsgeräte installiert und über eine Schnittstelle mit dem Testfeld verbunden werden, da diese in der Regel nicht direkt vorgesehen sind [Enge10, S.354], [LeWi17, S.11ff.].

**A11 – Beobachtbarkeit der Testfeld-Infrastruktur.** Das Testfeld muss Informationen über den Zustand der Testfeld-Infrastruktur bereitstellen, die nicht direkt die Aufgabenerfüllung des zu testenden Systems betreffen. Dies umfasst die jeweilige Konfiguration des physikalischen Testfeldes zur Durchführung eines Tests sowie das Verhalten und der Zustand der Testfeld-Komponenten und Kommunikationsinfrastruktur. Insbesondere die in Kapitel 2.2.6 vorgestellte Referenzarchitektur eines generischen Testfeldes für automatisierte CPS stellt die Notwendigkeit entsprechender Ausrüstung zur Beobachtung der eingesetzten Testfeld-Infrastruktur heraus, um hierauf basierend beispielsweise Tests zu initialisieren und automatisieren [Wass05, S.665f], [NHMG17, S.7ff.].

**A12 – Evaluation der Messungen.** Wie in Kapitel 2.2.4 erörtert muss ein physikalisches Testfeld das Verhalten eines zu testenden Systems im Open-Loop und Closed-Loop-Verfahren durch die Evaluation der gemessenen Eingabedaten und Ausgabedaten des zu testenden Systems und gegebenenfalls weiterer Parameter sowie anschließendem Vergleich mit einem Referenzwert ermöglichen. Als Referenzwert für die Konformitätsprüfung dienen technische Spezifikationen sowie die Leistungsstandards der Regularien, wie in Kapitel 2.2.2 beschrieben. Die Evaluation erfolgt entweder durch eine direkte Verarbeitung (Online) oder nachgelagert auf der Basis gespeicherter Messdaten (Offline) [Wass05, S.693], [BaMu12], [NHMG17, S.7ff.].

**A13 – Dokumentation der Testausführung.** Das physikalische Testfeld muss die gemessenen und ausgewerteten Daten der Testausführung sowie Metadaten bezüglich des Test-szenarios aufnehmen und zu Dokumentationszwecken im Rahmen einer Qualifikation archivieren. Hierbei erfolgt die Dokumentation sämtlicher gemessener Daten (Eingabedaten und Ausgabedaten des zu testenden Systems, Umweltdaten des Szenarios), gegebenenfalls Informationen über das interne Systemverhalten, weiterer Informationen bezüglich der Ausführung des Testszenarios, wie Testfälle, Zeit und Ort, Bedingungen wie Umweltbedingungen, der Überprüfung zu Grunde liegenden Regularien und Normen, angewandte Testmethoden und die Konfiguration sowie der Zustand der Testfeld-Infrastruktur [HöHR08, S.17ff.], [GrVE09, S.37ff.], [WRFH15, S.275].

**A14 – Reproduzierbarkeit.** Die Ausführung eines Tests muss deterministisch und auf der Grundlage der dokumentierten Informationen der Testausführung nachvollziehbar sein. Für die Analyse auftretenden Fehlverhaltens werden umfangreiche Informationen des Testfeld-Aufbaus sowie des Testszenarios benötigt. Für eine Wiederholung des Experiments beispielsweise im Rahmen von Regressionstests muss der Testfeld-Aufbau und die Datenverarbeitungskette identisch sein, um durch die dokumentierten ganzheitlichen Daten des Testszenarios eine reproduzierbare Wiederholung durchführen zu können, wie in Kapitel 2.2.5 beschrieben [Din05].

**A15 – Nahtlose modellbasierte Systementwicklung durch Integration virtueller und physikalischer Testfeld-Komponenten.** Für die effiziente Unterstützung des Front-Loading der V+V und der modellbasierten Entwicklung durch simulative und gemischt virtuell-physikalische Verfahren, stellt Kapitel 2.2.5 an ein physikalisches Testfeld die Anforderung, das zu testende System möglichst nahtlos entlang des Systementwicklungsprozesses zu überführen und dynamische Erprobungsfähigkeiten bereits in der Entwurfs- und Implementierungsphase zu bieten. Für diesen Zweck erfolgt das sukzessive Ersetzen der simulierten Komponenten durch physikalische, wie in Kapitel 2.2.6 erläutert. Das Testfeld muss daher eine Schnittstelle zur Integration virtueller Simulatoren in die physikalische Testfeld-Infrastruktur berücksichtigen [ThFu00], [MSRH10, S.435ff.], [ScZu10, S.279].

**A16 – Manipulierbarkeit der Datenströme.** Die in Kapitel 2.2.2 vorgestellten Standards für die Entwicklung der sicherheitskritischen Systeme nutzen für die Zuverlässigkeitsüber-

prüfung umfangreiche Fehlerinjektionstests, in denen das zu testende System in der Durchführung der Testszenarien den spezifizierten Ausnahmebedingungen ausgesetzt wird. Ein physikalisches Testfeld muss hierfür eine steuerbare Infrastruktur besitzen, um durch die Manipulation auf der Basis der manuellen Eingaben oder durch die automatisierten Eingaben von Fehlermodellsimulatoren die entsprechenden Fehlerzustände herzustellen [ULRC17], [DJMT96], [ZiAV04].

**A17 – Simulationsbasierte Komplementierung des Realweltszenarios.** Wie in Kapitel 2.2.2 beschrieben erfolgt das szenariobasierte Testen von automatisierten Schiffsführungssystemen gemäß dem V-Modell anhand von Testszenarien, die auf der Basis von Rahmenbedingungen und Parametern das notwendige Testszenario spezifizieren. In der Kombination mit der unter A15 und A17 vorgestellten Integration von Simulatoren eines virtuellen Testfeldes ist das szenariobasierte Testen im Closed-Loop-Verfahren möglich, indem die vorherrschende Testsituation der Realwelt durch simulierte Daten gemäß den Anforderungen des Testszenarios komplementiert wird, wie in Kapitel 2.2.4 dargestellt [ScZu10, S.277ff.], [JHHK10], [Hahn15].

**A18 – Rückkopplung.** Um einen geschlossenen Regelkreis zur Durchführung eines Tests im Closed-Loop-Verfahren im Rahmen des szenariobasierten Testens zu erlangen muss das physikalische Testfeld das Umweltszenario der Realwelt an die Simulatoren des virtuellen Testfeldes rückkoppeln. Dies ist notwendig damit das virtuelle Testfeld eine auf das reale Umweltszenario reagierende Reaktion simulieren und an das zu testende System weiterleiten kann. Durch die Rückkopplung des physikalischen Testfeldes erfolgt die Realisierung einer kontinuierlich auf die tatsächlich vorherrschende Situation reagierende Steuerbarkeit der Testumgebung. Dies erfolgt durch die Integration virtueller und physikalischer Komponenten, wie näher in Kapitel 2.2.4 erläutert [Baum06], [KlAl11].

### 2.3.3 Anforderungen der Gestaltung

Die gestalterischen Anforderungen resultieren aus den allgemeinen Anforderungen des Testfeldes, wie in den Kapiteln 2.2.5 und 2.2.6 beschrieben. Weitergehend ergeben sich diese aus der Zielsetzung der Anpassbarkeit und Erweiterbarkeit an zukünftige Testszenarien zur Erlangung eines wiederverwendbaren und nachhaltig gestalteten physikalischen Testfeldes (Ziel 5).

**A19 – Interoperabilität.** Aus der in dieser Arbeit beschriebenen Systemkomplexität folgt, dass die kooperierenden Systeme und Komponenten unterschiedlichste inhärente physische und logische Schnittstellen aufweisen, über die sie miteinander interagieren. Die Schnittstellen der Komponenten basieren auf zahlreichen Standards oder Industriespezifikationen, wie die in Kapitel 2.1.3 beschriebenen NMEA 2000, NMEA 0183, IVEF, S-57, S-100, ASTERIX oder diverse proprietäre. Hierbei sind unterschiedliche Betrachtungsebenen vorhanden, denn die interoperabel zu gestaltenden Systeme des physikalischen Testfeldes bestehen wiederum aus Teilkomponenten, wie Sensorik, Aktorik, Software, Kommunikationstechnik [Iec18b], [Shet99], [Benj16].

**A20 – Polymorphie.** Ein physikalisches Testfeld besteht aus kostenintensiven und aufwändig zu installierenden Komponenten, die es für unterschiedliche Tests effizient bereitstellt. Entgegen der allgemeingültigen Interoperabilität innerhalb des physikalischen Testfeldes muss dieses eine dynamische Integrationskomponente besitzen. Um die dynamische Integration von heterogenen zu testenden Systemen zu ermöglichen muss ein Testfeld die technologisch unterschiedlichen Schnittstellen in polymorpher Form bedienen, wie in Kapitel 2.2.6 beschrieben. Der Begriff Polymorphie wird im Duden als „Vielgestaltigkeit, Verschieden gestaltigkeit“ bezeichnet. In der Informatik bezeichnet Polymorphie die Fähigkeit einer Software mehrere Typen für eine Variable oder einen Wert anzunehmen. Das im jeweiligen Testszenario zu testende System wird nicht während der Entwicklung und Installation des physikalischen Testfeldes integriert, sondern durch eine entsprechende Systemarchitekturgestaltung während der Laufzeit dynamisch eingebunden. Die konkrete Ausprägung der polymorphen Schnittstelle wird während der Laufzeit des Tests auf der Basis des jeweilig übergebenen Datenobjektes realisiert [BaDG06], [Hahn15], [Imo14].

**A21 – Lose Kopplung dezentraler Testfeld-Komponenten.** Die mobilen und stationären Komponenten des physikalischen Testfeldes werden geografisch verteilt zur Durchführung eines Testszenarios in die Kommunikationsinfrastruktur des physikalischen Testfeldes integriert. Für ein wiederverwendbares und gemäß verschiedener Testszenarios konfigurierbares Testfeld muss die Integration der verteilten Testfeld-Komponenten lose gekoppelt erfolgen [Hahn15]. Die Gestaltung des physikalischen Testfeldes unter Aspekten der losen Kopplung führt hierbei zu einer Erweiterbarkeit durch die Integration weiterer Testfeld-Komponenten und somit Nachhaltigkeit des Testaufbaus.

**A22 – Datenstromorientierte Kommunikation.** Die Sensoren und Aktoren des physikalischen Testfeldes sowie das zu testende System produzieren eine großvolumige Menge von Datenströmen. Das physikalische Testfeld muss eine Kommunikationsinfrastruktur bereitstellen, die in der Lage ist, die großen Datenströme zu verarbeiten und übertragen.

**A23 – Datenverfügbarkeit.** Das physikalische Testfeld muss derart gestaltet sein, dass es die Sensordaten zur Lagebilderzeugung mit einer für das zu testende System geeigneten zeitlichen Aktualität und räumlichen Abdeckung liefert. Für das Abdecken eines weiträumigen Gebietes bedarf es gegebenenfalls mobiler und stationärer Testfeld-Komponenten. Weitergehend muss die Kommunikationsinfrastruktur die zeitliche Verfügbarkeit entsprechend der Anforderungen des Testszenarios konfigurieren können, um Daten vorzuhalten beziehungsweise veraltete Daten zu verwerfen.

**A24 – Datenvielfältigkeit.** Die Gestaltung des physikalischen Testfeldes muss die Bereitstellung von unterschiedlichen Datentypen für diverse Systemtests ermöglichen, wie Videodaten, Bilder, räumliche Daten oder Audiodaten. Die Kommunikationsinfrastruktur muss in der Lage sein, die von der jeweiligen Testfeld-Komponente erzeugten Daten über die Kommunikationsinfrastruktur effektiv zu übertragen.

**A25 – Konfiguration der Datenströme.** Das physikalische Testfeld muss die Konfigurierbarkeit der Datenströme ermöglichen, in dem zum einen die Kommunikationsinfrastruktur

zentral gemäß den Testszenarien eingerichtet wird. Die Eigenkonfiguration der Kommunikationsendpunkte des jeweiligen Senders und Empfängers hingegen erfolgt dezentral durch die jeweilige Testfeld-Komponente, um insbesondere bei der Integration von weiteren Testfeld-Komponenten und im Falle einer Kommunikationseinschränkung anpassbar zu sein.

**A26 – Konfiguration der Testfeld-Komponenten.** Da die Testfeld-Komponenten geografisch verteilt und unter Umständen durch die Kommunikationstechnologie eingeschränkt sind wie beispielsweise auf Schiffen, muss das Testfeld neben einer zentralen Konfigurationsmöglichkeit ebenso eine dezentrale Konfiguration zur Laufzeit ermöglichen. Konfigurationsparameter sind beispielsweise die Aufbereitungsparameter eines Sensors, wie Abfrageintervall und Auflösung gemäß dem Testszenario.

**A27 – Benutzerschnittstelle für die Konfiguration des Testszenarios.** Das physikalische Testfeld muss eine Benutzerschnittstelle bereitstellen, über welche die Testfeld-Infrastruktur gemäß dem Testszenario konfiguriert und die Testdurchführung sowie Testergebnisse, visualisiert und beobachtet werden können.

**A28 – Logische Trennung verschiedener Tests.** Da ein wiederverwendbar und effizient gestaltetes Testfeld von mehreren Benutzern und unterschiedlichen Unternehmen parallel verwendet wird, muss das physikalische Testfeld die Mechanismen zur Datenintegrität und Sicherheit durch logische oder physikalische Trennung der verschiedenen Tests berücksichtigen [Hahn15].

## 2.4 Zusammenfassung der Grundlagen

Dieses Kapitel der vorliegenden Arbeit beschreibt den aktuellen Stand der Technik sowohl des Zielsystems in Form eines automatisierten Schiffsführungssystems als auch der physikalischen Testfelder für die V+V während der Systementwicklung. In diesem Zusammenhang betrachtet Kapitel 2.1 zunächst die Systemkonzepte des Zielsystems eines automatisierten Schiffsführungssystems. Die Grundlage hierfür bildet insbesondere die Architektur zur Führung eines Schiffes auf der Basis etablierter Konzepte der Führung, Navigation und Kontrolle. Die Entwicklung dieser Systeme unterstützt die in Kapitel 2.2 beschriebenen Entwicklungs-, Validierungs- und Verifikationsmethoden. Der Lösungsansatz des V-Modells und die jeweils hiermit zusammenhängenden statischen und dynamischen Methoden der V+V wurden beschrieben. Insbesondere erfolgte in diesem Zusammenhang die Betrachtung der Konzepte von Testfeldern, im Speziellen der generischen Testsystem-Architektur als Rahmen für die V+V automatisierter CPS.

Auf dieser Basis leiten sich systematisch die in Kapitel 2.3 identifizierten Anforderungen ab, die zur Zielerfüllung in der Gestaltung einer Testfeld-Architektur berücksichtigt werden müssen. Die Anforderungen wurden gemäß den Schwerpunktbereichen der theoretischen Grundlagenbetrachtung in die Anforderungsgruppen des zu testenden Systems, der Systementwicklungsmethodik und der Gestaltung des Testfeldes kategorisiert und den in Kapi-

tel 1.3 hergeleiteten Zielen zugeordnet. Der weitere Verlauf dieser Arbeit untersucht auf dieser Basis die verwandten Arbeiten unter Berücksichtigung der Problemstellungen, Lösungsansätze und der Anforderungserfüllung. Hierdurch werden Forschungs- und Entwicklungspotenziale für die Begründung des Handlungsbeitrages dieser Arbeit offengelegt.

## 3 Verwandte Arbeiten

Seit langer Zeit dienen maritime Testaufbauten für automatisierte Schiffsführungssysteme der Erprobung verschiedenster Systementwicklungen durch die Unterstützung unterschiedlicher V+V-Methoden. In der maritimen Domäne bezeichnet ein Testaufbau oftmals aus der Anwendungsperspektive die Gesamtheit der Dinge, die für die Durchführung eines spezifischen Testszenarios notwendig sind. Das ist aus der Sicht der in Kapitel 2.2.5 beschriebenen ingenieurwissenschaftlichen Entwicklungsmethodik nicht üblich. Daher werden die als verwandte Arbeiten identifizierten maritimen Testaufbauten in den folgenden Kapiteln auf der Basis bereitgestellter Publikationen und Dokumente kritisch systematisch analysiert und bewertet. Viele Projekte entwickeln für das Testen von speziellen maritimen Technologien simulative und/oder physikalische Testaufbauten in Seegebieten [Hahn14]. Ergänzend betrachtet diese Arbeit die Testfelder von automatisierten Transportsystemen der übrigen naheliegenden Domänen der Luftfahrt und Kraftfahrt.

In die Untersuchung eingebunden werden die folgenden von einer Lösung zu erfüllenden Kernanforderungen: Bereitstellung der operativen Umgebung (KA1), Beobachtbarkeit (KA2), Analysierbarkeit und Auswertbarkeit (KA3), Steuerbarkeit (KA4), Interoperabilität (KA5), Polymorphie (KA6), lose Kopplung (KA7) und Konfigurierbarkeit (KA8). Diese Kernanforderungen leiten sich auf der Basis der in Kapitel 2.3 erhobenen Anforderungen her, wie in Anhang B dargestellt. Durch die Analyse der Anforderungserfüllung erfolgt die Identifikation eines Handlungsbedarfs zur Beantwortung der in Kapitel 1.3 beschriebenen Zielsetzung und zur weiteren Gestaltung einer Testfeld-Architektur für automatisierte Schiffsführungssysteme.

### 3.1 Maritime Testaufbauten

Die IALA bietet eine Übersicht über existierende maritime Testaufbauten durch die Bereitstellung einer kontinuierlich administrierten Datenbank [Iala18]. Diese Datenquelle bildet die Grundlage für die Identifikation von relevanten Lösungsansätzen, die der Öffentlichkeit im Rahmen von Forschungsprojekten zugänglich sind. Die auf die Testaufbauten bezogene Recherche ergab insgesamt 46 potenzielle verwandte Arbeiten von denen sieben den Auswahlkriterien entsprachen. Das Ergebnis der Identifikation und Auswahl befindet sich nachvollziehbar aufgearbeitet in Anhang A.

**ACCSEAS** (Accessibility for shipping, efficiency advantages and sustainability) bezeichnet ein Projekt, welches einen verteilten physikalischen Testaufbau in der Nordseeregion initiierte, dessen Implementierung im Jahr 2014 für die Erprobung von digitalen Diensten und IT-Lösungen im Rahmen der e-Navigation erfolgte. Neben physikalischen Tests fand ebenfalls eine simulative Überprüfung unter anderem einer Multiquellenpositionierung, eines taktischen Routenaustauschdienstes, eines Dynamik-Prädiktors oder dem harmonisierten Datenaustausch statt. In diesem Kontext nutzte ACCSEAS den in Abbildung 13 dargestellten

temporären Aufbau der Test-Infrastruktur für die Erprobung des harmonisierten Datenaustauschs im IVEF Format zwischen einem schiffseitigen installierten, proprietären integrierten Navigationssystem und küstenseitigen VTS [PoOl15]. Das zu testende System ist der Datenverteiler, welcher das schiffseitige Lagebild erzeugt und die küstenseitige Komponente zur Vereinigung des schiffseitigen und küstenseitigen Lagebilds inklusive der Kommunikationstechnologie. Das zu testende System erhält die notwendigen Daten von einem VTS der Küstenwache und dem integrierten Navigationssystem, welche durch entsprechende Sensorsysteme das Umweltszenario erfassen und dem zu testenden System über statisch implementierte Schnittstellen bereitstellen. Das integrierte Navigationssystem sendet die erfassten Daten in dem Datenformat NMEA. Das küstenseitige VTS sendet die Daten in dem vom testenden System benötigten IVEF Format über einen statischen Adapter. Eine seekartenbasierte Darstellungsoberfläche der Referenzstrecke dient zur Validierung der Lagebilderzeugung durch das zu testende System. Diese ist sowohl schiffseitig wie küstenseitig installiert [Bill15, S.23ff.]. Die Komponenten des Testaufbaus und zu testenden Systems sind starr miteinander über spezifisch implementierte Transformationskomponenten gekoppelt.

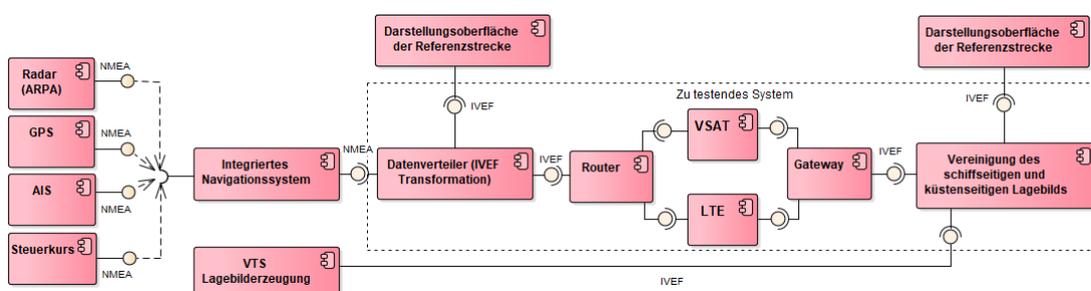


Abbildung 13: Systemarchitektur des physikalischen Testaufbaus für die Erprobung des in ACCSEAS entwickelten VTS und IVEF Informationsaustauschformats nach [Bill15, S.23ff.]

**Dublin Bay Digital Diamond** realisiert einen physikalischen Testaufbau für maritime Studien und Umweltbeobachtung zur Unterstützung der e-Navigation. In diesem Zusammenhang erfolgte die flächendeckende Abdeckung der Bucht von Dublin mit WLAN, um die Erprobung neuer digitaler Dienste für die Schiffsführung zu ermöglichen. Weitergehend erzeugen Sensoren und Kameras ein Lagebild der Referenzstrecke basierend auf Daten des Umweltszenarios und Hindernisse, die in einer Cloud gespeichert und bereitgestellt werden. Es erfolgt eine empirische Datensammlung und Lagebilderstellung, um diese nachgelagert für Tests von unter anderem automatisierten Schiffsführungssystemen bereitzustellen. Der Fokus der Gestaltung der Systemarchitektur lag auf dem Test sowohl schiffseitiger sowie küstenseitiger Dienste der e-Navigation. Insbesondere stellt Dublin Bay Digital Diamond das große Bedürfnis der Interessengruppen nach Daten heraus. Die installierte Infrastruktur versetzt die Partnergruppen wie Reedereien, Regulatoren und weiteren in die Lage deren Systementwicklungen durch die bereitgestellten Daten zu testen [Lane17], [Zhan15].

**EfficienSea** bezeichnet ein Projekt, das eine Test-Infrastruktur in der Ostsee installierte und im Rahmen des Folgeprojekts EfficienSea2 erweiterte. Mit dieser Installation werden Dienste der e-Navigation getestet, wie der Austausch von Routen zwischen der Küste und einem

Schiff sowie Schiffen untereinander. Über eine internetbasierte Technologie tauschen hierfür schiffseitige und landseitige Systeme die zur Realisierung der Dienste notwendigen Daten aus. Die Demonstration der entwickelten Technologien erfolgte mit der Hilfe einer durch das zu testende System spezifizierten Test-Infrastruktur, um die Technologien der e-Navigation insbesondere durch die Entwicklung eines Kommunikationsframeworks zu realisieren [Boru12], [KnDH09]. Abbildung 14 stellt den Testaufbau zur Erprobung eines Smart Buoy Systems dar, das die Umgebung durch Bojen-seitige Sensorik wahrnimmt und über die Maritime Connectivity Platform als dienstorientierte Infrastruktur einem VTS und schiffseitigen ECDIS bereitstellt. Es war neben der Kommunikation über die Maritime Connectivity Platform über AIS möglich, Informationen direkt mit dem ECDIS eines Schiffes auszutauschen und über die Lichtstärke der Navigationshilfe optische Signale für Schiffe auszugeben. Es existieren keine dedizierten Werkzeuge für die V+V mit denen das Systemverhalten während der Testausführung weiterführend analysiert werden könnte. Die Bewertung der Funktionserfüllung fand durch Überprüfung der Darstellung auf den Benutzeroberflächen des VTS und ECDIS statt. Das zu testende System war unmittelbar durch statische Adapter mit den Komponenten des physikalischen Testaufbaus verbunden [LeDz17].

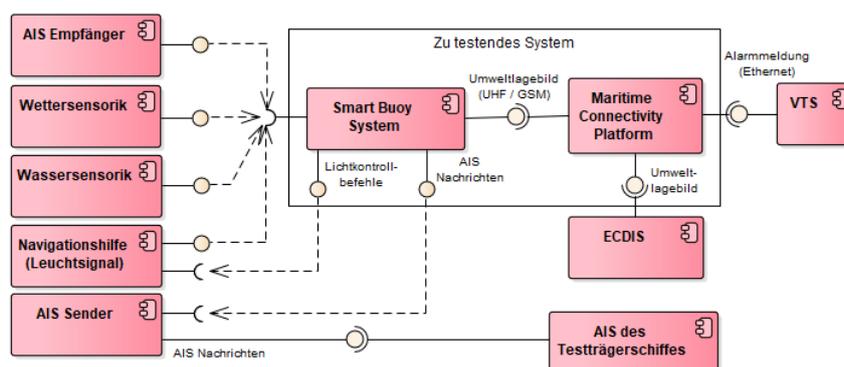


Abbildung 14: Systemarchitektur des physikalischen Testaufbaus für die Erprobung des in EfficienSea2 entwickelten Smart Buoy Systems, das als aktiv kommunizierende Komponente der Navigationshilfe und Lagebilderstellung dient nach [LeDz17]

**Hermitage** stellt die erste Bemühung Russlands dar einen physikalischen Testaufbau zur Erprobung von Technologien der e-Navigation zu installieren. Hierfür erfolgt die Errichtung sowohl seeseitiger wie flussseitiger Infrastruktur. Dies dient der Erprobung von Anwendungen für die sichere Führung der Schiffe in der Region des Überganges zwischen Fluss und See durch die Bereitstellung der vom zu testenden System benötigten Daten und Kommunikationstechnologien. Wie Abbildung 15 darstellt existieren als Komponenten der Test-Infrastruktur die Wasserstandsensoren und mehrere AIS. Für die Kommunikation dient ergänzend zu einer Funkübertragung das globale Satellitennavigationssystem GLONASS, das vom Verteidigungsministerium der Russischen Föderation betrieben wird. Nach der erfolgten Umwelterfassung der proprietären Sensorkomponenten in dem NMEA Datenformat transformiert ein statischer Adapter die erfassten Daten in ein Format gemäß den S-63 und S-57 Standards. Eine Middleware als abstrahierende Kommunikationsinfrastruktur empfängt die erfassten Daten und leitet diese an die entsprechenden zu testenden Anwendungen weiter,

die ebenfalls über einen statischen Adapter für die Datentransformation in die Test-Infrastruktur integriert werden. Als zu testende Systeme dienen ein schiffseitiges ECDIS und landseitiges VTS, auf denen Anwendungen der e-Navigation installiert sind. Eine webbasierte Anwendung des physikalischen Testaufbaus dient der Beobachtung in dem diese die durch die Test-Infrastruktur erfassten Daten des Wasserstandsensors und AIS entgegennimmt und zur Überprüfung und Beurteilung visualisiert [Isma17].

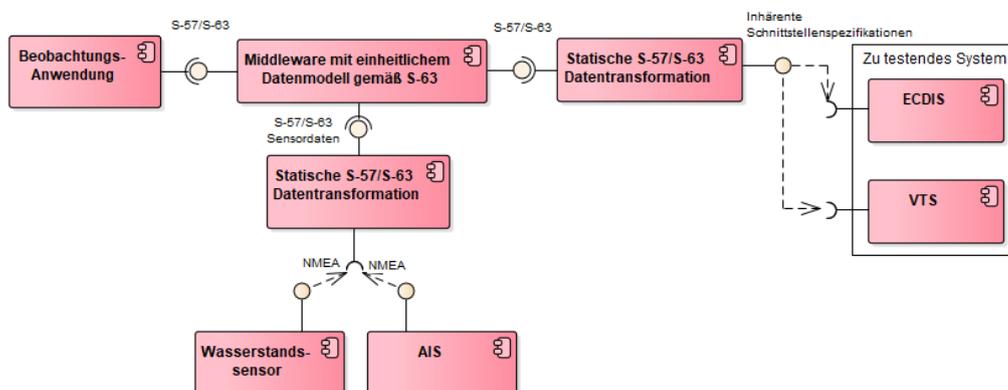


Abbildung 15: Systemarchitektur des physikalischen Testaufbaus Hermitage für die Erprobung von Anwendungen der e-Navigation nach [Isma17]

**MONALISA** (Motorways and electronic navigation by intelligence at sea) betitelt ein Projekt, welches einen maritimen Testaufbau in der Ostsee und dem Mittelmeer für das Testen für Anwendungen der Routenplanung und des Routenaustauschs zwischen ECDIS Systemen implementiert, um ein schiffsübergreifendes Lagebild des maritimen Verkehrs zu realisieren. Hierfür werden sowohl virtuelle, wie physikalische Komponenten installiert und für das Testen genutzt. Die simulative Erprobung des zu testenden Systems ermöglicht das European Maritime Simulator Network (EMSN) in einem speziellen Testszenario. Weitergehend dient der physikalische Testaufbau der Demonstration des kollaborativen Entscheidungsfindens im Hafen durch den Routenaustausch. Die Bemühungen von MONALISA werden im Sea Traffic Management Validation Projekt weitergeführt, ebenso die Implementierung der Testaufbauten als Demonstrationsumgebung. Da das Sea Traffic Management Validation Projekt unmittelbar als Folge von MONALISA die Erprobung der entwickelten Technologien darstellt erfolgt eine gemeinsame Untersuchung der Systemarchitekturen unter dem Titel MONALISA [JBBB14].

Das Projekt MONALISA erprobte einen Routenaustausch zwischen Schiffen und einem küstenseitigen VTS. Wie Abbildung 16 darstellt existieren in dem hierfür installierten physikalischen Testaufbau zwei Schiffe als Testträger, die jeweils einem schiffseitigen e-Navigation Prototype Display (EPD-ship) ECDIS als zu testendes System die notwendige Testumgebung bieten. Hierfür stellen die Testträger das AIS und GNSS bereit, deren Daten über eine statische Datentransformation an die Schnittstelle der EPD gesendet werden. Weitergehend nutzt das küstenseitige VTS implementiert durch das e-Navigation Prototype Display (EPD-shore) System als zu testendes System, ein AIS zur Erfassung des Umweltlagebilds. Diese Systeme tauschen über die Kommunikationsinfrastruktur des Testaufbaus aus-

geprägt durch die bereits in EfficienSea eingesetzten und beschriebenen Maritime Connectivity Platform die Routen untereinander aus und stellen sie dar. Durch Experten erfolgt die visuelle Beurteilung der korrekten Funktionalität dieser Systeme im Open-Loop-Test [PBJP14]. Eine dedizierte Ausrüstung für die V+V wird nicht dargestellt.

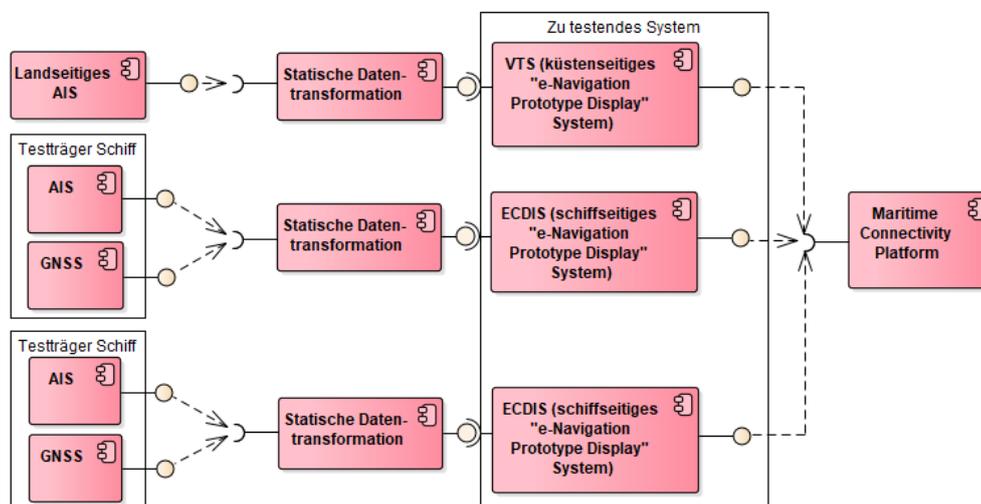


Abbildung 16: Systemarchitektur des physikalischen Testaufbaus für die Erprobung einer Anwendung für den Routenaustausch im Rahmen des Projekts MONALISA nach [PBJP14]

**SESAME Straits** (Secure, efficient, and safe maritime traffic management in the Straits of Malacca and Singapore) beschreibt einen an der Straße von Malakka lokalisierten Testaufbau. Mit der Testinfrastruktur werden Technologien der e-Navigation zur Verbesserung von hochfrequent befahrenen Seebereichen erprobt. Hierfür werden mit dem physikalischen Testaufbau spezifizierte Testszenarien durchgeführt, die Systeme und Anwendungen für das geteilte Situationsbewusstsein zwischen Schiffen und küstenseitigen VTS verbessern, um eine kooperative Entscheidungsfindung zu ermöglichen. Die Ziele des Testaufbaus umfassen die Entwicklung neuer Technologien und deren Demonstration. In diesem Zusammenhang wird eine Kommunikationsinfrastruktur für den digitalen Datenaustausch auf der Basis des S-100 Frameworks sowie eine virtuelle Schiffsverkehrssimulation realisiert [Schu17], [Kong17].

**SSAP** (Smart Ship Application Platform) bezeichnet ein Projekt, welches in den Seegebieten um Japan einen physikalischen Testaufbau für die Erprobung einer offenen Datenplattform installiert. Diese stellt Daten für den Test von Anwendungen der e-Navigation bereit. Eine Fähre und ein Tanker dienen als Testträger für die zu testenden Systeme, welche Anwendungen wie Wetterinformationen, Leistungsüberwachung von Schiffen und weitere realisieren. Das Ziel des Testaufbaus ist die Unterstützung der Systementwicklung durch die Bereitstellung des Zugriffs auf schiffseitige Sensorik und Systeme für zu testende Systeme, wie das ECDIS, der Schiffsdatenschreiber oder Maschinendateninformationen. Dies erfolgt durch die standardisierte Speicherung der Daten in einer zentralen Datenbank, deren Technologie auf dem in SSAP entwickelten ISO DIS 19848 Datenformat basiert. Die zentrale Datenbank stellt dem zu testenden System über eine IP-basierte Kommunikationsinfrastruktur die Teststimulationsdaten für die spezifizierten Anwendungen der Testszenarien bereit. Die ISO DIS 19848

standardisiert die Identifikatoren von Sensoren, ein gemeinsames Datenmodell und ein Datenformat. Weitergehend erfolgt der Aufbau und die Nutzung einer virtuellen Testumgebung, welche die Elemente des physikalischen Testaufbaus simulativ abbildet und somit eine Erprobung während des gesamten Systementwicklungsprozesses ermöglichen soll [Ando14], [Ando17].

Abbildung 17 illustriert die Systemarchitektur des für den physikalischen Test genutzten Testaufbaus. In diesem Zusammenhang fand die Nutzung umfangreicher Sensorik auf zwei als Testträger dienenden Schiffen statt. Über eine spezifisch implementierte Schnittstelle erfolgte die statische Datentransformation gemäß der jeweiligen inhärenten Schnittstelle eines jeden proprietären Sensors oder Systems.

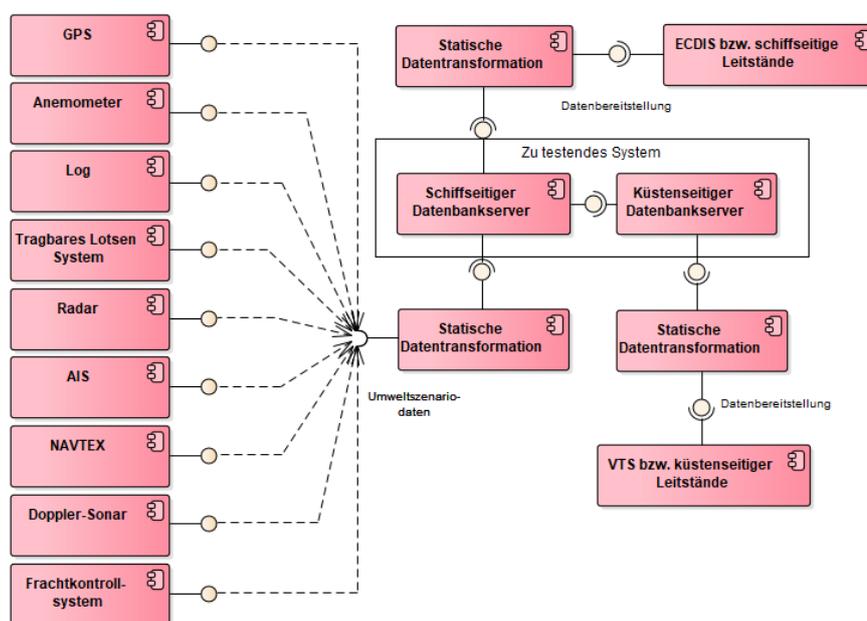


Abbildung 17: Systemarchitektur des physikalischen Testaufbaus für die Erprobung der in SSAP entwickelten Datenplattform nach [Ando17]

Das zu testende System besteht aus einer Datenplattform, die sowohl schiffseitige, wie küstenseitige Datenbankserver synchronisiert und somit die notwendigen Daten auf dem Schiff oder an der Küste über diverse Programmierschnittstellen bereitstellt. Für die Integration des zu testenden Systems erfolgte die Nutzung von statisch implementierten Adaptionern, welche die notwendigen Datentransformationen ausführen. Zur Überprüfung der Funktionsfähigkeit des zu testenden Systems dienen diverse Softwareanwendungen auf dem Testträger, die in einem ECDIS System ausgeführt und die korrekte Funktionsweise überprüft wird. Analog dient ein küstenseitiges VTS als Ausführungstechnologie für diverse Softwareanwendungen für die Funktionsüberprüfung. Die Integration sämtlicher Komponenten erfolgte in statischer Ausprägung und somit eng gekoppelt.

## 3.2 Luftfahrt- und Kraftfahrt-Testfelder

Das folgende Kapitel betrachtet eine Auswahl der relevanten Testfelder für Transportsysteme aus den Bereichen der Kraftfahrt und der Luftfahrt.

**AIM** (Anwendungsplattform Intelligente Mobilität) bezeichnet ein wiederwendbar und modular gestaltetes physikalisches Testfeld des Deutschen Zentrums für Luft- und Raumfahrt, das eine komponentenorientierten Test-Infrastruktur realisiert. Vergleichbar mit einem Baukasten erfolgt die Kombination von physikalischen Testfeld-Komponenten, wie Referenzstrecken oder Vehikel als Testträger. Ergänzt wird dies durch simulative Testfeld-Komponenten, wie ein Virtual Reality Simulator, der mit realen Kraftfahrzeugen als Testträger integriert ist. Hierdurch ist die frühe Erprobung neuer Fahrerautomatisierungssysteme möglich und der nahtlose Übergang zwischen dem simulativen und physikalischen Test [ScKr15], [FrSK12]. Die integriert simulativ und physikalisch erzeugten Daten der virtuellen und physikalischen Testfelder werden für Projektpartner durch standardisierte Schnittstellen eines Industrie-PCs als Infrastrukturzugriffsrechner einem zu testenden System als Teststimulation bereitgestellt, wie Abbildung 18 illustriert. Während der Gestaltung der Systemarchitektur des physikalischen Testfeldes lag ein Fokus auf der Modularität zur effizienten Integration weiterer Testfeld-Komponenten. Durch eine Komponente für die Datentransformation und Datenverteilung ist die Ergänzung des Testfeldes durch weitere Komponenten möglich. Die erzeugten Daten werden in ein einheitliches Datenformat gemäß IEEE 802.11p transformiert.

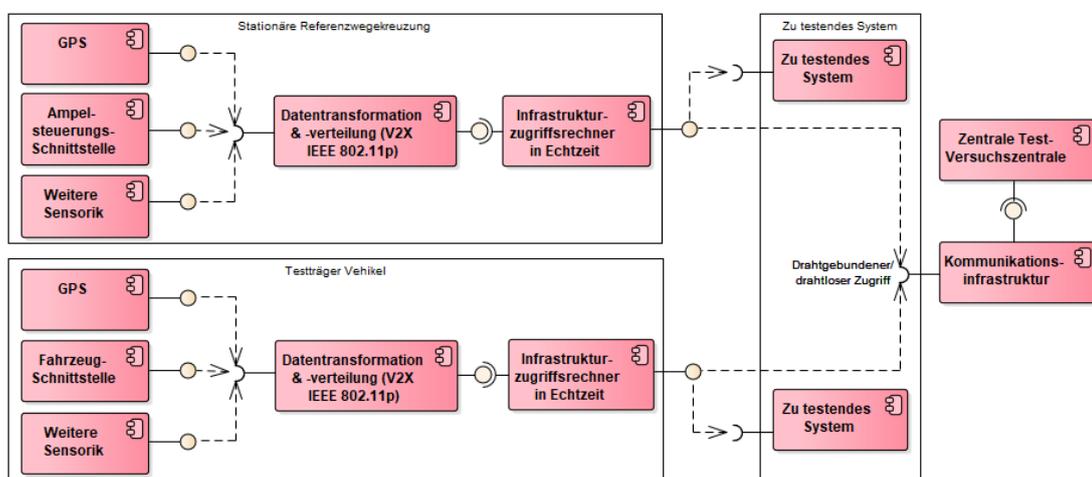


Abbildung 18: Systemarchitektur des physikalischen Testfeldes der Anwendungsplattform für intelligente Mobilität für den Test automatisierter Kraftfahrzeugführungsfunktionen nach [FrSK12]

Dieser Testaufbau besitzt sowohl stationäre Testfeld-Komponenten einer Referenzstrecke wie auch Testträger-Vehikel als mobile Einheiten. Neben der hierdurch erfolgenden Bereitstellung der Teststimulationsdaten für zu testende Systeme findet der Sendevorgang über eine Kommunikationsinfrastruktur zu der zentralen Testversuchszentrale statt. Diese ermöglicht die Beobachtung und Auswertung des jeweiligen Testszenarios durch diverse Werkzeuge der V+V [Lemm18], [FrSK12].

**Digitales Testfeld Autobahn** bezeichnet die Bemühung des Bundesministeriums für Verkehr und digitale Infrastruktur die hochfrequentierte Autobahn A9 in Bayern mit IT-Infrastruktur für die Erprobung automatisierter und vernetzter Fahrzeuge im Realbetrieb zu errichten. Das physikalische Testfeld bietet hierfür verschiedene Sensordaten von Radarsystemen, eine von einem zu testenden System nutzbare drahtlose Kommunikationsinfrastruktur und eine Digitalisierung der Strecke als hochauflösende Karte. Insbesondere erfolgt eine standardisierte Datenaufbereitung, so dass Nutzer des Testfeldes in interoperabler Weise die Testfeld-Infrastruktur zum Testen nutzen können [Bmvi18]. Ausdrücklich zielt das Digitale Testfeld Autobahn darauf ab einer Vielzahl unterschiedlicher zu testender Systeme eine Testumgebung zu bieten, wie ein intelligentes Steuerungssystem von Lastkraftwagen-Flotten (Platooning), der Umweltlagebildbereitstellung für Fahrerassistenzsysteme oder intelligente Kreuzungen [Bmvi17].

**DITCM** (Dutch Integrated Testsite for Cooperative Mobility) stellt ein physikalisches Testfeld für die Entwicklung, das Testen und die V+V von intelligenten Transportsystemen und kooperierenden Fahrertechnologien dar, ähnlich wie das Digitale Testfeld Autobahn. Für diesen Zweck erfolgte die Einrichtung einer Testfeld-Infrastruktur an der A270 und N270 zur Einrichtung von Referenzstraßen in den Niederlanden. Die Ausrüstung des physikalischen Testfeldes umfasst stationäre Komponenten für die Wahrnehmung der Umgebung durch Kameras und eine Kommunikationsinfrastruktur auf der Basis der Vehicle-to-Everything Technologie (V2X). Weitergehend findet die Nutzung dieser Daten für die Sensordatenfusion, den Test von unterschiedlichen zu testenden Systemen und der Lagebilderstellung und Bereitstellung durch ein Verkehrskontrollzentrum statt, ähnlich dem VTS in der maritimen Domäne [Tass14].

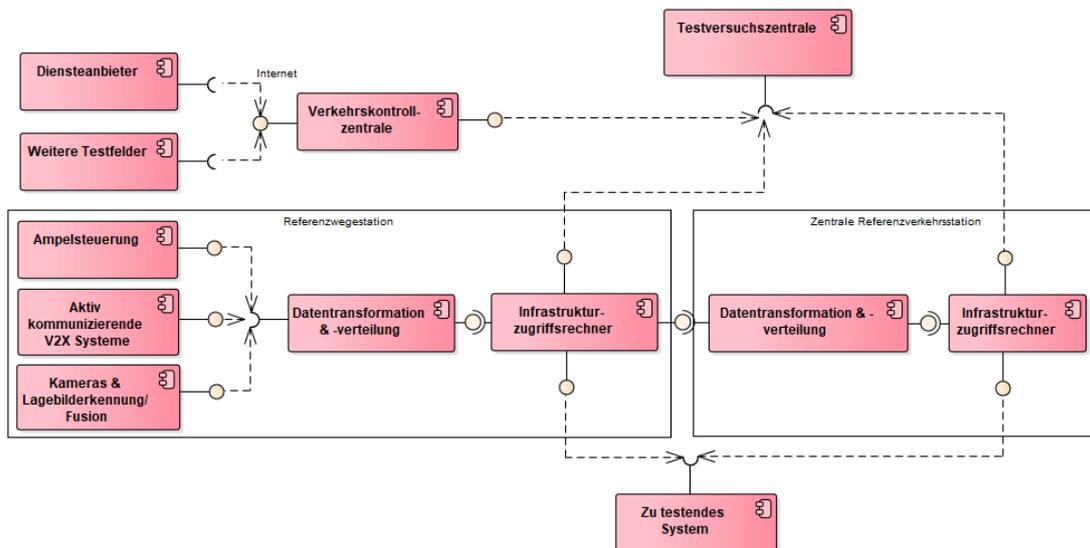


Abbildung 19: Systemarchitektur des physikalischen Testfeldes DITCM für die stationäre Erfassung des Verkehrslagebilds zur Erprobung von V2X Anwendungen nach [PNWM13]

Ergänzend bietet eine Simulation eine virtuelle Umgebung, die im Rahmen einer Laborumgebung für Fahrzeuge die X-in-the-Loop Tests ermöglicht. Für die Steuerung und Funktionalität der V+V dient eine zentrale Testversuchszentrale, die Teststeuerung, Monitoring, Protokollierung sowie Analyse- und Evaluationsfunktionalitäten bietet.

Insbesondere stellt das Testfeld DITCM die Notwendigkeit einer unabhängigen Beobachtungsausrüstung für das Testen, Validieren und Evaluieren eines zu testenden Systems heraus. Wie Abbildung 19 illustriert existieren verteilte Sensorsystemknoten, die eine Referenzwegestation bilden und lokale Umgebungsinformationen erfassen. Eine zentrale Referenzverkehrsstation fusioniert die Daten der lokalen Referenzwegestationen und ermöglicht die ganzheitliche Erfassung eines geographisch verteilten Gebietes. Ein zu testendes System wird über einen Infrastrukturzugriffrechner in die vom zu testenden System entkoppelte Testfeld-Infrastruktur integriert. Die dem Testaufbau übergeordnete Testversuchszentrale abonniert sämtliche Sensorsystemknoten, um Methoden der V+V zu unterstützen [PNWM13], [NPWM13]. In der Systemarchitektur des physikalischen Testfeldes von DITCM sticht insbesondere die lose Kopplung sämtlicher Testfeld-Komponenten, den Untersuchungswerkzeugen und des zu testenden Systems heraus.

**ATTAS** (Advanced Technologies Testing Aircraft System) bezeichnet ein modifiziertes Flugzeug als Testträger für zu testende Navigations- und Steuersysteme des Deutschen Zentrums für Luft- und Raumfahrt. In diesem Zusammenhang erfolgte die Modifikation unter anderem des mechanischen Kontrollsystems durch ein elektronisches Flugzeugsteuerungssystem und diverse Aktoren, um eine Vielzahl von verschiedenen Testszenarien unterstützen zu können. Insbesondere die Integration einer Simulationsumgebung und Kopplung mit dem Testträger ermöglicht eine nahtlose und integrierte Systementwicklung und Erprobung. Zwar ist ATTAS nicht mehr im Einsatz, eine anknüpfende Flotte unter der Bezeichnung ATRA (Advanced Technology Research Aircraft) führt den Ansatz jedoch weiter [Dlr18a].

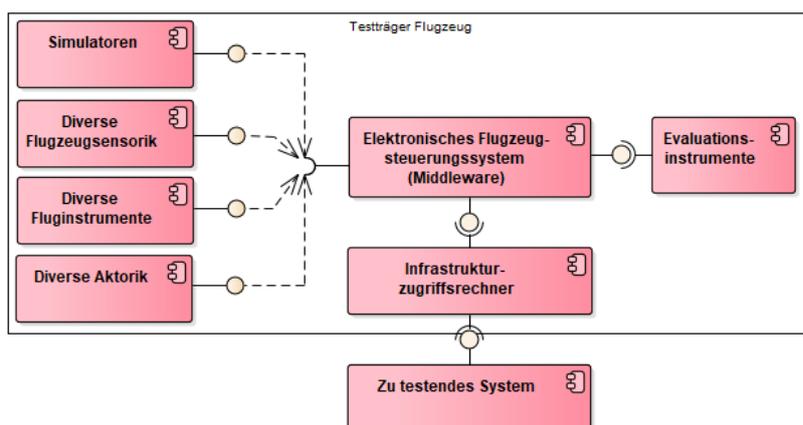


Abbildung 20: Systemarchitektur des physikalischen Testfeldes ATTAS zur Erprobung von unter anderem Flugzeugführungssystemen nach [Dlr18b]

Der Testträger bietet den verschiedenen in einem Flugzeug zum Einsatz kommenden proprietären Komponenten dedizierte Schnittstellen für den Datenexport und Datenimport mit dem elektronischen Flugzeugsteuerungssystem als abstrahierende Middleware.

Diese Middleware dient sämtlichen Komponenten des physikalischen Testfeldes zur Kommunikation und insbesondere Evaluationsinstrumente zur Durchführung der V+V. Durch das Abonnieren der Datenströme des elektronischen Flugzeugsteuerungssystems ist es den Evaluationsinstrumenten ganzheitlich möglich die Testdurchführung zu Beobachten und Analysieren. In diesem Zusammenhang erfolgt die Kommunikation mit einer Bodenstation, um die gemessenen Daten zu Protokollieren und für die weitere Analyse nutzen zu können. Bei der Gestaltung der Testfeld-Architektur liegt der Fokus auf einer modularen und flexiblen Funktionalität, um weitere Sensoren und Systeme integrieren und beliebige Testszenarien durchführen zu können. Ein Infrastrukturzugriffsrechner abstrahiert die Infrastruktur des physikalischen Testfeldes von dem zu testenden System, indem verschiedene Schnittstellen bereitgestellt werden [Dlr18b], [Holz11, S.563ff.].

### 3.3 Untersuchungsergebnisse

Nach der Identifikation und Auswahl der relevanten Testaufbauten sowie der definierten Kernanforderungen (KA) als Analysekriterien beschreibt das folgende Kapitel die Erfüllung der Anforderungen durch ein ingenieurwissenschaftlich gestaltetes physikalisches Testfeldes durch die verwandten Arbeiten.

**Abdeckung der operativen Umgebung (KA1):** Die Kernanforderung der Bereitstellung einer operativen Umgebung für die Durchführung der V+V und das Testen des jeweiligen Zielsystems erfüllen die verwandten Arbeiten unter Vorbehalt. Der überwiegende Teil der physikalischen Testaufbauten orientiert sich nicht an der Gestaltung und Bereitstellung einer generischen Testplattform, sondern spezifiziert im Vorfeld die für das Testszenario sowie Zielsystem notwendige Test-Infrastruktur und realisiert diese als eine starre Kopplung, wie ACCSEAS, EfficienSea, MONALISA oder SESAME Straits. Dieses Vorgehen bei der Gestaltung der maritimen Testaufbauten schränkt die potenziellen Testszenarien sowie insbesondere die operative Umgebung für zu testende Systeme ein und bietet resultierend nicht die von der Systementwicklungsmethodik geforderte Generizität.

In den meisten Fällen lassen sich keine systemarchitektonischen Gestaltungsentscheidungen für die flexible Erweiterung der operativen Umgebung erkennen, um eine flexible Integration von weiteren Testfeld-Komponenten in den maritimen Testaufbauten zu ermöglichen. Dies führt bereits während der Gestaltung der maritimen Testaufbauten zu einer eingeschränkt testbaren Funktionalität der zu testenden Systeme, beziehungsweise zu hohen Erweiterungsaufwänden aufgrund der starren Kopplung. Eine von den Testszenarien unabhängige und durch weitere Sensoren erweiterbare operative Umgebung bieten unter Vorbehalt die Testaufbauten Dublin Bay Digital Diamond, Hermitage und SSAP. Dies erfolgt durch die Modellierung und Zusammenführung sämtlicher erfasster Daten der Test-Infrastruktur durch ein einheitliches Datenmodell und die Nutzung einer zentralen Kommunikationsinfrastruktur als abstrahierende Middleware.

Durch die Untersuchung der physikalischen Testfelder der Kraftfahrt- und Luftfahrt domäne konnten entkoppelnde Systemarchitekturgestaltungselemente hinsichtlich der beschriebenen Erweiterbarkeit erkannt werden. Die Erweiterung der Lagebild- und Steuerungsinfrastruktur für die Bereitstellung der operativen Umgebung eines zu testenden Systems ist durch geringe Integrationsaufwände möglich auf der Basis der existierenden Testfeld-Infrastruktur im Falle von AIM, Digitales Testfeld Autobahn, DITCM und ATTAS. Das physikalische Testfeld ATTAS bietet beispielsweise das elektronische Flugzeugsteuerungssystem als abstrahierende Middleware für den vollständigen Zugriff auf das Flugnavigationssystem und die über standardisierte Schnittstellen integrierte Sensorik, Aktorik und weitere flugrelevante Komponenten. ATTAS modifizierte hierdurch den Testträger, um als ein generisches Testfeld eine große Vielzahl beliebiger Testszenarien durchführen zu können und den Testträger zu instrumentalisieren.

Im Wesentlichen nutzen die physikalischen Testfelder der Kraftfahrt- und Luftfahrt domäne für die Bereitstellung der operativen Umgebung als systemarchitektonischen Lösungsansatz ein abstrahierendes Element, das modular die Integration verschiedener Sensoren oder Systeme in die Infrastruktur des physikalischen Testfeldes ermöglicht. Insbesondere stellen die Lösungen durch diese lose Kopplung sicher, dass sich die verschiedenen Testfeld-Komponenten beispielsweise im Falle einer technologischen Änderung nicht gegenseitig beeinflussen. AIM bietet eine Komponente der Testfeld-Infrastruktur, welche für die Integration weiterer Systeme und Komponenten die Datentransformation verschiedener Implementierungen und die Datenverteilung über Ethernet umsetzt. Weitergehend ermöglicht AIM den Verbund verschiedener Sensorsystemknoten durch eine Kommunikationsinfrastruktur des physikalischen Testfeldes, um die operative Umgebung zu erweitern. Die im Kontext von AIM und ATTAS beschriebenen Lösungsansätze sind ebenfalls bei dem physikalischen Testfeld DITCM zu identifizieren.

**Abdeckung der Beobachtbarkeit (KA2):** Die Beobachtbarkeit wird von den verwandten Arbeiten überwiegend erfüllt. Der maritime Testaufbau SSAP beobachtet das Umweltszenario durch die historische Speicherung der erfassten Daten des Umweltlagebilds in einer Datenbank. Diese Daten der Navigation, Maschine und Schiffsdynamik werden zeitlich synchronisiert gespeichert und über eine nutzbare Programmierschnittstelle bereitgestellt. Eine ähnliche Funktionalität weist der maritime Testaufbau Dublin Bay Digital Diamond auf. Hermitage, MONALISA und SESAME Straits implementierten Komponenten zur visuellen Beobachtung der Testausführung und für das Aufzeichnen von unter anderem Messwerten des Umweltlagebilds und der Maschinendaten. Hermitage bietet für Testbenutzer eine Webseite zur visuellen Beobachtung des Studienbereichs. Eine visuelle Beobachtungsstation und die Darstellung auf einer elektronischen Seekarte ist die meistverwendete Beobachtungsart, mit deren Hilfe die Durchführung der Testszenarien visuell beobachtet wird. Hierbei erfolgt in vielen Fällen die Echtzeitbeobachtung der Systemtests (MONALISA, SESAME Straits, EfficienSea, ACCSEAS, AIM, ATTAS, DITCM, Digitales Testfeld Autobahn) und/oder die historische Langzeitbeobachtung (Hermitage, SSAP, AIM, Dublin Bay Digital Diamond). Es

ist zu erkennen, dass sich die maritimen Testaufbauten auf die Darstellung der Umweltszenarien und Eingabe- und Ausgabedaten des zu testenden Systems beschränken, jedoch nicht den aktuellen Zustand der Testfeld-Infrastruktur in beobachtbarer Weise ermöglichen. Hierbei findet die visuelle Beobachtung oftmals nicht durch dedizierende Werkzeuge der V+V statt, die eine weiterführende Fehleranalyse ermöglichen würden. Oftmals wird das zu testende System selbst für die Beobachtung instrumentalisiert. Eine weiterführende ganzheitliche Beobachtung des Testszenarios inklusive der Ausführungsbedingungen des Tests erfolgt nur eingeschränkt durch die verwandten Arbeiten der maritimen Testaufbauten.

Die physikalischen Testfelder der Kraftfahrt- und Luftfahrt domäne bieten umfangreiche Infrastrukturkomponenten für die ganzheitliche Beobachtung der verschiedenen Testparameter, um das Systemverhalten des zu testenden Systems sowie der Testfeld-Infrastruktur zu ermöglichen. Die Beobachtungsausrüstung in den existierenden Infrastrukturen ist generisch gestaltet und ermöglicht hierdurch hohe Flexibilität in der Auswahl der zu beobachtenden Testparameter und die Wiederverwendung in verschiedenen Testszenarien, wie die Testversuchszentrale von DITCM für umfangreiches Monitoring. AIM, DITCM und ATTAS verwenden eine dedizierte Beobachtungsinfrastruktur, welche über die Kommunikationsinfrastruktur die für die Beobachtung relevanten Daten neben den Daten für die Teststimulation des zu testenden Systems entgegennimmt.

**Abdeckung der Analysierbarkeit & Auswertbarkeit (KA3):** Die Abdeckung der Analysierbarkeit und Auswertbarkeit war durch die verwandten Arbeiten überwiegend erfüllt. Es werden Analysewerkzeuge zur Eigenschaftenüberprüfung genutzt, wie Datenprotokollierungs- und visuelle Analysewerkzeuge, um gemäß der Referenzwerte des Testszenarios die korrekte Funktionalität zu überprüfen. Die Auswertung erfolgt überwiegend durch Experten unter Zuhilfenahme einer Benutzeroberfläche, wie im Falle von ACCSEAS, SSAP und SESAME Straits. Im Rahmen von MONALISA beziehungsweise STM erfolgt die Auswertung von aufgestellten Hypothesen durch die Verwendung des maritimen Testaufbaus und einer Software zur Auswertung der Erfüllung der Annahmen. Für diesen Zweck erfolgt die Implementierung verschiedener durch das zu testende System resultierende Werkzeuge für die V+V, die auf den Testträgern, Häfen und in der Simulation installiert wurden, um die Testausführung analysieren und die Ergebnisse dokumentieren zu können. Beispielsweise in dem Kontext des Human-in-the-Loop Tests und Akzeptanztests des Projekts ACCSEAS maß eine Auswertungskomponente gemäß dem Testszenario die definierten Testparameter der Testausführung und wertete diese aus. Allgemein lässt sich identifizieren, dass zwar Analyse- und Auswertungswerkzeuge existieren, diese jedoch aus der Zielsetzung des Testszenarios resultieren und sich somit wenig flexibel beziehungsweise wiederverwendbar darstellen. Dies ist im Sinne der Systementwicklungsmethodik jedoch notwendig, wie in Kapitel 2.2 beschrieben.

Insbesondere die Testversuchszentralen von DITCM und AIM stellen eine integrierte Plattform für eine generische Protokollierung, Echtzeitanalyse und Evaluation der zu testenden Systeme dar. Hierfür erfolgt eine vollständige Protokollierung sämtlicher Eingabe-, Ausgabe- und Testfeld-Infrastrukturdaten, um insbesondere die nachträgliche Auswertbarkeit

zu ermöglichen. Für die Bewertung der Testausführung dient ein Referenz-Messsystem, anhand dessen die Testausführung beurteilt wird. Ein Referenz-Messsystem ist beispielsweise in Form eines exakte Messdaten liefernden Positionierungssystems ausgeprägt. In diesem Kontext stellt AIM heraus unter anderem durch Protokollierung und einer Datenbank für die langfristige Datenexploration gestaltet zu sein, um eine hohe Datengrundlage für die Analyse und Auswertung mit der Hilfe von Referenzdaten bereitstellen zu können. Weitergehend ermöglicht die Verknüpfung der interoperablen Datenquellen in Kombination mit einer ganzheitlichen Beobachtung des Testszenarios die Möglichkeiten der Analyse und Auswertung. Erkennbar ist bei den Systemarchitekturen der physikalischen Testfelder der Kraftfahrt- und Luftfahrt die Generizität der verwendeten Testwerkzeuge, die nachhaltig und wiederverwendbar sowie strikt von dem zu testenden System getrennt gestaltet sind.

**Abdeckung der Steuerbarkeit (KA4):** Eine nahtlose Unterstützung der Systementwicklungsmethodik und die Steuerbarkeit eines physikalischen Testfeldes durch die Integration virtueller und physikalische Komponenten erfolgt nicht in ausreichendem Maße durch die untersuchten Ansätze. Im Rahmen der Testaufbauten ACCSEAS, EfficienSea, SSAP, MONALISA und SESAME Straits findet die Nutzung von Simulatoren statt, um in frühen Systementwicklungsstadien eine gesteuerte Simulation für die Erprobung des zu testenden Systems zu nutzen. Die maritimen Testaufbauten ACCSEAS und SSAP ermöglichen theoretisch eine integrierte Nutzung physikalischer und virtueller Komponenten der Test-Infrastruktur. Dies liegt in dem einheitlichen Datenmodell entsprechend dem S-100 Framework und der resultierenden möglichen Interoperabilität begründet. Eine Nutzung integrierter virtueller und physikalischer Komponenten der Test-Infrastruktur zur nahtlosen Unterstützung stellt jedoch keine Lösung heraus, wenngleich dies insbesondere im sicherheitskritischen Kontext der automatisierten Schiffsführungssysteme für das szenariobasierte Testen notwendig wäre.

Dies führt zu einer Lücke zwischen den rein simulativen Tests der MIL, SIL und HIL Verfahren und der Erprobung in der Realwelt, die in hohen Aufwänden für die V+V resultieren und diese nicht im Sinne des Front-Loading der V+V durch die frühe Erprobung senken. Diese Argumentation weiterführend stellt sich die Frage, inwiefern die physikalischen Testaufbauten in der Lage sind, die notwendigen Testszenarien und Fehlerzustände zu realisieren, sofern ein physikalisches Umweltszenario dieses nicht ungesteuert bietet. Insbesondere AIM stellt jedoch durch das Virtual Reality Labor und die modularen Testträger-Vehikel eine integrierte Testumgebung für zu testende Systeme dar, die gemischt virtuell-physikalisch genutzt wird. Der Ansatz in DITCM berücksichtigt ebenfalls die Integration einer Simulation in eine physikalische Testfeld-Umgebung, um X-in-the-Loop Tests und insbesondere wie AIM den Test mit vollständigen physikalischen Testträgern in einer Simulationsumgebung durchzuführen (Vehicle-in-the-Loop). Eine ähnliche Gestaltung ist bei dem physikalischen Testfeld ATTAS zu erkennen, dass insbesondere den Testträger in Form eines Flugzeugcockpits identisch zu dem in der Realwelt verwendeten Testträgeraufbau in eine Simulationsumgebung integriert.

**Abdeckung der Interoperabilität (KA5):** Ein wesentlicher Teil der untersuchten maritimen Testaufbauten orientiert sich an der e-Navigation Strategie und entsprechend den darin standardisierten Harmonisierungsempfehlungen. Dublin Bay Digital Diamond, SSAP und SESAME Straits nutzen das S-100 Framework für die Datenübertragung, während ACCSEAS sämtliche erfassten Daten in den Standard IVEF transformiert. Dies ermöglicht eine partielle Interoperabilität innerhalb der internen Test-Infrastruktur und für zu testende Systeme, welche ebenso eine S-100 konforme Datenübertragung verwenden. MONALISA nutzt für einen interoperablen Datenaustausch die Maritime Connectivity Platform als eine auf einem standardisierten Protokoll basierende anwendungsorientierte Middleware. Diese Kommunikationskomponente ist jedoch gleichzeitig ein Bestandteil des zu testenden Systems und somit abhängig von dem konkreten Testaufbau. In den maritimen Testaufbauten herrscht Interoperabilität zwischen den partizipierenden Systemen, die jedoch überwiegend durch eine starke Bindung der Komponenten erreicht wurde.

Die physikalischen Testfelder der Kraftfahrt- und Luftfahrt domäne verwenden als Interoperabilitätskonzept die auf etablierten Standards basierenden Infrastrukturzugriffsrechner. Mit dieser Hilfe ist die Entkopplung der Testfeld-Infrastruktur und zu testenden Systeme beziehungsweise proprietären Testfeld-Komponenten möglich. ATTAS verwendet das elektronische Flugzeugsteuerungssystem als eine Middleware, das den Komponenten und Systemen des Flugzeuges eine entsprechende Schnittstelle bietet. Durch einen in dem Testträger installierten Infrastrukturzugriffsrechner ist die Integration des zu testenden Systems möglich.

**Abdeckung der Polymorphie (KA6):** Im Rahmen der Untersuchung der verwandten Arbeiten ließ sich kein Ansatz identifizieren, welcher die effiziente Integration verschiedener zu testender Systeme in die Infrastruktur des Testaufbaus ermöglicht. Obgleich diese Fähigkeit notwendig wäre, um die steigenden Entwicklungsaufwände der heterogenen Systeme zu senken und einen Testaufbau nachhaltig und wiederverwendbar zu gestalten. Ansätze einer Polymorphie der Infrastruktur lassen sich in einigen Ansätzen durch die Verwendung des einheitlichen Datenmodells gemäß dem S-100 Framework erkennen, das diverse Interoperabilitätsmechanismen für unterschiedlichste Systeme bietet. Nötig ist jedoch ein Gestaltungselement der Systemarchitekturen, welches dynamisch die starre Infrastruktur der Testumgebung an die inhärente Schnittstelle des zu testenden Systems anpasst. Dies ist insbesondere notwendig, da nicht jede Komponente auf dem S-100 Framework basiert. Ebenfalls weisen die Testfelder der Kraftfahrt- und Luftfahrt domäne keine entsprechenden Mechanismen zur Erhöhung der Generizität durch eine dynamische Integration der zu testenden Systeme auf. ATTAS erwähnt die Unterstützung einiger ausgewählter, auf etablierten Standards basierender Schnittstellen, fokussiert jedoch nicht die dynamische Integration der beliebigen zu testenden Systeme.

**Abdeckung der losen Kopplung (KA7):** Eine geringe Abhängigkeit der Komponenten des Testaufbaus untereinander konnte in dem überwiegenden Teil der maritimen Testaufbauten nicht festgestellt werden. MONALISA, ACCSEAS und SESAME Straits implementierten die Test-Infrastruktur ohne den Fokus auf entkoppelnde Elemente wie eine Middleware oder

Abstraktionskomponenten zu legen. Dublin Bay Digital Diamond, SSAP und Hermitage stellen ein Zwischensystem in Form einer Datenplattform bereit, welche die Komponenten des Testaufbaus voneinander räumlich und zeitlich entkoppelt. Eine lose Kopplung ist dennoch zwischen den Komponenten des Testaufbaus auf der Basis der Schnittstellen für einen modularen Testaufbau nicht in ausreichendem Maße zu erkennen. Die Schnittstellen sind überwiegend stark von der jeweiligen Implementierung abhängig und eine Änderung der Implementierung einer Komponente des Testaufbaus führt zu notwendigen Änderungen am Gesamtsystem der Test-Infrastruktur. Insbesondere sind die Komponenten der V+V, wie Benutzeroberflächen für die visuelle Beobachtung und Beurteilung, in dem überwiegenden Teil der maritimen Testaufbauten durch das zu testende System repräsentiert, wie in Kapitel 3.1 illustriert. Daher sind die Test-Infrastrukturen stark an das zu testende System gebunden.

Insbesondere in der Gestaltung der Systemarchitektur von AIM erfolgt die Berücksichtigung einer losen Kopplung durch eine Middleware, welche als Zwischenschicht die verschiedenen Komponenten voneinander entkoppelt. DITCM und ATTAS stellen ein modular gestaltetes Testfeld vor, dessen Elemente über eine entsprechende Middleware beziehungsweise Kommunikationsinfrastruktur verknüpft werden können, um als generische physikalische Testfelder eine Vielzahl von Testszenarien zu unterstützen. Die untersuchten verwandten Arbeiten der Luft- und Kraftfahrt stellen in den vorliegenden Publikationen uneingeschränkt dar, dass die Modularität und lose Kopplung der verschiedenen Komponenten der physikalischen Testfelder ein wesentliches Paradigma in der Entwicklung der Systemarchitekturen darstellt.

**Abdeckung der Konfigurierbarkeit (KA8):** Die Anforderung der Konfigurierbarkeit wird von dem wesentlichen Teil der untersuchten verwandten Arbeiten nicht ausreichend unterstützt. Das Projekt Dublin Bay Digital Diamond besitzt insofern eine erweiterbare und konfigurierbare Gestaltung, als dass der Testaufbau verschiedene Testkonfigurationen durch eine Datenspeicherungsinfrastruktur basierend auf einer Cloud und konfigurierbarer Anfragen zulässt. EfficienSea fokussiert lediglich die Möglichkeit das zu testende System zu konfigurieren. Eine Systemarchitektur des Testaufbaus mit Konfigurationsmöglichkeiten gemäß einem Testszenario ist nicht identifizierbar. MONALISA beziehungsweise STM hingegen stellen ebenso wie SESAME Straits nur für den speziellen Anwendungsfall eine Test-Infrastruktur bereit und legen keinen Fokus auf eine konfigurierbare Funktionalität zur Unterstützung unterschiedlicher Testanforderungen. Die nicht ausreichende Berücksichtigung einer Konfiguration der Kommunikationsinfrastruktur, Komponenten des Testaufbaus oder einer Benutzerschnittstelle für die Testausführung im Rahmen der Gestaltung maritimer Testaufbauten lässt sich möglicherweise dadurch begründen, dass sie für eine konkrete Testausführung realisiert wurden. Gestaltungsentscheidungen für die logische Trennung verschiedener Testinteressenten wurden nicht beschrieben.

Insbesondere bei AIM und dem Digitalen Testfeld Autobahn steht eine Konfiguration verschiedener Komponenten des Testaufbaus je nach Test im Vordergrund. AIM stellt einen zentralen Server-Pool und eine Middleware heraus, mit deren Hilfe verschiedene Test-Infrastrukturen gemäß den Testanforderungen konfiguriert werden können. DITCM unterstützt

durch die Testversuchszentrale die Konfiguration des Testaufbaus, ebenso das ATTAS Testfeld durch diverse Werkzeuge für die V+V. Die physikalischen Testfelder der Kraftfahrt und Luftfahrt sind nach dem Baukastenprinzip entworfen, um durch modulare Komponenten das jeweilige Testszenario zu realisieren und konfigurieren.

### 3.4 Zusammenfassung und Handlungsbedarf

Die verwandten Arbeiten in Form physikalischer Testaufbauten und Testfelder für automatisierte Transportsysteme beinhalten unterschiedliche Lösungsansätze und Fragestellungen. Zusammenfassend betrachtet erfüllen gegenwärtige maritime Testaufbauten nur partielle Kernfunktionen eines Testfeldes, wie die Abbildung der operativen Umgebung, Interoperabilität innerhalb des Testaufbaus, eine unter Vorbehalt ausreichend beobachtbare Test-Infrastruktur und Analyse- und Auswertungskomponenten. Jedoch adressieren die untersuchten maritimen Testaufbauten nicht eine nachhaltige sowie offene Gestaltung der Systemarchitektur und eine hohe Generizität. Die Orientierung an den aus den normativen Vorgaben resultierenden Systementwicklungsmethoden ist ebenfalls nicht ausreichend. Der überwiegende Teil der maritimen Testaufbauten weist eine starke Abhängigkeit des zu testenden Systems und der im Testaufbau genutzten Technologien auf. Durch die nicht vorhandene Trennung besitzen die Lösungen eine starre Infrastruktur und sind kurzfristig für spezielle Anwendungsszenarien gestaltet.

Ein Teil der maritimen Testaufbauten löst die Herausforderung der Interoperabilität durch die Verwendung eines einheitlichen Datenmodells gemäß S-100 und entsprechend einer lokalen Anpassung der Komponenten. Auffallend hinsichtlich der maritimen Testaufbauten ist das Fehlen einer nahtlosen Orientierung am Systementwicklungsprozess durch die Unterstützung gemischt virtuell-physikalischer Tests unter anderem zur Unterstützung des Front-Loading der V+V und des szenariobasierten Testens. Insbesondere die physikalischen Testfelder der Kraftfahrt- und Luftfahrtdomäne stellen dies als wesentliches und notwendiges Merkmal heraus. Die untersuchten maritimen Testaufbauten implizieren die Realisierung eines speziellen Test-Anwendungsszenarios. Die Gesamtheit der Elemente für die Durchführung des Tests wird als Testfeld verstanden, was aus der ingenieurwissenschaftlichen Perspektive der in 2.2.6 beschriebenen Herleitung eines Testfeldes und insbesondere gemäß der Definition der IALA nicht zutrifft. Die Richtlinien der IALA zur Gestaltung eines Testfeldes fordern explizit, dass „ein Testfeld [...] weder beschränkt, noch eingegrenzt durch aktuelle Architekturen, Datenstrukturen oder Prozesse sein“ soll [Iala17, S.4]. Dies erfüllen gegenwärtige maritime Lösungen nicht und unterstützen somit unzureichend die in Kapitel 1.3 beschriebene Reduktion von Systementwicklungs- und Testaufwänden durch ein erweiterbar und wiederverwendbar gestaltetes Testfeld.

Diese Arbeit fasst den Begriff „Testfeld“ wesentlich weiter und orientiert sich an der Definition gemäß dem Systems Engineering: „Ein Testfeld ist eine generische, nachhaltige Erprobungsumgebung, die alle Phasen der Systementwicklung unterstützt“ [Wass05, S.665]. In

diesem Sinne existieren derzeit im maritimen Bereich keine offenen und wiederverwendbaren Testfelder, auch nicht in den von der IALA genannten Bereichen. Dieser ingenieurwissenschaftlich anerkannten Auffassung eines generischen Testfeldes folgen andere Domänen und insbesondere die Transportsystementwicklung durchaus, wie beispielsweise der Kraft- und Luftfahrtbereich. Neben weltweiten Bemühungen existieren gerade in Deutschland intensive Testfeldaktivitäten. Auf der Autobahn A9 wurde ein physikalisches Testfeld für autonomes Fahren eingerichtet indem Landmarkenschilder, moderne Sensorik und eine digitale Karte als generisch nutzbare Test-Infrastruktur errichtet wurden [Bmvi16]. Weitergehend zeigen die untersuchten physikalischen Testfelder der Kraftfahrt- und Luftfahrt-domäne, dass unabhängig von im Vorfeld spezifizierten Testszenarien und Anwendungen durch eine dedizierte Testumgebung mit integriert virtuellen und physikalischen Komponenten ein Baukastenprinzip möglich ist. Diverse Werkzeuge der V+V bedienen sich der Testfeld-Infrastruktur und ermöglichen die konfigurierbare Umsetzung von verschiedenen Testszenarien. Dies erfüllt die Nachhaltigkeitsanforderungen eines physikalischen Testfeldes durch lose gekoppelte, mobile und stationäre Testfeld-Komponenten zur Nutzung entlang des Systementwicklungsprozesses.

Zusammenfassend erfolgt die Identifikation der folgenden Ansatzpunkte als Handlungsbedarf, um die in dieser Arbeit verfolgte Zielsetzung zu erreichen.

- Die Erstellung eines offenen und anpassungsfähigen, generischen Testfeldes, das für eine nachhaltige Verwendung in verschiedenen gegenwärtigen und zukünftigen maritimen Testszenarien nutzbar ist. In diesem Zusammenhang die Gestaltung der Testfeld-Architektur hinsichtlich einer hohen Konfigurierbarkeit der Komponenten des Testfeldes gemäß unspezifizierter Testszenarien.
- Die strikte Trennung des zu testenden Systems und des physikalischen Testfeldes durch Polymorphie als ein wesentliches Gestaltungsziel des zu entwickelnden Lösungsansatzes, insbesondere unter den Aspekten der Aufhebung von technologischen Abhängigkeiten.
- Die Orientierung an den etablierten Systementwicklungsmethoden durch die Integration virtueller und physikalischer Testfeld-Komponenten, um eine nahtlose Unterstützung entlang der X-in-the-Loop-Verfahren und eine steuerbare physikalische Testfeld-Infrastruktur zu ermöglichen.

## 4 Systemarchitektur eines physikalischen Testfeldes

Dieses Kapitel beschreibt auf der Basis des in Kapitel 2 und 3 dargestellten aktuellen Standes der Technik und Wissenschaft die Systemarchitektur eines physikalischen Testfeldes zwecks Unterstützung der Entwicklung von automatisierten Schiffsführungssystemen. Der Lösungsansatz für die in Kapitel 2.3 zu erfüllenden Anforderungen und das in diesem Kapitel daraus abgeleitete technologische sowie methodische Konzept wird mit sukzessive steigendem Detaillierungsgrad beschrieben.

Eine Systemarchitektur befasst sich auf entwerfender Ebene mit der Organisation und Gesamtstruktur eines Systems. Sie stellt die entwerfende Verbindung zu den Anforderungen durch die Identifikation der Hauptstrukturkomponenten eines Systems und deren Beziehungen dar. Dieser Entwurf ist von unterschiedlicher Detailtreue und Perspektive auf das zu gestaltende System [Somm12, S.184ff.]. Dies bedeutet, dass eine Systemarchitektur die integrierten Komponenten, Elemente und deren interne sowie externe Schnittstellen darstellt, durch welche die Fähigkeit zur Aufgabenerfüllung eines Systems gewährleistet wird. Dies erfolgt auf unterschiedlich detaillierter Ebene und aus verschiedenen Perspektiven auf das System [Wass05, S.67]. Eine Systemarchitektur beschreibt die folgenden Aspekte.

- Struktur der Elemente des Systems
- Schnittstellen und Kommunikation der Elemente des Systems
- Anwendungsszenarien
- Umgebung und Wechselwirkung der Elemente des Systems [BlFa13, S.26]

Um die Erklärungen des Lösungsansatzes zu verdeutlichen dient zunächst in Kapitel 4.1 der Anwendungsfall eines Systems für die Führung eines hochautomatisierten Schiffes als zu testendes System. Zur technologischen Gestaltung der Testfeld-Architektur findet anschließend in Kapitel 4.2 die Systematisierung der Elemente und Beziehungen eines Testfeldes für automatisierte Schiffsführungssysteme auf der Basis der Systementwicklungsmethodik und der Systemkonzepte des Zielsystems statt. Auf der Grundlage dieser Entwurfsziele wird in den folgenden Kapiteln deduktiv ein Lösungsvorschlag für eine Testfeld-Architektur hergeleitet und modelliert.

### 4.1 Anwendungsfall: Führung eines hochautomatisierten Schiffes

Zugunsten eines besseren Verständnisses des Lösungsansatzes dieser Arbeit stellt das folgende Kapitel einleitend den Anwendungsfall eines zu testenden Systems vor, welcher möglichst viele Aspekte der Konzepte des Zielsystems abdeckt. Ein Anwendungsfall beschreibt im Rahmen der Systementwicklung die möglichen Szenarien die dazu dienen ein Ziel zu erreichen [VWFG13, S.158]. Herangezogen wird ein Führungssystem für Schiffe, das

in hochautomatisierter Weise gemäß der Kategorisierung in Kapitel 2.1 die Navigation, Verkehrsführung und Kontrolle realisiert. Dieser Anwendungsfall leitet sich aus den in Kapitel 2.1 beschriebenen Systemkonzepten ab, welche diese Arbeit durch das physikalische Testfeld adressiert. Wengleich der Anwendungsfall als repräsentatives Beispiel genutzt wird, erhebt das physikalische Testfeld den Anspruch ebenfalls außerhalb dieser Systemgrenzen existierenden Komponenten eine Testumgebung bieten zu können. Beispiele für solche ebenfalls zu testenden Systeme sind moderne Kompasssysteme und automatisierte Multi-Sensor-Systeme.

Konsequent erfolgt im weiteren Verlauf der Arbeit die Bezugnahme zu den in diesem Kapitel vorgestellten funktionalen Komponenten. Hierbei beinhaltet das zu testende System die Hardware und Software welche zur Erfüllung einzelner oder mehrerer Aspekte der automatisierten Schiffsführungsaufgabe notwendig sind. Die folgenden Funktionskomponenten sind küstenseitig, schiffseitig oder küstenseitig sowie schiffseitig realisiert und durch eine oder mehrere Komponenten lokalisiert. Die folgenden Abschnitte beschreiben die Funktionskomponenten im Hinblick auf die Konstruktion der operativen Testumgebung des physikalischen Testfeldes.

**Routenplanung.** Die Routenplanung stellt im Kontext von automatisierten Schiffsführungssystemen ein wichtiges georäumliches System zur Unterstützung der Entscheidungsfindung dar [NiKi09]. Hierfür dient die Routenplanung der Generierung einer oder mehrerer Routen eines Schiffes, deren Ausprägung durch eine Sequenz von Wegpunkten erfolgt. Die kartesischen Koordinaten  $(x_k, y_k)$  mit  $k = 1, \dots, n$  beschreiben die Wegpunkte, welche durch das physikalische Testfeld dem zu testenden System bereitgestellt oder entgegengenommen werden. Die Routenplanung nutzt zusätzlich zu den Wegpunkten weitere Merkmale wie Geschwindigkeit, Steuerwinkel, statische Hindernisse eines Geoinformationssystems wie Land, Inseln, Flachbereiche, Riffe und durch die Wassertiefe begrenzte Gebiete oder weitere eingeschränkte oder reglementierte Gebiete [Tsou10]. Ein physikalisches Testfeld stellt diese Informationen für das dynamische Testen über einen Kommunikationskanal und entsprechender maritimer Infrastruktur bereit, beispielsweise durch eine Testfeld-Komponente in Form eines ECDIS.

**Verkehrsregelung.** Die Funktionalität der Verkehrsregelung (engl. „Traffic Management“) repräsentiert mehrere Komponenten, die auf taktischer Ebene die Verfolgung des Weges gemäß der geplanten Route und des geplanten Pfades überwachen. Aktuelle Regularien wie das Kapitel V der SOLAS geben die Ausrüstung vor, mit welcher der umgebende Verkehr eines Schiffes detektiert und überwacht wird. Diese Ausrüstung umfasst unter anderem ECDIS, Radargeräte und das AIS [Govu00]. Das physikalische Testfeld stellt einem zu testenden System diese Komponenten der maritimen Infrastruktur küstenseitig und seeseitig bereit, um die operative Umgebung zu repräsentieren. Insbesondere die maritimen Verkehrszentralen (VTS) nehmen durch die Wahrnehmung des Umweltlagebildes und die resultierende Steuerung und Anweisungen des maritimen Verkehrs einen Einfluss auf die Verkehrs-

regelung des Schiffes. Somit muss ein hochautomatisiertes Schiff in der Lage sein, elektronische Befehle einer küstenseitigen Station entgegenzunehmen und in der Wegeplanung zu berücksichtigen.

**Wegeplanung.** Die Wegeplanung (engl. „Path Planning“) ähnelt der Routenplanung, unterscheidet sich jedoch bezüglich des Betrachtungshorizonts der Planungsalgorithmen. Während die Routenplanung die vollständige Reise adressiert, betrachtet die Wegeplanung einige zukünftige Minuten beziehungsweise hunderte Meter. Die Wegeplanung berechnet gegebenenfalls einen alternativen Weg zu der ursprünglichen Route, um ein Kollisionsrisiko zu vermindern oder zu verhüten. Resultierend unterscheiden sich die durch das physikalische Testfeld bereitgestellten Sensoren, Daten und Kommunikationsinfrastruktur. Die Wegeplanung fordert von der operativen Testumgebung die Bereitstellung von Informationen über erkannte Objekte, wie Schiffe oder andere Hindernisse in der Umgebung. Das AIS und Radarsystem des physikalischen Testfeldes stellen der Wegeplanung die Informationen dieser Objekte bereit, wie die Position und Geschwindigkeit. Das physikalische Testfeld benötigt als Testträger entsprechende Schiffe, die in der Lage sind der Wegeplanung die notwendigen Daten der zugehörigen Schiffsdynamik bereitzustellen, beispielsweise bezüglich der Manövrierfähigkeiten im Hinblick auf den Navigationsraum [NiKi09], [Gupt13].

**Führungsregeln.** Obwohl die Routenplanung und Wegeplanung die zu folgende Sequenz aus Wegepunkten und der Geschwindigkeit forcieren ist die direkte Übergabe an die Bewegungsregelung nicht möglich. Für die Anpassung dieser Sollwerte an die Anforderungen der Bewegungsregelung benötigt das automatisierte Schiffsführungssystem die Funktionalität des Umgangs mit den Führungsregeln in Form eines Führungsalgorithmus. Im Allgemeinen ist das Element der Führungsregeln dafür verantwortlich kontinuierlich die notwendigen Referenzwerte für die Position, Geschwindigkeit und Beschleunigung aus den Wegpunkten zu berechnen. Diese berechneten Werte werden der Bewegungsregelung bereitgestellt. Hierfür stellt das physikalische Testfeld die erfassten Daten aus Bewegungssensoren bereit, wie GNSS, IMU und externe Daten wie Wetterdaten beispielsweise in Form der Windgeschwindigkeit und Windrichtung, Wellenhöhe und Wellenneigung, aktuellen Geschwindigkeit und Richtung [BrFo05].

**Bewegungsregelung.** Die Bewegungsregelung berechnet die notwendigen mechanischen Kräfte und Momente, die vom Schiff für das Erreichen der von den Führungsregeln bereitgestellten Regelungsziele notwendig sind. Es existieren eine Vielzahl von Regelungszielen, wie Sollwert-Regulierung für dynamische Positionierung oder das Folgen eines Weges (engl. „Path Following“). Für die Bewegungsregelung stellt ein physikalisches Testfeld ähnliche Daten bereit wie für die Führungsregeln [AbFH16]. Weitergehend benötigt die Bewegungsregelung ein Dynamikmodell des Testträgers sowie die Aktorik zur Umsetzung der mechanischen Kräfte und Bewegungskontrolle des Schiffes.

## 4.2 Systematisierung des Testfeldes für automatisierte Schiffsführungssysteme

Das folgende Kapitel konkretisiert zunächst die in Kapitel 2.2.6 vorgestellte generische Testsystem-Architektur als Rahmen der V+V eines automatisierten CPS durch die Konzepte des Zielsystems. Die in Kapitel 2.1 vorgestellte Architektur zur Führung eines Schiffes und die damit einhergehend beschriebenen Konzepte und Technologien, die als Anwendungsfall in dem vorherigen Kapitel 4.1 illustriert wurden, sind in diesem Kontext relevant. Die grundlegendsten Bestandteile zur Erprobung eines automatisierten Schiffsführungssystems sind das Test- und das V+V-Management für die Testdurchführung sowie die Elemente des zu testenden Systems und des physikalischen Testfeldes. Wie Abbildung 21 illustriert erfolgt eine strikte Trennung zwischen dem zu testenden System und dem physikalischen Testfeld.

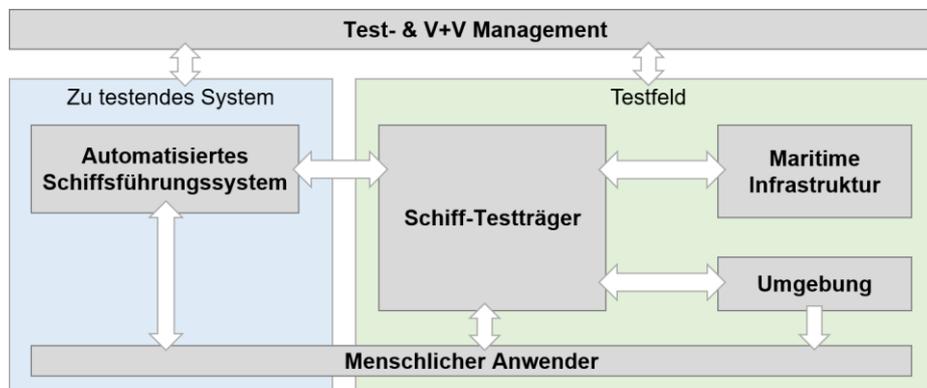


Abbildung 21: Genereller Überblick des Testfeldes für automatisierte Schiffsführungssysteme

Die in Abbildung 22 dargestellten Elemente des zu testenden Systems und Testfeldes repräsentieren verschiedene Komponenten zur Erfüllung der jeweiligen Aufgabe eines automatisierten Schiffsführungssystems gemäß den in Kapitel 2.1 beschriebenen Automatisierungsstufen eines Schiffes. Sofern sämtliche Elemente durch Technologie und nicht das menschliche Handeln erfüllt würden, ist die Automatisierungsstufe 5 und somit ein autonom geführtes Schiff erreicht. Das zu testende automatisierte Schiffsführungssystem übernimmt eine oder mehrere der Schiffsführungsfunktionen auf der linken Seite der Abbildung.

Dennoch ist es im Falle einer partiellen Funktionserfüllung durch das zu testende System notwendig, die entsprechenden Stimulationsdaten von der jeweils übergeordneten Komponente durch das Testfeld bereitzustellen, beispielsweise die Wegpunktsequenz einer gewählten Route der Routenplanung. Ebenso muss das Testfeld die Ausgabedaten der Reaktion des zu testenden Systems durch eine entsprechende Funktionskomponente entgegennehmen und ausführen. Dies ist beispielsweise die gewählte Trajektorie und Geschwindigkeit, welche durch eine Bewegungsregelung des Testfeldes ausgeführt wird. Daher ist von dem jeweiligen Testszenario abhängig, welche Komponenten des Schiffsführungssystems das Testfeld gemäß der Test-Konfiguration bereitstellt oder das zu testende System realisiert. Die Abbildung 22 leitet sich auf der Basis des Anwendungsszenarios des automatisierten Schiffsführungssystems und dessen operativer Umgebung sowie von der Referenzarchitektur für

Testfelder in Kapitel 2.2.6 ab. Diese Abbildung stellt die notwendigen Elemente eines Testfeldes für automatisierte Schiffsführungssysteme dar. Unter der Berücksichtigung der methodischen Ausführungen in Kapitel 2.2 existieren entsprechend dem jeweiligen Testszenario verschiedene Instanzen der entweder virtuell oder physikalisch ausgeprägten Elemente. Beispielsweise findet die Bereitstellung der Elemente im Falle einer MIL Erprobung durch Simulation statt, während in den folgenden Erprobungsphasen sukzessive der Austausch der simulierten durch physikalische Komponenten erfolgt [NHMG17, S.7ff.].

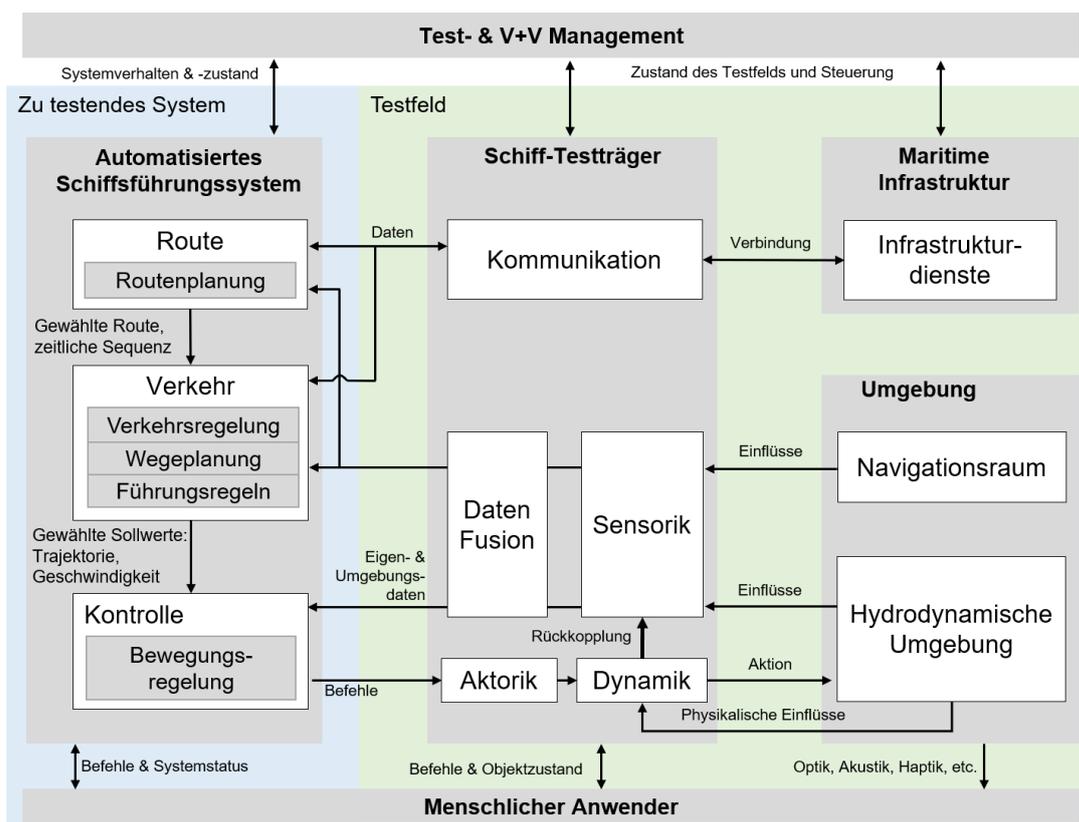


Abbildung 22: Elemente eines Testfeldes für automatisierte Schiffsführungssysteme

Grundsätzlich ist ein automatisiertes Schiffsführungssystem ein mit der Umgebung interagierendes CPS, zum Beispiel bei dem Manövrieren auf einer ebenfalls von anderen Verkehrsteilnehmern verwendeten Seeschiffahrtstraße. Hierfür nimmt das automatisierte Schiffsführungssystem die durch das physikalische Testfeld bereitgestellte Testumgebung mittels Sensorik oder die aktive Kommunikation mit den Systemen der maritimen Infrastruktur wahr (A1). Das automatisch zu führende Schiff besitzt eine Dynamik, die das automatisierte Schiffsführungssystem berücksichtigt (A2) und entsprechende Reaktionen zur mechanischen Kontrolle des Schiffskörpers durch die Aktorik des Testträgers bereitstellt (A3). Die Pfeile in Abbildung 22 zeigen die grundlegenden Interaktionen der Elemente des physikalischen Testfeldes. Für den Datenaustausch der Komponenten auf unterschiedlichen Funktionsebenen dienen drahtgebundene und drahtlose Kommunikationstechnologien (A4). Das physikalische Testfeld stellt die Testumgebung bereit, um mit der Hilfe von Werkzeugen des Test- und V+V-Managements das System zu erproben.

Auf der Basis der in diesem Kapitel vorgestellten Elemente eines Testfeldes für automatisierte Schiffsführungssysteme und dem in Kapitel 4.1 vorgestellten Anwendungsfall eines Systems zur hochautomatisierten Führung eines Schiffes werden im Folgenden die wesentlichen Kernelemente des physikalischen Testfeldes beschrieben, die es in der Testfeld-Architektur zu verorten gilt.

**Schiff-Testträger.** Die Erprobung eines automatisierten Schiffsführungssystems benötigt ein Schiff als Testträger. Der Testträger stellt zur Durchführung der Testszenarien die Sensoren, Aktoren und Komponenten bereit, um die Funktionalität des zu testenden Systems erproben zu können. Eine Besonderheit ist, dass der Testträger eine Dynamik je nach Schiffmodell besitzt, die veranschaulichend beim Vergleich eines Tankers und Sportboots unterschiedlich ausfällt. Somit ist die Auswahl des Testträgers und die entsprechenden Modifikationen für die Beobachtung und Steuerung durch das Testfeld für das automatisierte Schiffsführungssystem je nach Zielplattform zu berücksichtigen.

**Umgebung.** Die Umgebung beschreibt sämtliche in der Testdurchführung zu beobachtenden oder steuernden sowie für die dynamische Ausführung des zu testenden Systems notwendigen Aspekte, die sich in der operativen Umgebung des zu testenden Systems befinden. Dies sind Umweltbedingungen und Informationen des Umfelds, wie Strömung, Wassertiefe, Wind, Maschineninformationen oder Seezeichen. Die Umgebung umfasst Aspekte, die durch einen Sensor erfasst und nicht der aktiven Kommunikation zugeordnet werden.

**Maritime Infrastruktur.** Die aktiv kommunizierenden Elemente werden als maritime Infrastruktur bezeichnet. Dies umfasst aktiv mit dem zu testenden System kommunizierende Einrichtungen, wie beispielsweise landseitige VTS Systeme, andere automatisierte Schiffsführungssysteme und maritime Dienste zur Routenoptimierung. Diese Kommunikation erfolgt im Anwendungsfall des automatisierten Schiffsführungssystems entweder zwischen dem Schiff und der Küste (beispielsweise ein VTS) oder zwischen einem Schiff und einem anderen Schiff (beispielsweise der Informationsaustausch für die Kollisionsverhütung durch AIS).

**Test und V+V-Management.** Das Test und V+V-Management beschreibt die für die methodische Durchführung der jeweiligen Erprobung notwendigen Aspekte. Hierbei gilt es den Fokus dieser Arbeit zu beachten, der nicht auf den Methoden der V+V liegt. Die sind zwar als Anforderung eingeflossen, deren Gestaltung und Umsetzung erfolgt allerdings nicht im Rahmen dieser Arbeit. Relevant ist die spezielle Gestaltung des physikalischen Testfeldes für automatisierte Schiffsführungssysteme und dessen notwendige Ausrüstung, um die Nutzung der verschiedenen Methoden zu ermöglichen. Der für diese Arbeit relevante Aspekt ist die zusätzlich notwendige Ausrüstung für die Testüberwachung und Teststeuerung. In einigen Testfällen ist es nicht möglich, die relevanten Parameter direkt durch die notwendige Sensorik und Aktorik für die Ausführung eines zu testenden Systems zu beobachten und steuern. Daher muss die Gestaltung des physikalischen Testfeldes entsprechende Messkomponenten berücksichtigen, um das ganzheitliche Testszenario und insbesondere Kontextbedingungen beobachten zu können.

**Menschlicher Anwender.** Nach wie vor nimmt der menschliche Anwender bei automatisierten Schiffsführungssystemen einen hohen Stellenwert ein. Wie in Kapitel 2.1 beschrieben erfüllt ein Mensch entsprechend des jeweiligen Automatisierungsgrades unterschiedliche Aufgaben, wie die Beobachtung der Umgebung, strategische Reiseplanung oder gegebenenfalls den Eingriff in die Steuerung des automatisierten Schiffsführungssystems. Zwar werden sukzessive die menschlichen Funktionen des automatisierten Schiffsführungssystems durch entsprechende Komponenten ersetzt, dennoch berücksichtigt das physikalische Testfeld dieses Element je nach Testszenario.

Die in diesem Kapitel identifizierten Elemente eines physikalischen Testfeldes müssen im Rahmen der technologischen Gestaltung der Systemarchitektur des physikalischen Testfeldes berücksichtigt werden.

### 4.3 Testfeld-Architektur

Dieses Kapitel bietet einen Überblick über die Systemarchitektur des physikalischen Testfeldes. Kapitel 4.2 stellt die Elemente eines Testfeldes für automatisierte Schiffsführungssysteme vor, anhand derer die nötigen Komponenten, Schnittstellen und Funktionalitäten der operativen Umgebung ersichtlich werden, um einem zu testenden System die notwendige Testumgebung bereitzustellen. Das folgende Kapitel beschreibt auf dieser Basis die Ausgestaltung der Elemente in Form einer Systemarchitektur.

Für die Modellierung der Gestaltung der Systemarchitektur findet in dieser Arbeit durchgängig die Systems Modeling Language (SysML) der Object Management Group (OMG) in der Version 2 Verwendung. Die Diagrammart für die Modellierung der Systemarchitektur ist ein Komponentendiagramm. Das Komponentendiagramm ermöglicht die Darstellung der Komponenten und deren Schnittstellen, was insbesondere im Kontext der komplexen Systemlandschaft eines CPS angemessen ist. Dies zielt darauf ab, die technischen Komponenten des physikalischen Testfeldes deutlich voneinander abzugrenzen und die Schnittstellen darzustellen. Der komponentenorientierte Ansatz bietet insbesondere wegen der Erweiterbarkeit den Vorteil, dass die einzelnen Komponenten des Systems durch unterschiedliche technologische Implementierung realisiert und mit den weiteren Komponenten durch spezifizierte Schnittstellen interoperieren können.

Wie Abbildung 23 zeigt existiert ein zu testendes System (beziehungsweise eine zu testende Komponente), welches in die IT-Infrastruktur des physikalischen Testfeldes durch eine sich dynamisch anpassende und daher polymorphe Schnittstelle integriert wird. Die Kernfunktionalität des physikalischen Testfeldes ist die Übermittlung von Daten über die Kommunikationsinfrastruktur in der Form einer zentralen nachrichtenorientierten Middleware mit einheitlichem Datenmodell gemäß S-100 zwischen den verschiedenen Teilsystemen des physikalischen Testfeldes. Diese Teilsysteme sind Testfeld-Komponenten, zu testende Systeme und Komponenten des Test und V+V-Managements, welche die für den Test relevanten Funktionalitäten wie die Beobachtung, Auswertung und Dokumentation realisieren. Die Testfeld-Komponenten sind sowohl schiffseitige Einrichtungen wie ein AIS, Radarsystem

oder sonstige Sensorik und Aktorik oder landseitige maritime Infrastruktursysteme, wie ein VTS oder küstenseitiger Leitstand. Kapitel 4.3.5 stellt die seeseitige und landseitige Verteilung der Testfeld-Komponenten näher dar. Ein zu testendes System besteht gemäß der Beschreibung in Kapitel 2.1.3 aus Hardware, Software und einer oder mehreren Schnittstellen. Üblicherweise basieren sowohl die Schnittstellen für den Datenaustausch der Komponenten des physikalischen Testfeldes sowie zu testende Systeme auf existierenden Standards, die in Kapitel 2.1.3 vorgestellt wurden. Für die simulative Anreicherung der Testumgebung dient ein Simulations-Adapter mit dessen Hilfe Simulatoren in das physikalische Testfeld integriert werden. Das physikalische Testfeld lässt sich im Wesentlichen in die drei übergeordneten Strukturkomponenten der Kommunikationsinfrastruktur, Testfeld-Komponenten sowie Test und V+V-Management gliedern, die in den folgenden Kapiteln detailliert beschrieben werden.

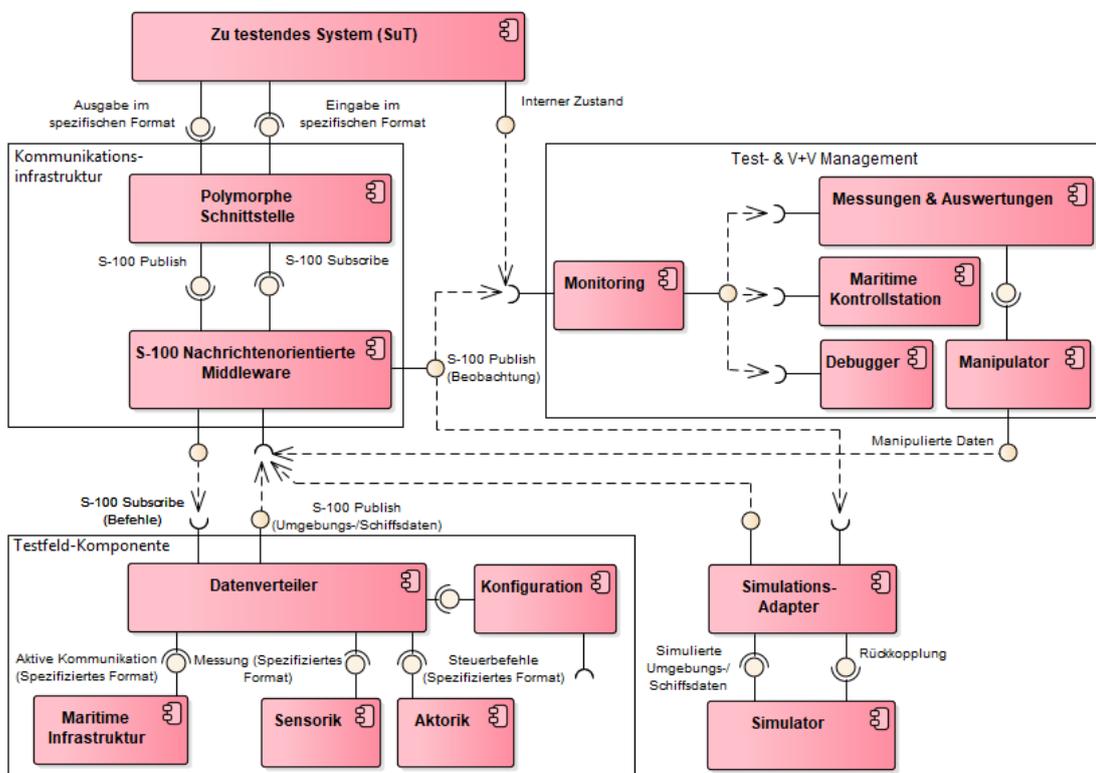


Abbildung 23: Ganzheitliche Systemarchitektur des physikalischen Testfeldes

### 4.3.1 Einheitliches Datenmodell gemäß S-100

Die Gestaltung einer wiederverwendbaren und erweiterbaren Testfeld-Architektur stellt eine Menge von Anforderungen an die Interoperabilität und Integrationsfähigkeit. Zum einen gilt es heterogene Komponenten in die IT-Infrastruktur des physikalischen Testfeldes zu integrieren, jedoch ebenfalls die zu testenden Systeme unterschiedlicher Implementierung (A19, A20). In diesen Fällen erschwert zudem die Datenvielfältigkeit die Gestaltung des Datenaustausches (A24). Weitergehend benötigt die Testfeld-Architektur für die nahtlose

Unterstützung des Systementwicklungsprozesses die Integration virtueller und physikalischer Testfeld-Komponenten (A15, A19).

Für die Realisierung eines interoperablen Datenaustausches zwischen heterogenen Komponenten findet die Übersetzung der lokalen inhärenten Datenmodelle in eine gemeinsame von allen Komponenten verständliche Form statt. Durch die Festlegung der Gestaltung auf das einheitliche Datenmodell und der Integration aller Systeme in dieses wird die Komplexität reduziert, da entgegen einer N: M Transformation eine N: 1: M Transformation durch transitive Abbildung stattfindet. Durch die Verwendung eines einheitlichen Datenmodelles besitzen die virtuellen und physikalischen Testfeld-Komponenten eine einheitliche Referenz als Bezugspunkt, der nach der Anpassung an diesen zu der Integrationsfähigkeit der virtuellen und physikalischen Komponenten unterschiedlicher Implementierung führt, wie in Abbildung 23 dargestellt [HoWo04, S.355].

Für diese Kopplung der verschiedenen Komponenten gilt es den unterschiedlichen Schnittstellen besondere Beachtung beizumessen. Um die Interoperabilität zwischen Komponenten und Systemen zu erlangen, müssen deren inhärente Schnittstellen in der Lage sein auf unterschiedlichen Ebenen der Interoperabilität miteinander zu operieren [WaTW09]. Der Begriff Schnittstelle (engl. „Interface“) beschreibt den Teil des Systems, welcher die Kommunikation zwischen zwei oder mehreren Systemen durch Übersetzungs- und Vermittlungsfunktionen erfüllt. Im Kontext des automatisierten Schiffsführungssystems als CPS existieren Schnittstellen in Form von Hardwareschnittstellen, Softwareschnittstellen, Hardware-Software-Schnittstellen und Mensch-Maschine-Schnittstellen [Halb94, S.139 ff.]. Um die Eingabe- und Ausgabesignale zu übersetzen ist das Wissen über die Form und Struktur der Schnittstellen der involvierten Systeme notwendig [GaEK01, S.33].

Der Lösungsansatz eines einheitlichen Datenmodells für einen interoperablen Datenaustausch etablierte sich im Rahmen e-Navigation der IMO, wie näher in Kapitel 2.1.3 erörtert. Die e-Navigationsarchitektur der IMO verlangt einen Datenaustausch auf der Basis der domänenweit etablierten einheitlichen maritimen Datenstruktur (engl. „Common Maritime Data Structure“) gemäß dem S-100 Framework. Weitergehend schlagen die Gestaltungsrichtlinien der IALA für maritime Testfelder die Verwendung des S-100 Frameworks für den Datenaustausch vor oder Mechanismen zur Konformität einzurichten [Iala17]. Für die Konformität mit aktuellen technischen und normativen Entwicklungen der maritimen Domäne folgt diese Arbeit diesem Lösungsansatz.

**Interoperabilitätsmechanismen von S-100.** Das S-100 Framework besteht aus zwölf Teilbereichen, von denen für den in dieser Arbeit entwickelten Ansatz die in Abbildung 24 dargestellten relevant sind. Die S-100 Registry ist ein Informationssystem, welches verschiedene standardisierte Register unter anderem als einheitliche Taxonomie verwaltet. Die Verwendung einer einheitlichen Taxonomie ist ein weit verbreitetes Lösungskonzept zum Erreichen von Interoperabilität und der Bewältigung der Herausforderungen von Synonymie und Polysemie, insbesondere auf der semantischen Ebene [BBBL14]. Als Register bezeichnet das S-100 Framework verwaltete Listen, von denen viele das Kernkonzept des Merkmalskonzept-Verzeichnisses (engl. „Feature Concept Dictionary“) sind [WAGP08, S.7f.]. Ein Feature

bezeichnet die Beschreibung eines abstrahierten räumlichen Objekts der realen Welt [BRRN05]. Es existieren verschiedene Register, wie das Darstellungsregister (engl. „Portrayal Register“), Metadatenregister und weitere. Durch die Verwendung eines Registers als Referenzquelle zur verwalteten Speicherung der Datenschemata wird eine zur Interoperabilität führende Vereinheitlichung geschaffen. Für die Spezifikation eines Systems erfolgt die Erstellung eines Merkmalskatalogs (engl. „Feature Catalogue“) auf der Basis verschiedener Feature Concept Dictionary. Ein Feature Catalogue beschreibt die Inhalte eines auf dem S-100 Framework basierenden Systems, in dem es die definierten Merkmale und Attribute verschiedener Feature Concept Dictionary für ein System verbindet. Das allgemeine Merkmalsmodell (engl. „General Feature Model“) beschreibt das konzeptionelle Modell zur Definition der Merkmale und Informationen sowie deren Attribute und Beziehungen. Auf der Basis eines Feature Catalogue findet die Modellierung des Anwendungsschemas als konkrete Ausprägung eines Systems statt [Iho17].

Das einheitliche Datenmodell der Testfeld-Architektur stellt dem folgend ein Anwendungsschema dar, welches gemäß den strukturellen Vorgaben des S-100 Datenframeworks modelliert ist, wie in Abbildung 24 dargestellt. Somit ergibt sich die Nutzung eines standardisierten Datenschemas zur Modellierung des Datenmodells des physikalischen Testfeldes und der Nutzung einer einheitlich definierten Terminologie. Ein physikalisches Testfeld verwendet gemäß der Testfeld-Architektur S-100 konforme Datenmodelle für die Beschreibung des Inhalts und der Struktur der Datenobjekte, welche durch die nachrichtenorientierte Middleware der Testfeld-Infrastruktur ausgetauscht werden.

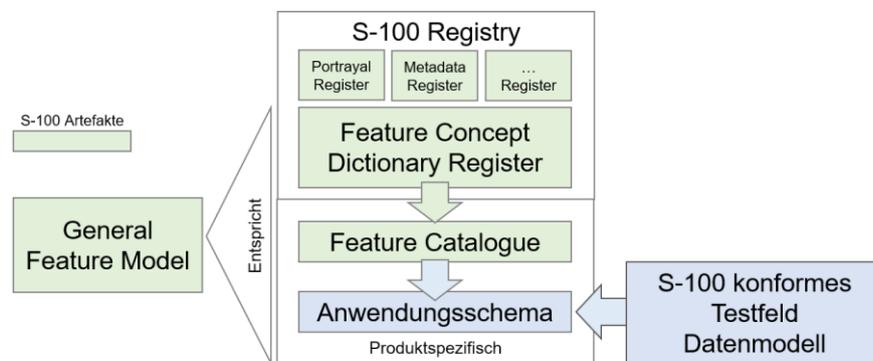


Abbildung 24: Interoperabilitätsmechanismen des S-100 Frameworks für den Datenaustausch und die Einordnung des Datenmodells des physikalischen Testfeldes

Gemäß den in Kapitel 2.1.4 beschriebenen Ebenen der Interoperabilität findet ein erfolgreicher Datenaustausch durch das syntaktische und semantische Verständnis des Dateninhalts statt, das in einem Feature Catalogue definiert ist. Weitergehend durch ein mit der Hilfe des Anwendungsschemas gemeinsames Verständnis der Struktur der Daten, das den Strukturvorgaben von S-100 entspricht. Ergänzend standardisiert das S-100 Framework eine interoperable Kommunikation durch die Vorgaben für die Kodierung gemäß ISO/IEC 8211 (engl. „Specification for a data descriptive file for information interchange“). Dieser bereits in dem S-57 Standard und weiteren geographischen Informationsübertragungen verwendete Standard spezifiziert das Austauschformat für stark variierende, komplexe

und große Daten. Hierfür findet die Verwendung einer Baumstruktur als Datenrepräsentation statt, die unter anderem Bezeichnungen und Relationen ausdrückt [Iho17].

Abbildung 25 stellt beispielhaft ein mit dem S-100 Framework konformes Datenobjekt dar. Hierin ist ein Schiff mit einer Rufnummer des mobilen Seefunkdienstes (engl. „Maritime Mobile Service Identity“, kurz MMSI) und einer Position unter der Verwendung des geodätischen Referenzsystems World Geodetic System 1984 (WGS84) in zwei Dimensionen modelliert. Ein S-100-Klassifizierer ist eine verallgemeinerte Klasse, die wiederum eine beliebige Menge von Features enthält [Iho17].

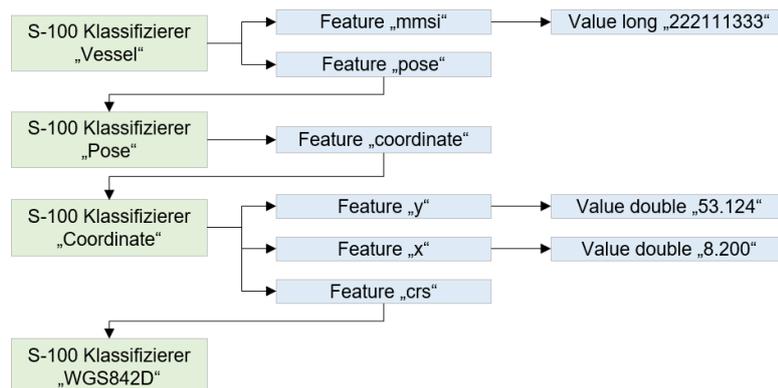


Abbildung 25: Schiff mit einer Position und einer MMSI modelliert als ein mit dem S-100 Framework konformes Datenobjekt

### 4.3.2 Publish-Subscribe nachrichtenorientierte Middleware

Für die Datenübertragung zwischen einem zu testenden System und den Testfeld-Komponenten sowie dem Test- und V+V-Management bildet die Kommunikationsinfrastruktur das Rückgrat des physikalischen Testfeldes. Da diese Arbeit den Themenbereich der CPS zuzuordnen ist existiert in den meisten Fällen eine geografisch verteilte Kommunikationsinfrastruktur, die aus einzelnen Systemknoten des physikalischen Testfeldes besteht. Diese verwendet für den Datenaustausch oft ein kabelloses Übertragungsverfahren, wie LTE, Satellit, WLAN oder UKW. Die folgenden Bestandteile realisieren die Funktionen für die Kommunikation der internen Testfeld-Infrastruktur. Wie in Kapitel 2.3.1 beschrieben forciert diese Arbeit nicht die Unterstützung des Testens unterschiedlicher Kommunikationstechnologien, sondern abstrahiert die Testfeld-Infrastruktur von den Technologien des zu testenden Systems.

Die Anforderungen an die Kommunikation des physikalischen Testfeldes gestalten sich anders als in einem virtuellen Testfeld. Entgegen der hohen technischen Anforderungen an die notwendige Konsistenz in der virtuellen Umgebung liefert ein physikalisches Testfeld automatisch ein konsistentes Umweltszenario. Die Herausforderung eines physikalischen Testfeldes liegt in der küstenseitigen und seeseitigen Kommunikationsinfrastruktur und der teilweise schwankenden Datenübertragungsrate sowie die durch die Datenverarbeitung resultierende Latenz. Weitergehend muss ein physikalisches Testfeld den Datenaustausch zur Erfüllung der Beobachtungs- und Steuerungsfunktionalitäten bereitstellen. Resultierend aus

den umfangreichen Komponenten und Systemen, die das physikalische Testfeld bilden, ist die Testfeld-Architektur unter dem Paradigma der losen Kopplung gestaltet. Die Umgebungsdaten werden aus Sensoren über eine entsprechende Kommunikationsinfrastruktur für ein zu testendes System und den Testwerkzeugen transportiert. Weitergehend sendet ein zu testendes System die Steuerbefehle an das automatisierte Schiffsführungssystem.

Eine Middleware ist eine „[...] generische Softwareplattform zur Überbrückung der Heterogenität unterschiedlicher Systeme und Netze, die gleichzeitig auch eine Reihe wichtiger Systemdienste wie etwa Sicherheitskonzepte, Transaktionsmechanismen und Verzeichnisdienste anbieten“ [ScSp12, S.8]. Wie in Abbildung 26 dargestellt reduziert die Middleware die Komplexität der verteilten und dynamischen Umgebung bestehend aus heterogenen Systemen und Komponenten über Adapter, welche die spezifischen Eigenschaften einer Implementierung der Schnittstellen abstrahieren [RoGS14]. Dies unterstützt insbesondere die gestalterischen Anforderungen der Interoperabilität (A19), losen Kopplung von dezentralen Testfeld-Komponenten (A21) und Konfigurierbarkeit der Datenströme (A25). Abbildung 26 stellt eine schematische Übersicht der Komponenten und Schnittstellen einer Middleware dar, während im Folgenden näher auf die spezifischen Eigenheiten unterschiedlicher Konzepte für eine Middleware und deren mögliche Auswirkungen auf das physikalische Testfeld eingegangen wird.

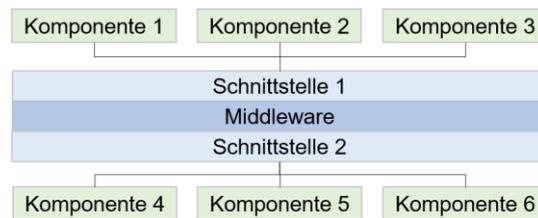


Abbildung 26: Elemente einer Middleware nach [EFGK03]

Eine Middleware ist ein etabliertes Konzept zum Austausch von heterogenen Daten (A24) und der Erlangung von Interoperabilität innerhalb eines CPS durch Abstraktion. Grundsätzlich unterteilt sich die Art des Datenaustauschs in synchrone und asynchrone Kommunikationsformen.

**Synchrone Kommunikation.** Die synchrone Kommunikationsform bezeichnet den zeitlich aufeinander abgestimmten Austausch von Daten, beispielsweise zwischen einem Client und Server. Hieraus folgt die zeitliche Abhängigkeit zwischen den Komponenten und die starke Bindung. Im Rahmen der synchronen Kommunikation kommen anwendungsorientierte beziehungsweise kommunikationsorientierte Middleware Systeme zum Einsatz, wie Remote Procedure Call (RPC), Common Object Request Broker Architecture (CORBA) oder Webdienste. Innerhalb eines solchen Systems existieren Anwendungsdienste und eine Laufzeitumgebung, welche Funktionen für verteilte Anwendungen ermöglichen und ein Bewusstsein bei den Sendern und Empfängern über den Kommunikationsstatus schaffen, beispielsweise hinsichtlich der erfolgreichen Zustellung der Daten [ScSp12, S.45]. Hierfür exportieren Komponenten die Funktionen an Clients, welche die für ihre Zwecke dienlichen Dienste auswählen. Aus diesem Grund bilden die Technologien der anwendungsorientierten

Middleware eine Form der synchronen und durch Methodenaufrufe und Funktionsaufrufe gesteuerten Middleware. Mithilfe dieser Technologien sind heterogene Systeme unterschiedlicher Anbieter integrierbar und ermöglichen somit komplexe Systeme, die jedoch Leistungseinbußen wegen der nötigen Metainformationen zur Steuerung der Kommunikation aufweisen [EFGK03].

**Asynchrone Kommunikation.** Die asynchronen Kommunikationsformen bieten Vorteile hinsichtlich der Parallelisierbarkeit und losen Kopplung der Kommunikationsteilnehmer (A21). Die asynchrone Kommunikation basiert auf dem Austausch von Nachrichten (engl. „Message Passing“) indem die Nachrichten verschickt und asynchron im Rahmen der datenstromorientierten Kommunikation abgerufen werden (A22). Hierdurch lassen sich große Datenmengen schnell und effizient von einem Sender zu mehreren Empfängern übertragen, ohne dass sich diese gegenseitig beeinflussen oder eine große Menge an Kommunikationssteuerungsdaten notwendig ist [ScSp12, S.78ff.]. Eine nachrichtenorientierte Middleware ist eine schnelle und anpassungsfähige Lösung für ein hochkonfigurierbares physikalisches Testfeld, das den Austausch von Nachrichten ohne eine Steuerung über die Anwendungsebene verfolgt [Curr05].

Die nachrichtenorientierte Middleware ermöglicht eine M: N Kopplung der verschiedenen Datenquellen und Datensinken, denn eine nachrichtenorientierte Middleware folgt einem Message Queuing oder Veröffentlichen und Abonnieren (engl. „Publish-Subscribe“) Kommunikationsmuster. Während sich eine Client-Server Kommunikationsform an der Verbindung mit einem N: 1 Ansatz orientiert, bietet eine nachrichtenorientierte Kommunikationsform mit dem Veröffentlichen-Abonnieren (engl. „Publish-Subscribe“) Kommunikationsmuster eine freie Konfiguration des Datenflusses und des Kontrollmechanismus (A25) [EFGK03]. In einer Message Queue sendet ein Produzent die Nachrichten an eine zeitlich entkoppelte und vom Empfänger abzurufende Warteschlange. Die Umsetzung als eine Publish-Subscribe-Kommunikationsform bezeichnet ein System, das als Basis einer Middleware für hoch dynamische und vollständig entkoppelten Teilkomponenten dient. Im Rahmen dieses Paradigma kommunizieren Komponenten nicht direkt miteinander, sondern interagieren über eine logische Schicht in Form eines Nachrichtenkanals, der als Mediator agiert [Curr05], [PuRP06, S.1ff.].

Als wesentliches Steuerungskonzept der Kommunikation dienen Ereignisse ausgelöst durch eingehende Nachrichten anstatt der Methodenaufrufe, weswegen das Publish-Subscribe-Kommunikationsmuster eine lose gekoppelte Middleware Variante bezeichnet. Eine konsumierende Komponente deklariert Ereignisse von Interesse, welche sie und sämtliche weiteren Interessenten beim Veröffentlichen durch die Quellkomponente erhält. Hierdurch resultiert der Vorteil eines dynamischen Hinzufügens weiterer Komponenten und das nicht nötige Wissen über Implementierungs- und Konfigurationsaspekte anderer Komponenten des Systems, wie in Abbildung 27 dargestellt [BaDG06], [ScSp12, S.78ff.].

**Publish-Subscribe-Kommunikationsmuster.** In dieser Arbeit fand auf der Basis der Tabelle 2 und der vorherigen Ausführungen die Gestaltungsentscheidung für die Verwendung

einer asynchronen Kommunikationsform in Form des Publish-Subscribe-Kommunikationsmusters statt. In der Tabelle 2 sind die synchronen Kommunikationstechnologien in einer Zeile zusammengefasst. Insbesondere basieren die in Kapitel 2.1.3 vorgestellten maritimen Übertragungsformate wie NMEA oder CAN auf Bussystemen und somit asynchronen Datenströmen (A22). Eine räumliche Entkopplung liegt vor, sofern ein Sender und Empfänger den gleichen Datenraum verwenden. Die Herausforderungen der datenstromgetriebenen und zunehmend komplexen Systeme des physikalischen Testfeldes adressiert eine synchrone Kommunikation der Middleware unter anderem wegen der entstehenden zusätzlichen Steuerungsdaten der Kommunikation nicht ausreichend [Curr05]. Zudem würde die zeitliche Abhängigkeit zwischen den Komponenten und der notwendigen Kommunikationssynchronisation entstehen. Für das physikalische Testfeld eignet sich daher insbesondere das Publish-Subscribe-Kommunikationsmuster für die nachrichtenorientierte Middleware. Die Kommunikation des physikalischen Testfeldes wird somit ereignisorientiert durch die datenerzeugenden Komponenten wie Sensoren und Aktoren gesteuert, ohne anwendungsspezifische Methodenaufrufe zu benötigen. Im Gegensatz zu dem Message Queuing, in dem ein Konsument der Daten nach wie vor eine Synchronisation mit der Warteschlange durchführt, existiert in der Publish-Subscribe-Kommunikationsform ein Kommunikationskanal, der keinerlei zeitliche oder räumliche Synchronisation aufweist und somit eine lose Kopplung beziehungsweise Entkopplung erreicht wird.

Konzept	Räumliche Entkopplung	Zeitliche Entkopplung	Synchronisierungs-entkopplung
Synchrone Kommunikation	Nein	Nein	Produzentenseitig
Message Queuing	Ja	Ja	Produzentenseitig
Publish-Subscribe	Ja	Ja	Ja

Tabelle 2: Technologieübersicht der Kommunikationsmuster einer Middleware nach [EFGK03]

Eine nachrichtenorientierte Middleware mit einem Publish-Subscribe-Kommunikationsmuster stellt eine hinreichende Entkopplung bezüglich der Aspekte der Zeit, Raum und Synchronisation der Komponenten dar, die geographisch verteilt in den maritimen Szenarien der Schiffsautomatisierung die Nachrichten austauschen (A21).

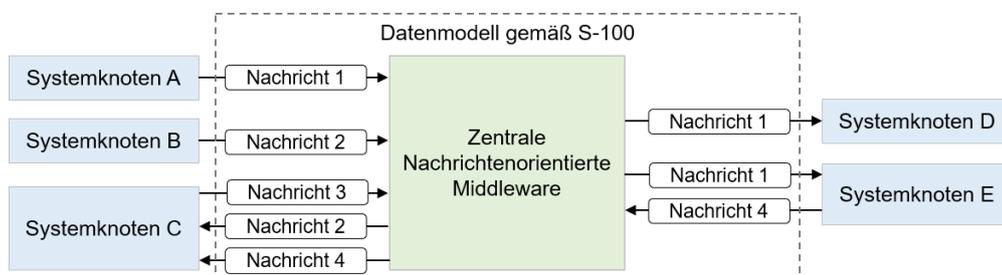


Abbildung 27: Datenverteilung der Systemknoten des physikalischen Testfeldes über zentrale nachrichtenorientierte Middleware nach [ScSp12, S.80]

Die Konfigurationsfähigkeit der Middleware ermöglicht eine Unterteilung der Kommunikationkanäle in Subnetzwerke, beispielsweise für die logische Trennung bezüglich

des Datenraumes und der Zeit während paralleler Experimente (A28). Durch diesen Ansatz ist die zentrale Einrichtung der nachrichtenorientierten Middleware und die dezentrale Konfiguration des Datenstromes durch das Publish-Subscribe-Kommunikationsmuster möglich, wie in Abbildung 27 dargestellt. Die Verwendung einer zentralen nachrichtenorientierten Middleware ermöglicht insbesondere die Beobachtungsfähigkeit durch entsprechendes Abonnieren der zentralen Middleware durch die der Testausführung dienenden Komponente, wie das Test- und V+V-Management.

Der semantische und strukturelle Inhalt der Datenobjekte ist flexibel, da die nachrichtenorientierte Middleware den Inhalt nicht interpretiert. Sämtliche Systemknoten übernehmen die Aufgabe, die zu übertragenden Datenobjekte gemäß dem einheitlichen Datenmodell im S-100 Format über die zentrale nachrichtenorientierte Middleware zu übertragen. Dies geschieht durch die Definition des einheitlichen Datenmodells gemäß S-100 als ein konzeptionelles Modell und entsprechender lokaler Nutzung in einer Implementierung durch den jeweiligen Systemknoten. Eine Änderung der Datenformate im Testfeld stellt somit keine Herausforderung an die Kommunikationsinfrastruktur dar [ScSp12, S.8]. Hierfür benötigen die die Testfeld-Infrastruktur nutzenden Komponenten die in Abbildung 28 dargestellten Funktionen.

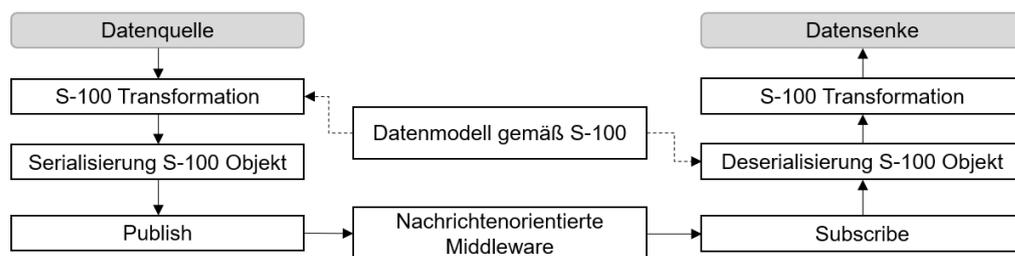


Abbildung 28: Veröffentlichen und Abonniert von Daten in dem physikalischen Testfeld

Zunächst findet die Transformation der generierten Daten einer Datenquelle in das einheitliche Datenmodell gemäß S-100 statt. Auf dieser Basis geschieht die Generierung von S-100 Datenobjekten und den entsprechenden Klassifizierern, die eine eindeutige Bezeichnung darstellen und auf eine Menge von Features verweisen. Diese zugeordneten Features, wie beispielsweise ein Sensordatum eines AIS, sind flexibel modellierbar, so dass im Falle der Erweiterung des Datenmodells diese hinzugefügt und entsprechend serialisiert/deserialisiert werden. Durch die in Kapitel 4.3.1 beschriebenen Interoperabilitäts-Mechanismen von S-100 ist somit ein flexibles und erweiterbares Übertragen der Datenstrukturen möglich.

In diesem Kontext erfolgt die Serialisierung eines S-100 konformen Datenobjekts, um es zu Veröffentlichen und über die nachrichtenorientierte Middleware zu übertragen. Die Serialisierung bezeichnet das Übersetzen von Datenstrukturen und Datenobjekten in ein sequentielles Format, um die Leistung der Datenübertragung in einem verteilten System zu erhöhen. Analog zu dieser Operatorenkette existiert für das Empfangen zunächst ein Abonnent, der eine oder mehrere gewünschte Informationen über die nachrichtenorientierte Middleware auswählt und diese anschließend erhält. Darauffolgend deserialisiert die der Serialisierungsfunktion entsprechende zugehörige Deserialisierungsfunktion die

übertragenen Datenobjekte und transformiert sie von dem S-100 Format in das jeweilige Zielformat der Datensenke.

### 4.3.3 Integration von Testfeld-Komponenten

Das Kernstück des physikalischen Testfeldes für automatisierte Schiffsführungssysteme stellen die Testfeld-Komponenten dar, wie beispielsweise ein Schiff als Testträger oder eine maritime Kontrollstation zur Überwachung der Testumgebung. Ein physikalisches Testfeld besitzt eine oder mehrere gekoppelte Testfeld-Komponenten, welche die Hardware, Software und Kommunikationsinfrastruktur für ein zu testendes System bereitstellen, um die Funktionalität in der operativen Umgebung auszuführen. Die verschiedenen Testfeld-Komponenten stellen die Umgebung zur Erprobung des zu testenden Systems entsprechend der Regelkreismethoden (z. B. HIL, SIL, oder Realwelt Erprobung) bereit, wie in Kapitel 2.2.5 dargestellt. Testfeld-Komponenten sind küstenseitig oder seeseitig installiert und über die Kommunikationstechnologie an die weitere Infrastruktur des Testfeldes angebunden. In diesem Zusammenhang unterstützen die Testfeld-Komponenten ein zu testendes System durch die Bereitstellung der eingehenden und ausgehenden Datenströme von Systemen der maritimen Infrastruktur, Sensorik und Aktorik für die mechanische Interaktion mit der Umwelt. Eine Testfeld-Komponente beinhaltet die im Kapitel 4.2 genannten Elemente des Schiff-Testträgers inklusive der notwendigen Komponenten für die Wechselwirkung mit der Umgebung und der maritimen Infrastruktur, entsprechend der Bedürfnisse eines zu testenden Systems an die Datenverfügbarkeit (A23).

Die Testfeld-Komponenten weisen eine domänenspezifische Ausprägung durch exklusive Sensoren, Aktoren und maritime Infrastrukturkomponenten auf, wodurch das physikalische Testfeld für automatisierte Schiffsführungssysteme von anderen Testfeldern unterschieden wird. Die Gestaltung der Testfeld-Architektur im Hinblick auf die dedizierten und lokal konfigurierbaren Testfeld-Komponenten unterstützt die Entkopplung und Flexibilität des Testfeldes (A21). Die Testfeld-Komponenten übernehmen eine Vorverarbeitung der Daten je nach der Anforderung des zu testenden Systems, wie Filteroperationen, Datenfusion oder eine Datenauswahl. Die Technologien der Testfeld-Komponenten sind entweder virtuell, integriert virtuell/physikalisch oder physikalisch ausgeprägt.

**Datenverteiler.** Die Testfeld-Komponenten besitzen verschiedene Sensoren und Funktionsbestandteile, wie ein Radarsystem, Echolot, eine IMU, Aktoren wie das Ruder, Querstrahlruder oder die Umdrehungen pro Minute (UpM) des Motors und aktiv kommunizierende maritime Infrastrukturkomponenten. Die proprietären Schnittstellen dieser Komponenten sind heterogen und basieren beispielsweise auf den Standards NMEA 0183, NMEA 2000 oder proprietären. Daher integriert die Komponente für die Datenverteilung diese heterogenen Technologien in die Testfeld-Infrastruktur als eine in das einheitliche S-100 Framework integrierende Standardisierungskomponente (A24). Hierbei existieren von der Integration eines zu testenden Systems unterschiedliche Anforderungen, wie Abbildung 29 illustriert. Die Integration des zu testenden Systems zeichnet sich durch eine dynamische Transformation

zur Laufzeit aus, die je nach Implementierung des zu testenden Systems stetig variiert und daher eine spezielle Gestaltung benötigt. Die Funktionalität der dynamischen Transformation zur Integration der zu testenden Systeme findet durch die in Kapitel 4.5 beschriebene polymorphe Schnittstelle statt. Die Implementierung einer Testfeld-Komponente ist nicht stetig variierend und wird für unterschiedliche Tests wiederverwendet, daher ist eine statisch implementierte Funktionskomponente für die S-100 Transformation ausreichend.

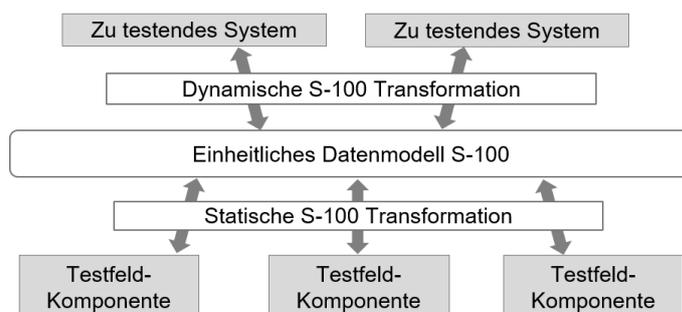


Abbildung 29: Dynamische und statische Integration der Komponenten in das einheitliche Datenmodell gemäß S-100

Die in der Testumgebung der Testfeld-Infrastruktur erfassten und vom zu testenden System an die Testfeld-Infrastruktur gesendeten Daten werden durch einen Adapter in das einheitliche Datenmodell gemäß S-100 der Testfeld-Infrastruktur innerhalb des Datenverteilers umgewandelt. Ein Adapter ist ein Entwurfsmuster der Softwareentwicklung und dient als Mediator zwischen Schnittstellen. In der Rolle des Mediators übernimmt ein Adapter die notwendigen Übersetzungstätigkeiten, um das Kommunikationsprotokoll sowie syntaktische und semantische Datenmodelle und resultierend die Interoperabilität zwischen inkompatiblen Schnittstellen herzustellen [GaRi03, S.171 f.]. In der Testfeld-Architektur für hochautomatisierte Schiffsführungssysteme dient der Adapter als statischer Mediator zwischen den Schnittstellen der Komponenten und Systeme und dem einheitlichen Datenmodell des physikalischen Testfeldes. Der Datenverteiler einer Testfeld-Komponente stellt die Daten an Bord eines Schiffes sowie die Daten für maritime Überwachungssysteme über kabelgebundene und kabellose Datenübertragung über die nachrichtenorientierte Middleware der Kommunikationsinfrastruktur zur Verfügung. Hierfür stellt eine Testfeld-Komponente über den Datenverteiler eine beliebige Kommunikationsschnittstelle bereit, wie LAN, WLAN oder LTE. Nach der erfolgten Transformation in das S-100 Format erfolgt das Senden über einen entsprechenden Publisher, der die Daten an die zentrale nachrichtenorientierte Middleware übermittelt.

Unterschiedliche Testszenarien benötigen verschiedene Konfigurationen der Testfeld-Komponenten gemäß der durch den Benutzer spezifizierten Testparameter (A25). In einem physikalischen Testfeld für automatisierte Schiffsführungssysteme finden verschiedenste Sensoren und Aktoren Verwendung, um beispielsweise ein Lagebild über die Verkehrs- und Umweltsituation zu erzeugen. Hierbei besitzen Parameter zur Anreicherung eventuell fehlender Daten eine Relevanz, wie die Auflösung, Frequenz oder Datenfusion. Da eine dezentrale IT-Infrastruktur vorliegt erfolgt die Integration der verschiedenen Test-

feld-Komponenten mit der Hilfe einer losen Kopplung. Bestimmt durch das jeweilige Test-szenario findet die Konfiguration der einzelnen Testfeld-Komponenten statt, um sie individuell zu konfigurieren. Daher verwendet die Komponente für die Datenverteilung eine Schnittstelle, durch die sie beispielsweise über eine lokale Konfigurationsdatei oder über ein Netzwerk konfiguriert wird. Da in einer realen Umgebung die Teilsysteme dezentralisiert installiert sind erfolgt die Konfiguration neben einer zentralen Möglichkeit lokal innerhalb einer jeden Testfeld-Komponente. Diese Gestaltungsentscheidung stellt sich insbesondere sinnvoll dar, denn die Komponenten sind teilweise an Bord eines Schiffes verortet (A26).

### 4.3.4 Monitoring

Wie in Kapitel 4.2 beschrieben ermöglichen Werkzeugen des Test- und V+V-Management die Testdurchführung während der Erprobung gemäß den normativen Standards und der Methoden der V+V. Die Werkzeuge des Test- und V+V-Management umfassen Aspekte, die unabhängig von dem jeweiligen Testszenario gültig und für verschiedene zu testende Systeme wiederverwendbar sind.

Ein direkter Zugriff auf die nachrichtenorientierte Middleware erlaubt den abonnierenden Systemen die Beobachtung (engl. „Monitoring“) des Zustandes des physikalischen Testfeldes für die Datensammlung, Überwachung und Bewertung (A8, A9, A11). Die Daten der verschiedenen Beobachtungsparameter der Umwelt, der Eingabe- und Ausgabedaten des zu testenden Systems und der Testfeld-Infrastruktur werden über die nachrichtenorientierte Middleware dem Monitoring zur Verfügung gestellt. Durch die Abbildung sämtlicher Daten auf das einheitliche Datenmodell der Testfeld-Infrastruktur als Referenzpunkt ist es dem Monitoring möglich, die Datenströme des ganzheitlichen Testszenarios in interoperabler Weise zu beobachten. Hierdurch erfasst das Monitoring die relevanten Datenströme der Testfeld-Infrastruktur ohne direkt einzugreifen, den Datenstrom zu modifizieren und das zu testende System zu beeinflussen.

Das Monitoring besitzt sowohl eine Schnittstelle zu der nachrichtenorientierten Middleware des physikalischen Testfeldes sowie optional zur Beobachtung des internen Zustands zu dem zu testenden System (A10). Diese optionale Schnittstelle ist für einige Methoden der V+V wie die Laufzeitverifikation notwendig, die ein White-Box oder Grey-Box Verfahren realisieren. Da hierfür die internen Implementierungen eines zu testenden Systems beachtet werden müssen ist die Entwicklung einer individuellen Beobachtungsschnittstelle notwendig.

Das Monitoring bietet verschiedenen Anwendungen die Möglichkeit über eine Schnittstelle die Beobachtung und Auswertung des jeweiligen Tests durchzuführen. Dies sind die visuelle Überwachung der Testdurchführung durch eine maritime Kontrollstation sowie eine Datenstromanalyse für die Fehleranalyse durch einen Debugger. Ein Debugger ermöglicht die direkte Einsicht in die übertragenen S-100 Datenobjekte der nachrichtenorientierten Middleware zum Auffinden von Fehlern im System und ist in der Lage, die Datenströme zu steuern [Zell09, S.9ff.]. Die Evaluation der gemessenen Daten erfolgt durch eine Komponente

für die Messung und Auswertung, entsprechend der normativen Vorgaben der V+V im Rahmen des Systems Engineering und Auswahl der auszuwertenden Testparameter (A12). Im Sinne der Nachvollziehbarkeit und Reproduzierbarkeit erfolgt die Dokumentation der Testergebnisse, durchgeführten Testszenarien, der Konfiguration des physikalischen Testfeldes und der Testparameter sowie die gemessenen Eingabe- und Ausgabedaten der Testumgebung und des zu testenden Systems (A13, A14).

Das Monitoring des physikalischen Testfeldes ist unter den Aspekten der notwendigen Generizität unabhängig von dem jeweiligen Testfall gestaltet. Sämtliche Abonnenten und Veröffentlichender verwenden eine konzeptionelle Repräsentation des einheitlichen Datenmodells gemäß S-100, um die zu übertragenden Daten entsprechend zu modellieren. Hierdurch entsteht insbesondere für das Test- und V+V-Management die Fähigkeit des semantischen Verständnisses sämtlicher zu übertragender Daten. Die vom Test- und V+V-Management erforderlichen Funktionen erfolgen auf dieser Basis, wie Speichern, Filtern und Auswerten von Daten. Die konkreten Datenstrukturen des Tests beeinflussen nicht die Realisierung der Testfeld-Infrastruktur sondern werden durch entsprechende Operatoren realisiert, wie näher in Kapitel 4.3.3 beschrieben.

### 4.3.5 Physikalische Verteilung

Das physikalische Testfeld für automatisierte Schiffsführungssysteme übermittelt Daten gemäß dem S-100 Framework über eine nachrichtenorientierte Middleware durch ein Publish-Subscribe-Kommunikationsmuster, wie in den vorherigen Kapiteln beschrieben. In diesem Zusammenhang existieren schiffseitige und landseitige Komponenten, die als geographisch verteilte Systemknoten im Testgebiet untereinander Sensordaten und Daten der Infrastruktur lose gekoppelt austauschen. Das folgende Kapitel beschreibt die physikalische Verteilung der Komponenten des Testfeldes und deren Verbindungen in dem Studiengebiet untereinander, da die Datenverfügbarkeit insbesondere im Kontext der CPS eine hohe Bedeutung aufweist (A23). Das in Abbildung 30 dargestellte Verteilungsdiagramm zeigt die physikalische Verteilung des physikalischen Testfeldes. Die Ableitungsbasis stellen die Elemente des Testfeldes für automatisierte Schiffsführungssysteme in Kapitel 4.2 und der Entwurf der Systemarchitektur in Kapitel 4.3.3 dar.

Sämtliche Testfeld-Komponenten als Systemknoten, das Test- und V+V-Management und die zu testenden Systeme werden über die Kommunikationsinfrastruktur des physikalischen Testfeldes verbunden. In der Abbildung 30 ist zu beachten, dass zwar eine Testfeld-Komponente jeweils seeseitig und landseitig dargestellt ist, jedoch mehrere parallel existieren. Eine Testfeld-Komponente als Systemknoten besitzt hierbei einen regional begrenzten lokalen Wahrnehmungsbereich. Um diesen Wahrnehmungsbereich zu erweitern decken mehrere geographisch verteilte Systemknoten ein Erfassungsgebiet des physikalischen Testfeldes ab. Ergänzend repräsentieren mehrere Testträger verschiedene Szenarien, in denen Systemknoten auf den Testträgern ein kooperierendes Testszenario ermöglichen. Der physikalische Grundaufbau bleibt jedoch identisch.

Alle mobilen und stationären Systemknoten und die Benutzer des physikalischen Testfeldes verbindet die Kommunikationsinfrastruktur. Die Daten zwischen dem Land und der See sowie untereinander auf der See werden über eine drahtlose Datenübertragung (beispielsweise WLAN gemäß IEEE 802.11, Mobilfunk (LTE) oder Richtfunk) zwischen mobilen und stationären Testfeld-Komponenten und der zentralen nachrichtenorientierten Middleware ausgetauscht. Durch die Vielzahl der gemessenen Sensordaten und der drahtlosen Übertragung dieser entsteht eine hohe benötigte Datenübertragungsrate. Da mehrere konsumierende Systeme auf dieselben Daten der Sensorik zugreifen können wurde in dem Lösungsansatz dieser Arbeit ein zentrales Vermittler-System als Verteilstation berücksichtigt, wie in Abbildung 30 dargestellt. Hierdurch sendet ein produzierendes System (Publisher) die Sensordaten lediglich einmal über eine drahtlose Übertragung an Land beziehungsweise auf See. Anschließend konsumieren ein oder mehrere Systeme (Subscriber) über eine drahtgebundene Netzwerkverbindung (LAN gemäß IEEE 802.3) die jeweiligen Daten. Die Zeitsynchronisation der verteilten Systeme erfolgt durch jede Testfeld-Komponente auf der Basis eines Zeitprotokolls.

Auf der Ebene der nachrichtenorientierten Middleware wird der Datenfluss zwischen Komponenten kontrolliert und gesteuert. Die nachrichtenorientierte Middleware verwendet eine Komponente zum Beobachten der Datenkanäle, adressieren der Daten und für die Zugriffsteuerung. Die Publish-Subscribe Kommunikationsform ermöglicht das Konsumieren sämtlicher der nachrichtenorientierten Middleware zur Verfügung stehenden Daten ohne die Kommunikation beziehungsweise Systeme durch Methodenaufrufe und eine notwendige Synchronisation untereinander zu beeinflussen. Hierdurch entsteht die Möglichkeit eines mitschneidenden Konsumierens der über die Kommunikationsinfrastruktur übermittelten Daten durch Werkzeuge der V+V, wie ein Debugger oder das Monitoring.

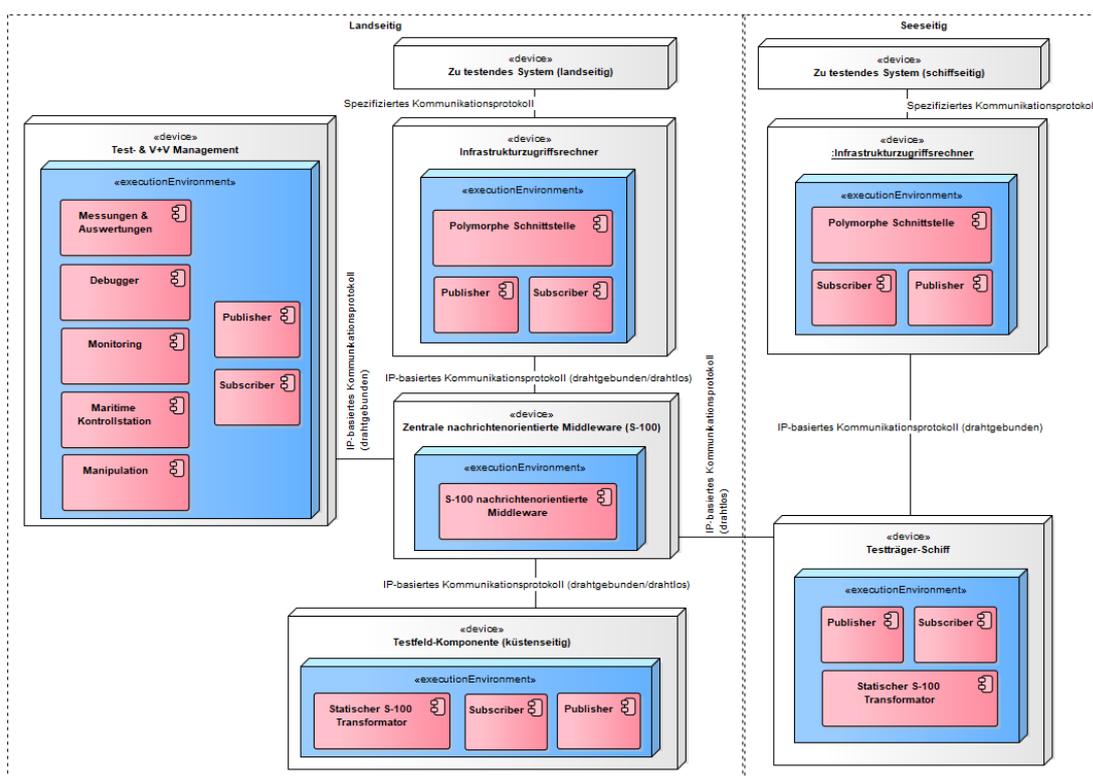


Abbildung 30: Physikalische Verteilung der Komponenten des physikalischen Testfeldes

Gemäß den funktionalen Ebenen des automatisierten Schiffsführungssystems (Route, Verkehr, Kontrolle) sind die zu testenden Systeme entweder landseitig oder schiffseitig verortet und nutzen einen Infrastrukturzugriffrechner für die Teilnahme an der Kommunikation mit dem physikalischen Testfeld. Insbesondere haben schiffseitige Komponenten zur direkten Ansteuerung der Aktorik einige Anforderungen an eine niedrigere Latenz der Datenübertragung und zuverlässige Kommunikationstechnologie, wie Komponenten der Verkehrsführung und Kontrolle (siehe Kapitel 2.1.3). Daher erfolgt im Falle des Testens von schiffseitigen Automatisierungssystemen eine direkte Verbindung der Testfeld-Komponenten des Schiffes mit dem Infrastrukturzugriffrechner ohne vorher die Daten für Zwecke der V+V über die drahtlose Kommunikationsinfrastruktur landseitig zu senden. Durch das Publish-Subscribe-Kommunikationsmuster erfolgt jedoch ein paralleles Senden der Steuerbefehle und Datenwahrnehmung an die zentrale nachrichtenorientierte Middleware zur weiteren Verarbeitung. Für strategische Komponenten der Navigation auf der Ebene der Route erfolgt ein direktes Senden der Befehle über die zentrale nachrichtenorientierte Middleware, wie in Abbildung 30 dargestellt.

## 4.4 Steuerung des physikalischen Testfeldes

Die Steuerung des physikalischen Testfeldes für Fehlerinjektion und Tests im Closed-Loop-Verfahren erfolgt auf der Basis von spezifizierten Testszenarien, deren Testparameter als Artefakte existieren, beispielsweise in Form eines Dokuments oder, im Falle von

Testautomatisierung, einer Software. Die Fähigkeit der in diesem Kapitel vorgestellten Steuerung des physikalischen Testfeldes ermöglicht vom Realszenario abweichende Datenströme durch Datenmanipulation oder Datenerzeugung zu erzeugen. Dies führt eine gewünschte Testsituation abweichend von der real vorherrschenden herbei oder bietet die Möglichkeit eine Fehlerinjektion für den Test der Robustheit des zu testenden Systems vorzunehmen. Im Kontext eines Tests im Closed-Loop-Verfahren erfolgen manuelle oder automatische Maßnahmen für die Modifikation der Testausführung durch eine Simulation. Hierbei erfolgt eine automatisierte Steuerung des Testszenarios durch die Rückkopplung der real vorherrschenden Situation, die unterstützend durch die Integration eines virtuellen Testfeldes und der enthaltenen Simulatoren möglich ist. In den folgenden Kapiteln wird sowohl die Möglichkeit der Fehlerinjektion durch die Manipulation der Datenströme sowie die Integration des virtuellen und physikalischen Testfeldes für Tests im Closed-Loop-Verfahren in dem physikalischen Testfeld beschrieben.

#### **4.4.1 Fehlerinjektion durch Manipulation**

Fehlerinjektionstests besitzen für das Testen von automatisierten Schiffsführungssystemen neben dem Testen der Erfüllung von Anforderungen und Systemeigenschaften eine wesentliche Rolle zur Überprüfung insbesondere von Randszenarien, wie in Kapitel 2.2 näher beschrieben. Durch das Injizieren von Fehlerzuständen in die steuerbare beziehungsweise manipulierbare Testfeld-Infrastruktur erfolgt die Durchführung von System-, Integrations- und Modultests von Hardware und Software. Unter Verwendung der Fehlerinjektion durch die Manipulation lässt sich mit der Hilfe des physikalischen Testfeldes das kontrollierte Testen durch das gezielte Injizieren von Fehlerzuständen beziehungsweise das Modifizieren der in das zu testende System übertragenen Teststimulation realisieren. Hierfür ist weitergehend die Beobachtung der Reaktion des zu testenden Systems notwendig [ULRC17]. Eine etablierte Methode für die Fehlerinjektion stellt das Fault, Activation, Readout, Measure Modell (FARM) dar, anhand dessen sich die Gestaltung einer entsprechenden Komponente des physikalischen Testfeldes orientiert. Dem FARM Modell folgend existiert eine Fehlerhypothese (Fault), die ebenso wie deren Aktivierungsmuster zum Auftreten des Fehlers (Activation) in einer entsprechenden Testspezifikation definiert sind. Mit der Hilfe des Manipulators der Testfeld-Infrastruktur erfolgt die Einspeisung des Fehlers durch entsprechende Modifikation der Datenströme und die anschließende Übermittlung an das zu testende System mit der Hilfe der Testfeld-Infrastruktur. Ferner erfolgt die Beobachtung und Messung des Verhaltens des zu testenden Systems (Readout), um die Messergebnisse und das Testergebnis bewerten und eine Aussage über die Zuverlässigkeit des zu testenden Systems erhalten zu können (Measure) [BePr04, S.180ff.].

Für die Manipulation der übertragenen Daten des Testfeldes ist der Manipulator in der Testfeld-Architektur wie in Abbildung 31 dargestellt verortet. Der Manipulator besitzt eine Benutzer- und/oder Softwareschnittstelle, um die jeweiligen Fehlerparameter beziehungsweise gewünschten Umweltzustände zu definieren und die Datenströme somit zu steuern.

In diesem Zusammenhang ist der Manipulator für die Fehlerinjektion zwischen der Datenquelle und Datensenke gesetzt und fängt die original übertragenen Daten ab. Die konkrete Umsetzung einer spezifischen Fehlerinjektions-Methode ist nicht Teil dieser Arbeit, jedoch die Fähigkeit des physikalischen Testfeldes zur Unterstützung dieser.

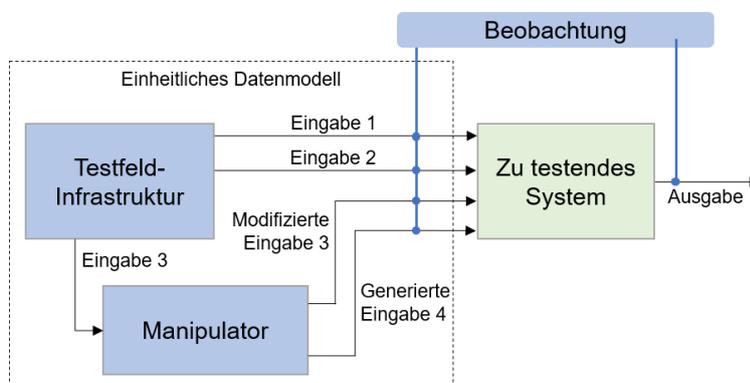


Abbildung 31: Schematische Darstellung der Fehlerinjektion und Steuerung des physikalischen Testfeldes durch eine die originalen Dateneingaben manipulierende Komponente

Als Lösungsansatz verwendet der Manipulator das einheitliche Datenmodell gemäß S-100 der Kommunikationsinfrastruktur, um die zwischen den Testfeld-Komponenten und dem zu testenden System übertragenen Daten zu beeinflussen. Sämtliche durch das einheitliche Datenmodell gemäß S-100 modellierten Umgebungsparameter sind durch die Integration des Manipulators in die Kommunikationsinfrastruktur des physikalischen Testfeldes abfangbar und manipulierbar. Da jeglicher Datenaustausch über die zentrale nachrichtenorientierte Middleware gemäß dem einheitlichen Datenmodell stattfindet ist hierdurch nach wie vor ein konsistenter Datenstrom durch die zwischengeschaltete Fehlerinjektion gewährleistet. Der Manipulator stellt hierbei die folgenden Möglichkeiten der Manipulation durch das Abfangen, Modifizieren und erneutes Veröffentlichen der Daten als intermediäre Komponente bereit.

- Existierende Datenobjekte, welche die Testfeld-Komponenten oder das zu testende System erzeugten, werden vor der Weiterleitung an die Datensenke durch die Kommunikationsinfrastruktur zunächst an den Manipulator zwischengesandt. Mit der Hilfe des Manipulators ist es möglich die realen Werte zu verändern und somit das Testszenario gemäß den Anforderungen zu manipulieren und dies dementsprechend durch die Beobachtung zu überprüfen. Das ursprüngliche aus dem physikalischen Testfeld erzeugte Datum, das tatsächlich vorherrscht, wird in diesem Fall ersetzt.
- Neue Datenobjekte ergänzen das real vorherrschende Szenario der Testfeld-Infrastruktur. Hierfür ist der Manipulator in der Lage neue Instanzen des einheitlichen Datenmodells gemäß S-100 der Kommunikationsinfrastruktur des Testfeldes zu erzeugen und sie über die Kommunikationsinfrastruktur an die jeweilige Datensenke zu senden [DJMT96].
- Durch das Abfangen der übertragenen Daten durch die nachrichtenorientierte Middleware erfolgt die Fehlerinjektion auf der Schicht des Kommunikationsprotokolls durch

Operationen wie das Verzögern, Verwerfen, Neuordnen und Duplizieren der Nachrichten durch den Manipulator. Neben der zuvor beschriebenen Fehlerinjektion auf der Ebene der Syntax und Semantik von Daten ist dies eine Möglichkeit das Kommunikationsverhalten zu beeinflussen und entsprechende Fehlerzustände durch einen entsprechenden Injektionszeitpunkt herbeizuführen [DaJa95].

Das physikalische Testfeld nutzt als Kommunikationsinfrastruktur eine zentrale nachrichtenorientierte Middleware, die eine Schnittstelle zu dem Manipulator über die Publish-Subscribe-Kommunikationsform besitzt. Die Verwendung einer nachrichtenorientierten Middleware als Kommunikationsinfrastruktur eines steuerbaren Testfeldes besitzt insbesondere im Kontext eines CPS den Vorteil gezielt die Fehler beziehungsweise Steuerungsparameter an ausgewählte Empfänger des physikalischen Testfeldes zu senden. Durch die lose Kopplung der Kommunikationsinfrastruktur ist es einem Empfänger der manipulierten Daten nicht möglich die Manipulation als solche zu erkennen. Optional ist sofern dies notwendig ist die Modellierung der Art der Nachricht (Real existierendes Datum, verändertes Datum, generiertes Datum) durch ein bezeichnendes Attribut möglich. Weitergehend erreicht die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform das parallele Transportieren sowohl korrumpierter wie real existierender Datenströme, wie in Abbildung 31 illustriert.

Durch den dargestellten Lösungsansatz für die Manipulation und Beobachtung der Datenströme der Testfeld-Infrastruktur ist es möglich, das Verhalten des zu testenden Systems in der operativen Umgebung unter den Effekten der realen Fehler zu beobachten und die Fehlertoleranz der automatisierten Führungsfunktion gemäß den in Kapitel 2.2.3 beschriebenen Risikoanalysen und Sicherheitsanalysen zu evaluieren. Die Fehlerinjektion lässt sich insbesondere durch die in dem folgenden Kapitel 4.4.2 beschriebene Integration des virtuellen Testfeldes ergänzen, in dem Simulatoren die Fehlerdaten im Rahmen der softwarebasierten Fehlerinjektion in einem Test im Closed-Loop-Verfahren herleiten und über die Infrastruktur an das zu testende System senden. Für diesen Zweck ist es notwendig entsprechende Fehlermodelle in dem virtuellen Testfeld zu hinterlegen und die Daten diesem entsprechend durch Simulatoren zu erzeugen und übertragen.

#### **4.4.2 Integration des virtuellen Testfeldes**

Gemäß der Definition in Kapitel 2.2.5 stellt ein Testfeld eine virtuelle und/oder physikalische Testumgebung bereit in der das zu testende System integriert und durch entsprechende Methoden der V+V erprobt wird. Eine Zielsetzung dieser Arbeit ist ein nahtloser Übergang im Sinne der modellbasierten Systementwicklung von dem Test in der Simulationsumgebung des virtuellen Testfeldes bis zum Test in der Realwelt des physikalischen Testfeldes. Hierfür ist neben der Verwendung der physikalischen Testfeld-Komponenten die Integration von als virtuelle Testfeld-Komponenten dienende Simulatoren notwendig. Mit steigender Reife der Systementwicklung findet das sukzessive Ersetzen der virtuell repräsentierten Komponenten durch physikalische Varianten statt (A15).

Weitergehend ermöglicht die Integration mit dem virtuellen Testfeld die Komplementierung des Realweltszenarios, um die spezifizierten Testszenarien unter realen Bedingungen im Sinne eines gesteuerten Tests im Closed-Loop-Verfahren durchführen zu können. Durch die Komplementierung des Realweltszenarios mittels virtueller Testfeld-Komponenten ist die simulative Anpassung der Testumgebung möglich. Ein Beispiel für einen solchen Anwendungsfall ist ein Kollisionsverhütungsszenario in dem aus Sicherheitsgründen virtuell repräsentierte Schiffe als potenzielle Fremdschiffe genutzt werden (A17). Sicherheitskritische Systeme fordern die Überprüfung von Randfällen der Testszenarien indem Fehlerinjektionstests durch Simulatoren durchgeführt werden (A16). Unter anderem hierfür ist eine Manipulierbarkeit der Datenströme notwendig, um das tatsächlich vorherrschende Realweltszenario entsprechend der Testanforderungen anpassen und den Test somit in dem physikalischen Testfeld steuern zu können (A16). Neben der Bereitstellung und Einspeisung simulierter Daten in die nachrichtenorientierte Middleware des physikalischen Testfeldes erfolgt die Rückkopplung des physikalisch vorherrschenden Szenarios der Realwelt in das virtuelle Testfeld. Durch dieses Bewusstsein innerhalb der Simulation ist das steuernde Reagieren auf das Szenario der Realwelt im Sinne eines Tests im Closed-Loop-Verfahren möglich (A18).

Um diese Anforderungen durch die Integration eines physikalischen Testfeldes mit einem virtuellen Testfeld zu ermöglichen folgt die notwendige Betrachtung der wesentlichen Elemente und Konzepte eines virtuellen Testfeldes und deren Charakteristika. Wie die Abbildung 32 exemplarisch anhand der Architektur des virtuellen maritimen Testfeldes HAGGIS (engl. „Hybrid Architecture for Granularly, Generic and Interoperable Simulations“) der eMIR Referenzplattform zeigt, existiert eine integrierte Infrastruktur für verschiedene virtuelle Testfeld-Komponenten, wie Simulatoren für verschiedene Einsatzzwecke, Szenario-Editoren für die Definition der Simulationsszenarien sowie Werkzeugen für die Beobachtung und Analyse der Simulationsausführung.

Die Simulationsplattform stellt einen Szenario-Editor zur Verfügung, der ein Modell für das statische Einrichten eines Simulationsszenarios besitzt. Dieses umfasst fundamentale Komponenten und Entitäten aller genutzten Ressourcen, Akteure wie Schiffe und Umweltfaktoren. Die statisch definierten Szenarien dienen der maritimen Verkehrssimulation als Eingabe zur Implementierung, Ausführung und Beobachtung des Verhaltens mehrerer Schiffe. Eine Umweltsimulation komplementiert die Simulationsplattform durch maritime Faktoren, wie den Wind, die Gezeiten und Strömung. Mit der Hilfe einer Sensorsimulation ist es möglich realistische Sensormessungen zu simulieren, beispielsweise auf der Basis eines mit der Hilfe des Szenario-Editor definierten maritimen Verkehrsszenarios. Insbesondere die Sensorsimulation spielt in der Integration des physikalischen und virtuellen Testfeldes eine wesentliche Rolle, da die physikalischen Sensorkomponenten der maritimen Domäne hierdurch virtuell repräsentiert und ergänzt werden [HGBS15]. Die virtuellen Testfeld-Komponenten verwenden als Laufzeitinfrastruktur (engl. „Runtime Infrastructure“) für die Koordination der einzelnen Simulationen und den Datenaustausch eine High-Level-Architecture (HLA).

HLA ist ein internationaler Standard (IEEE 1516) für integrierte sowie verteilte Simulation, der eine konsistente Simulationsumgebung ermöglicht [KuDW00]. Unterstützt wird die

Interoperabilität des virtuellen Testfeldes insbesondere durch die Verwendung eines einheitlichen Datenmodells gemäß S-100, ähnlich wie das physikalische Testfeld. Besonders diese Aspekte der Interoperabilität, ausgeprägt durch die Kommunikationsinfrastruktur in Form der HLA und das einheitliche Datenmodell gemäß S-100, sind für die Integration mit der Infrastruktur des physikalischen Testfeldes von besonderer Bedeutung.

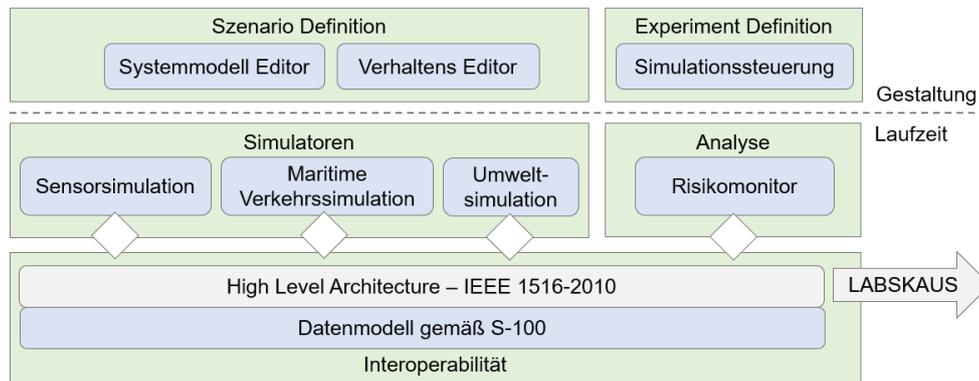


Abbildung 32: Architektur des virtuellen Testfeldes HAGGIS nach [HGBS15]

Abbildung 33 zeigt durch eine Systemarchitektur modelliert die elementaren Komponenten der integriert virtuell-physikalischen Testumgebung. Die verbundenen Kommunikationsinfrastrukturen des virtuellen und physikalischen Testfeldes stellen die bilaterale Datenübertragung zwischen den Simulatoren und physikalischen Testfeld-Komponenten her. Für diesen Zweck überträgt ein Simulations-Adapter die simulativ erzeugten Daten des virtuellen Testfeldes von der HLA in die nachrichtenorientierte Middleware mit einer Publish-Subscribe-Kommunikationsform des physikalischen Testfeldes. Weitergehend findet für die Rückkopplung das Überführen der Daten von der nachrichtenorientierten Middleware und anschließende Übertragung in die HLA durch den Simulations-Adapter statt. Die Rückkopplung ist für das Testen im Closed-Loop-Verfahren notwendig, um das Lagebild der vorherrschenden Situation im physikalischen Testfeld konsistent in der Simulation bereitzustellen und im Regelkreis die Sensorstimuli zu erzeugen, wie näher in Kapitel 2.2.4 erläutert.

Da die Kommunikationsinfrastrukturen des virtuellen und physikalischen Testfeldes unterschiedlich sind müssen entsprechende Transformationen stattfinden. Zwar basieren die Datenmodelle auf dem S-100 Framework zur Datenmodellierung, jedoch sind sie wegen unterschiedlicher Anforderungen andersartig ausgeprägt. Das S-100 konforme Datenmodell des virtuellen Testfeldes besitzt Metainformationen über den Zustand der Simulation und weitere Daten unter anderem für die Steuerung, die in dem physikalischen Testfeld nicht genutzt und erfasst werden. Da beide Datenmodelle auf dem einheitlichen Datenmodell gemäß S-100 basieren ist die Syntax interoperabel. Notwendig ist eine Komponente in der Systemarchitektur, welche die Transformation des Kommunikationsprotokolls und der Semantik der Daten adressiert, wie im Folgenden beschrieben. Das virtuelle Testfeld muss in der Lage sein das der Simulation zu Grunde liegende Simulationsmodell in eine derartige Form zu überführen, dass die Daten der Simulation in dem physikalischen Testfeld genutzt werden können.

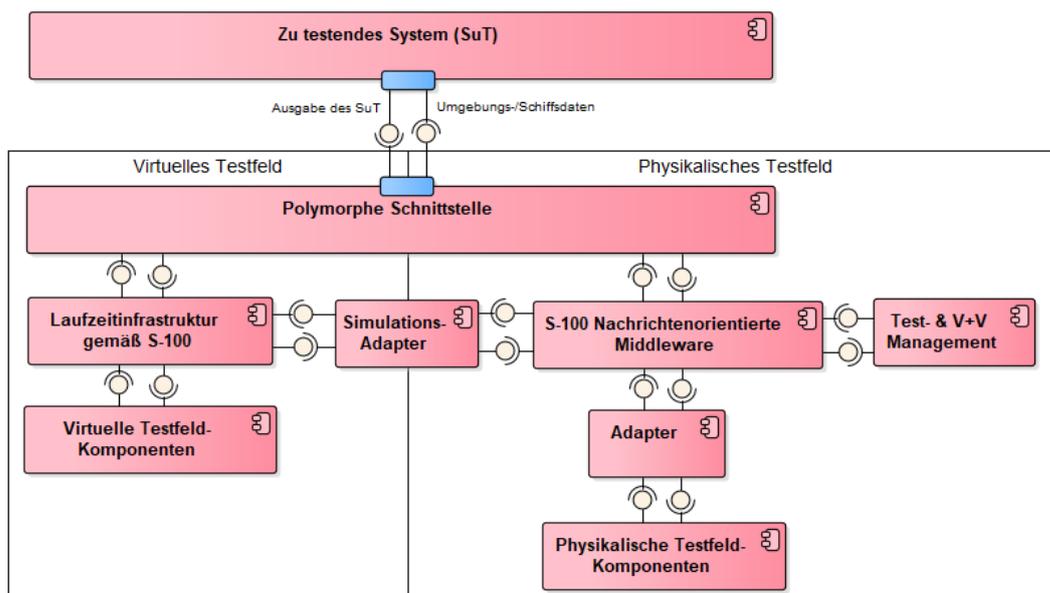


Abbildung 33: Integrierte Systemarchitektur des virtuellen und physikalischen Testfeldes

Das Functional Mock-Up Interface (FMI) ist ein weitläufig etablierter Standard für den Austausch von Simulationsmodellen und die Steuerung einer Co-Simulation. Für den Austausch von Simulationsmodellen erfolgt die Beschreibung der Schnittstelle der Simulatoren in XML und des Verhaltens in C [APEW13]. Eine XML beschreibt alle Variablen des Simulationsmodells die einem gemäß dem FMI Beschreibungsschema definierten Schema folgt, welches durch XSD spezifiziert ist. Für die Ausführung der Modelle durch eine Simulation existiert eine Programmierschnittstelle in Form von ausführbaren Dynamic Link Library (DLL) Dateien oder Quellcode in der Programmiersprache C [Fmid14].

FMI ist eine repräsentative Lösung für den Datenaustausch von simulativ erzeugten Daten zwischen verschiedenen Komponenten. Das physikalische Testfeld nutzt eine ähnliche Methode, um das virtuelle Testfeld zu integrieren. Wie Abbildung 34 darstellt basieren sowohl das virtuelle wie das physikalische Testfeld auf einem Datenaustausch basierend auf dem einheitlichen Datenmodell gemäß S-100. Daher stellt das virtuelle Testfeld die S-100 konformen Datenobjekte über eine Schnittstelle bereit, welche sämtliche im Simulationsmodell enthaltenen Daten umfassen.

In diesem Kontext erfüllt der Simulations-Adapter verschiedene Funktionen auf einigen der in Kapitel 2.1.4 vorgestellten Ebenen der Interoperabilität, ähnlich wie die polymorphe Schnittstelle oder der Datenverteiler einer Testfeld-Komponente. Der Simulations-Adapter übersetzt zwischen den verschiedenen Kommunikationsprotokollen des virtuellen und physikalischen Testfeldes. Der Simulations-Adapter ist modular und erweiterbar aufgebaut, in dem mit der Hilfe einer statischen Datenstromverarbeitungskette bestehend aus den Kommunikations-Handler, Syntax-Handler und Semantik-Handler ein interoperablen Datenaustausch zwischen virtuellem und physikalischem Testfeld ermöglicht wird.

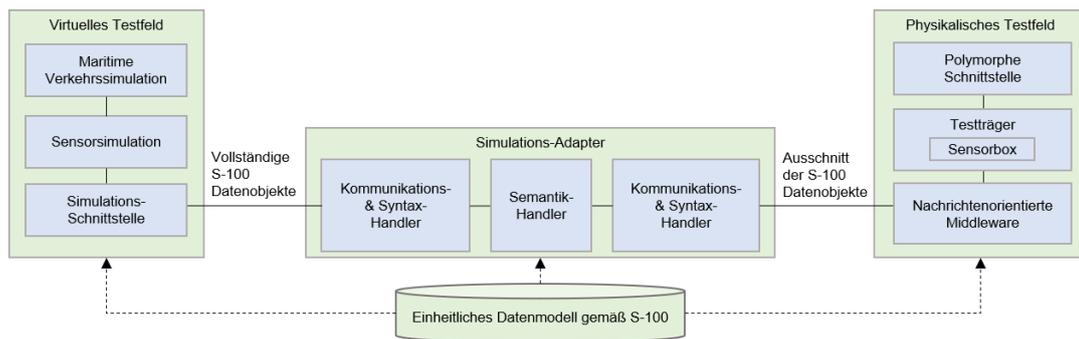


Abbildung 34: Simulations-Adapter für die Kopplung der HLA des virtuellen Testfeldes und nachrichtenorientierten Middleware des physikalischen Testfeldes

Da die Schnittstelle zu der Laufzeitinfrastruktur beziehungsweise den virtuellen Testfeld-Komponenten als Bestandteil der internen IT-Infrastruktur wiederverwendbar und statisch ist erfolgt keine Berücksichtigung polymorpher Konzepte, wie es für die dynamische Integration beliebiger zu testender Systeme notwendig wäre. Jedoch weichen die Aufgaben des Semantik-Handlers zu denen des statischen Adapters ab, welcher für die Integration der Testfeld-Komponenten genutzt wird. Alle für das physikalische Testszenario relevanten Informationen des Datenobjekts gemäß S-100 müssen als ein Teilausschnitt des gesamten Simulationsmodells gefiltert und in die nachrichtenorientierte Middleware gesendet werden, wie AIS und Radar. Hierfür bietet der Semantik-Handler des Simulations-Adapters für einen Benutzer die Fähigkeit, die relevanten Daten auszuwählen und nur diese an die zentrale nachrichtenorientierte Middleware durch einen entsprechenden Publisher zu senden. Dieser Prozess ist ebenso in entgegengesetzter Datenflussrichtung möglich, so dass das Szenario der Realwelt ebenfalls in der simulierten Umgebung abgebildet wird. Dies ermöglicht die Durchführung von Tests im Closed-Loop-Verfahren für zu testende Systeme gemäß einer definierten Spezifikation eines Testszenarios. Resultierend erhält das physikalische Testfeld im physikalischen Kontext keine unnötigen Daten des Simulationsmodells und müsste diese auf der Ebene der jeweiligen Testfeld-Komponenten verwerfen.

## 4.5 Polymorphe Schnittstelle

Aus den in Kapitel 2.3 hergeleiteten Anforderungen an eine Lösung der Zielsetzung dieser Arbeit und den in Kapitel 4.3.1 vorgestellten Methoden und Konzepten des S-100 Datenmodellierungsframeworks folgt der Bedarf nach einem Konzept zur Erhöhung der Generizität der Testfeld-Infrastruktur und einer effizienten Integration eines zu testenden Systems in das physikalische Testfeld. In diesem Rahmen ist ein Ziel dieser Arbeit die Gestaltung einer polymorphen Schnittstelle, um eine hohe Kompatibilität und Anpassbarkeit der Testfeld-Infrastruktur für zu testende Systeme aufzuweisen und somit das physikalische Testfeld wiederverwendbar und nachhaltig zu gestalten. Das physikalische Testfeld berücksichtigt gemäß der in Kapitel 3 beschriebenen Anforderungen der Polymorphie (A20) dass die Entwickler in der Regel nur das eigene System fokussieren und Interoperabilitätsmechanismen für

Testzwecke nicht berücksichtigen [Tolk13]. Daher ist eine dynamische Adaption des Datenmodells und Kommunikationsprotokolls gemäß der inhärenten Schnittstelle des zu testenden Systems notwendig, also eine Form der Polymorphie. Die Integration des zu testenden Systems führt zu anderen Herausforderungen der Interoperabilität, als es bei dem Einsatz der in Kapitel 4.3.3 beschriebenen Integration von Testfeld-Komponenten der Fall ist. Dies liegt darin begründet, dass das zu testende System stets wechselt und somit ein statischer Adapter regelmäßig auszutauschen beziehungsweise anzupassen wäre. Aus diesem Grund wird in der Testfeld-Architektur eine dedizierte Komponente für die dynamische Integration der zu testenden Systeme berücksichtigt, die wegen der Vielgestaltigkeit resultierend aus der Datenvielfältigkeit (A24) als polymorphe Schnittstelle bezeichnet wird.

Dieses Kapitel beschreibt zunächst das Konzept der Polymorphie im Kontext der Informatik als Lösungsansatz, um anschließend das dynamische Integrationskonzept der polymorphen Schnittstelle darzulegen. Die interne Datenstruktur des physikalischen Testfeldes in Form eines einheitlichen Datenmodells gemäß S-100 bietet zusammen mit der polymorphen Schnittstelle ein flexibles Konzept, um unterschiedliche Schnittstellen der zu testenden Systeme effizient bedienen zu können. Das Kapitel 4.3.1 beschrieb die Interoperabilitätsmethoden des S-100 Datenframeworks. Zwar existieren Gestaltungsprinzipien und Bedingungen für eine S-100 Interoperabilitätslösung, jedoch berücksichtigen diese nur einen interoperablen Datenaustausch zwischen S-100 konformen Produktspezifikationen. Das physikalische Testfeld weist wesentlich kompliziertere Interoperabilitäts Herausforderungen auf, denn neben möglichen S-100 Produktspezifikationen ist die Interoperabilität zwischen unter anderem den in Kapitel 2.1.3 vorgestellten Schnittstellen für den digitalen Datenaustausch der Schiffsführungssysteme notwendig [MaMo18].

Traditionelle Softwareentwicklung wird unter der Closed World Assumption (deutsch: Annahme zur Weltabgeschlossenheit) entwickelt, was zu der Annahme einer Nichtexistenz nicht modellierter Elemente führt. Beispielsweise werden zur Laufzeit nicht vorhandene Variablen als nichtexistierend betrachtet. Diese Annahme führt zu der Voraussetzung, dass die Grenzen zwischen den Systemen und der Umwelt bekannt sind und sich nicht verändern. Insbesondere fand diese Art der Entwicklung in der Vergangenheit die Anwendung in der streng spezifizierten Entwicklung von sicherheitskritischen Systemen [BaDG06]. Die Closed World Assumption blockiert die Entwicklung in einer technologischen Umwelt von intelligenten und datengetriebenen Systemverbänden, die im Rahmen eines cyber-physischen Systems kooperieren, wie in der Problematik in Kapitel 1.2 erörtert.

Es existieren verschiedene Typen der Polymorphie im Kontext der Softwareentwicklung. Der Begriff der Polymorphie im Sinne der Informatik unterteilt sich in zwei Über-Kategorien: Universelle und Ad-Hoc Polymorphie. In der Literatur finden sich jeweils zwei weitere Unter-Kategorien, deren Charakteristika im Folgenden erläutert werden. Die parametrische Polymorphie bezeichnet die Anpassung eines Elements auf Basis einer Menge an möglichen Typen. In dieser Ausprägung existieren verschiedene Arten des Datentyps, die je nach Kontext ausgewählt und sich wie definiert verhalten. Mithilfe von Parametern wird einem polymorph reagierenden Element die konkrete Ausprägung vorgegeben. Parametrische Poly-

morphie wird als eine Form der Nutzung von Vorlagen bezeichnet. Als Inklusionspolymorphie wird das Prinzip der Vererbung bezeichnet, in dem eine Komponente die Elemente anderer Komponenten übernimmt. Somit existieren Unterkomponenten, welche Elemente von vererbenden Komponenten übernehmen. Daher gehört das inklusionspolymorph variierende Objekt zu mehreren Teilelementen und vereint diese. In der Überladung wird eine identische Bezeichnung für dasselbe Element genutzt, um im Kontext der Laufzeit die jeweilige Logik auszuführen. Es existieren unterschiedliche Methoden oder Typen mit gleicher Bezeichnung. In einer Ad-Hoc Form der Polymorphie wird während der Laufzeit auf Basis des Kontextes entschieden, welche Implementierung ausgeführt wird. Die erzwungene Polymorphie beziehungsweise Coercion ist ähnlich der Überladung und bezeichnet eine Polymorphie mit der Notwendigkeit ein Element in den erwarteten Typ umzuwandeln. Um den Ziel-Typ der Umwandlung des Elements zu definieren dient die Betrachtung der jeweiligen Umwelt [CaWe85].

In dieser Arbeit bezeichnet die polymorphe Schnittstelle ein anpassbares Kopplungselement, das auf der Basis von Rahmeninformationen auf unterschiedliche maritime Standards wandelbar ist und somit die Integration unterschiedlicher zu testender Systeme zur Laufzeit erlaubt. Diese Art der Polymorphie basiert auf der erzwungenen Polymorphie beziehungsweise Coercion. Die polymorphe Schnittstelle erfüllt durch die Überladung des Funktionsaufrufes durch die jeweils konfigurierte Transformationsverarbeitung die Verbindung mehrerer Systeme dynamisch zur Laufzeit. In der physikalischen Testfeld-Architektur dient die polymorphe Schnittstelle zur dynamischen Integration als Mediator zwischen den inhärenten Schnittstellen eines zu testenden Systems und dem einheitlichen Datenmodell gemäß S-100 des physikalischen Testfeldes. Mithilfe des Konzepts der Polymorphie werden die Datenmodelle und Schnittstellen der verschiedenen Komponenten auf den in Kapitel 2.1.4 beschriebenen Interoperabilitäts-Ebenen verbunden.

#### **4.5.1 Transformationsmodul für Datenströme**

Eine Transformation im Kontext der Softwareentwicklung bezeichnet zum einen eine Modell-zu-Code-Transformation, durch die automatisiert aus einem formalen Modell ein Programmcode generiert wird. Zum anderen charakterisiert die Modell-zu-Modell-Transformation die Überführung eines Quellmodells in ein andersartiges Zielmodell. Das Anwendungsszenario der polymorphen Schnittstelle stellt die Notwendigkeit einer Modell-zu-Modell-Transformation dar, um unterschiedliche Modelle interoperabel zu gestalten [SVEH12, S.33]. Die polymorphe Schnittstelle muss gemäß der in dem jeweiligen Testfall genutzten inhärenten Schnittstellen des zu testenden Systems eine komplexe Datenverarbeitung unterstützen, die sich mit wechselnden Tests ebenfalls verändert. Daher etabliert diese Arbeit einen Ansatz, der wiederverwendbare Softwaremodule in einer austauschbaren Art und Weise realisiert. Eine unflexible Gestaltungsmöglichkeit wäre ein monolithischer Ansatz, bei dem in einem Modul die Transformationsanweisungen und Logik implementiert wird. Dies würde die Umgestaltung der polymorphen Schnittstelle und die Wiederverwendbarkeit stark einschränken sofern sich die Transformationsanforderungen ändern [Sher10, S.276].

Durch das in dieser Arbeit vorgestellte Konzept der polymorphen Schnittstelle ist mühelos die Zerlegung der auszuführenden Verarbeitungsschritte möglich, beispielsweise gemäß den Ebenen der Interoperabilität. Abbildung 35 stellt die konzeptionelle Softwarearchitektur der polymorphen Schnittstelle dar, welche gemäß Kapitel 2.1.4 die Verarbeitungsschritte für die Transformation ermöglicht. Die polymorphe Schnittstelle besteht aus einer linearen Sequenz von Softwaremodulen, die entsprechend der Interoperabilitätsebenen die nötigen Interpretations- und Transformationsaufgaben übernehmen. Diese Gestaltungsentscheidung basiert auf dem Softwarearchitekturmuster der Pipes und Filter [Sher10, S.276ff.].

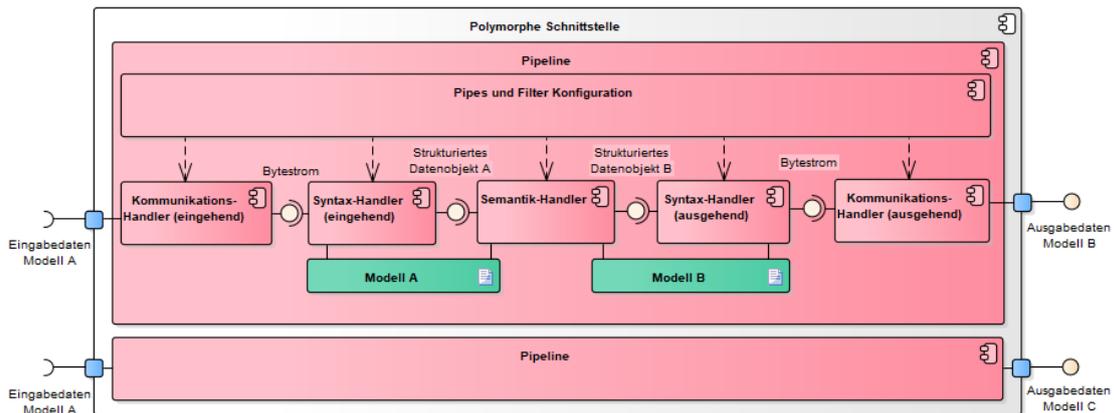


Abbildung 35: Softwarearchitektur der polymorphen Schnittstelle

In einer Umsetzung des Softwarearchitekturmusters der Pipes und Filter durchlaufen die Dateneingaben eine Menge von Verarbeitungsschritten, die als Filter bezeichnet werden. Ein Filter stellt eine Umwandlung der Daten dar. Die Filter als Transformationskomponenten werden in einer entsprechenden Reihenfolge geordnet und bieten die Möglichkeit, rekonfiguriert zu werden. Die Verbindungen zwischen den Filtern werden als Pipe bezeichnet. Eine Pipe regelt den Datenstrom und synchronisiert die Filterkomponenten. Dieses Softwarearchitekturmuster unterstützt die von der polymorphen Schnittstelle geforderte Erweiterbarkeit für beliebige dynamische Integrationen eines zu testenden Systems. Insbesondere in der Datenstromverarbeitung findet dieses Konzept weitläufige Verwendung. Aus diesem Grund wird für das datenstromorientierte physikalische Testfeld für automatisierte Schiffsführungssysteme ebenfalls dieses Softwarearchitekturmuster angewandt [BuLö00, S.55ff.], [HoWo04, S.71ff.].

**Pipeline.** Der Lösungsansatz für die dynamische Integration eines zu testenden Systems erfolgt durch das Konfigurieren von beliebigen gemäß dem Testszenario implementierter oder bereits existierender Filter in einer Pipeline. Insbesondere vermeidet dieser Lösungsansatz die Entwicklung redundanter Implementierungen, da die Wiederverwendung der jeweiligen Umsetzung in einem dedizierten Filter stattfindet [Sher10, S.276f.]. Die polymorphe Schnittstelle besteht aus mindestens einer oder mehrerer Pipelines zur Bedienung von multiplen Schnittstellen. Weitergehend ermöglicht die polymorphe Schnittstelle eine parallelierte Verarbeitung von multiplen Datenströmen durch die Bereitstellung mehrerer Pipelines, wie in Abbildung 35 dargestellt. Hierdurch wird eine höhere Leistung bei der Transformation durch das physikalische Testfeld erreicht.

**Kommunikations-Handler.** Da die polymorphe Schnittstelle für die Kommunikation im Rahmen eines CPS eingesetzt wird müssen verschiedene Kommunikationsprotokolle unterstützt werden, wie NMEA, CAN oder UDP. Um die Interoperabilität zwischen zwei heterogenen Schnittstellen zu erlangen existieren die Instanzen der Kommunikations-Handler für eingehende und ausgehende Datenströme. Ebenso wie die strikt standardisierten Technologien der physikalischen Konnektivität (RJ45, RS232 und weitere) sind einige Betrachtungsgegenstände der Kommunikationsschicht in Form von Übertragungsmechanismen und Transportsicherung ebenfalls in einem hohen Maße gemäß Standards spezifiziert. Die in dieser Arbeit betrachteten Aspekte der Kommunikationsprotokolle für verschiedene Sensoren oder Dienste, mit deren Hilfe eine automatisierte Schiffsführungsfunktion realisiert wird, sind üblicherweise auf den anwendungsorientierten Schichten des OSI-Modells eingeordnet.

**Syntax-Handler.** Ein Syntax-Handler für eingehende Daten nimmt die vom Kommunikations-Handler empfangenen und verarbeiteten Datenströme entgegen und übersetzt diese in eine Modellrepräsentation, beispielsweise in einer Syntax gemäß XML. Diese Umwandlung des Datenstroms in ein strukturiertes Datenobjekt wird als Deserialisierung bezeichnet. Im Falle des Syntax-Handler für ausgehende Daten erfolgt eine Transformation der Modellrepräsentation in die durch den ausgehenden Kommunikations-Handler erforderliche Datenrepräsentation. In diesem Fall serialisiert der Syntax-Handler die Modellrepräsentation des Datenobjekts in einen Bytecode, der vom Kommunikations-Handler entsprechend des notwendigen Kommunikationsprotokolls übermittelt wird. Die korrekte Entsprechung des Datenobjekts gemäß der in Form eines Modells zu Grunde liegenden Datenrepräsentation wird durch eine Modellvalidierung durchgeführt, nachdem ein Datenobjekt vom Syntax-Handler entgegengenommen oder bei erfolgter semantischer Transformation an den Syntax-Handler weitergegeben wird. Die Validierung der jeweiligen syntaktischen Modellrepräsentation findet auf der Basis eines zu Grunde liegenden Datenmodells in beliebiger Ausprägung statt, beispielsweise modelliert durch die UML oder XML Schema Definition (XSD).

Durch die Kombination eines eingehenden und ausgehenden Syntax-Handler erfolgt die syntaktische Transformation der Struktur (beispielsweise Elemente oder Elementgruppen), Datenrepräsentation (beispielsweise XML oder JSON) und Datentypen (beispielsweise Zahlen oder Texte) der Datenobjekte. Eine Unterscheidung zwischen der Syntax und der Struktur eines Datenobjekts erfolgt nicht in dieser Arbeit, da beide Transformationen sinnvoll durch einen Filter der polymorphen Schnittstelle erfolgen können. Die Bedeutung der ausgetauschten Daten ist dem Syntax-Handler unbekannt [BBBL14].

**Semantik-Handler.** Um zwei oder mehr Systeme miteinander kooperieren zu lassen ist das einheitliche Wissen über die Bedeutung der ausgetauschten Daten notwendig. Durch die semantische Transformation werden die übertragenen Daten hinsichtlich der Bedeutung analysiert und entsprechend der definierten Bedeutung interpretiert und transformiert. Der Semantik-Handler der polymorphen Schnittstelle transformiert ein Datenobjekt in Form einer validen Modellinstanz in ein Datenobjekt einer anderen Modellinstanz. Aus diesem Grund benötigt der Semantik-Handler zwei Modelle in beliebiger Repräsentation, beispiels-

weise modelliert durch die UML oder XSD. Die Herausforderung der semantischen Transformation liegt darin begründet, dass in unterschiedlichen Datenrepräsentationen verschiedene Bezeichnungen die gleiche Bedeutung synonym verwenden (Synonymie) oder verschiedene Bedeutungen das Gleiche referenzieren (Polysemie). Der Lösungsansatz in dieser Arbeit für die semantische Interoperabilität ist die Verwendung eines einheitlichen Datenmodells gemäß S-100 als Referenzpunkt, also unter anderem eine kontrollierte Taxonomie für die Wissensrepräsentation, wie in Kapitel 4.3.1 beschrieben.

Die semantische Transformation des Semantik-Handlers stellt wegen der Interpretation und des nötigen Wissens über die Daten die wesentliche Herausforderung der polymorphen Schnittstelle dar. Aus diesem Grund beschreibt das folgende Kapitel den Ansatz zur semantischen Transformation mit der Hilfe eines einheitlichen Datenmodells der polymorphen Schnittstelle im Detail.

#### 4.5.2 Semantische Annotation mit dem S-100-Referenzmodell

Das vorherige Kapitel beschrieb die Konzepte der polymorphen Schnittstelle als Transformationskomponente des physikalischen Testfeldes für die dynamische Integration von heterogenen zu testenden Systemen. Die besondere Herausforderung der semantischen Interoperabilität zwischen zwei Systemen stellt sich hierbei gegenüber einer Interoperabilität der Kommunikation und Syntax als besonders anspruchsvoll heraus. Eine wesentliche Gestaltungsentscheidung der Testfeld-Architektur ist die Verwendung eines einheitlichen Datenmodells gemäß S-100, wie in Kapitel 4.3.1 erläutert. Das Lösungskonzept für die Interoperabilität in dieser Arbeit basiert auf der Methode einer transitiven Abbildung zweier unterschiedlicher Systeme mit einer einheitlichen Referenz [MiTa04]. Dies geschieht durch die Verwendung des Referenzmodells gemäß S-100 als einheitliches Datenmodell und der semantischen Annotation der Modelle zweier unterschiedlicher Softwaresysteme, wie in Abbildung 36 dargestellt.

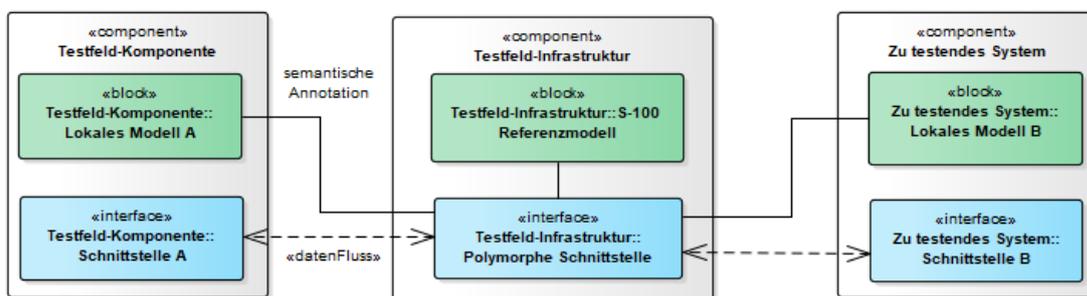


Abbildung 36: Komponenten für die transitive Abbildung einer semantischen Annotation mit dem einheitlichen Datenmodell gemäß S-100

Das S-100 Framework besitzt eine standardisierte Struktur und Taxonomie, die für die semantische Annotation verwendet wird. Insbesondere da die maritime Domäne das S-100 Framework zur Modellierung von Daten ebenfalls mit der Verwendung eines einheitlichen Referenzmodells und einer einheitlichen Taxonomie als Interoperabilitätskonzept nutzt, scheint die Verwendung eines hiermit kompatiblen Interoperabilitätskonzeptes schlüssig.

Die detaillierte Beschreibung der Strukturen, Relationen und des Konzepts des S-100 Frameworks und deren Etablierung in dem Konzept der Testfeld-Architektur bietet Kapitel 4.3.1.

Das Konzept der semantischen Transformation nutzt Modelle als repräsentatives Abbild der Realität. Durch ein Modell des zu testenden Systems und des einheitlichen Referenzmodells gemäß S-100 ist es möglich, die Beziehungen und Strukturen der Modelle semantisch zu annotieren und interpretieren. Ein Modell ist nicht zwingend technisch, sondern repräsentiert eine fachliche und funktionelle Beschreibung des Systems. Ein lokales Modell beschreibt die Strukturen und Elemente der inhärenten Schnittstelle eines Systems, welches in die Infrastruktur des physikalischen Testfeldes integriert wird.

Das Ziel der semantischen Annotation ist mit der Hilfe eines Referenzmodells einen Referenzpunkt für die lokalen Modelle der jeweiligen zu testenden Systeme und der Komponenten des physikalischen Testfeldes zu bieten, um über eine gemeinsame Taxonomie und Modellierung semantisch interoperabel zu sein. Hierfür werden semantische Annotationen für die relevanten Informationen des lokalen Modells definiert. Dies geschieht in Form einer automatisiert oder manuell erstellten Abbildung zu dem S-100 Referenzmodell, wie in Abbildung 37 dargestellt. Bei dem Schritt der semantischen Annotation findet das Referenzieren der Strukturen und Beziehungen zweier konzeptioneller Modelle statt.

Die semantischen Annotationen stellen Regeln und Vorschriften für die Übersetzung zwischen den unterschiedlichen konzeptionell ausgeprägten Datenmodellen dar. Hierdurch lassen sich die Beziehungen zweier oder mehrerer Modelle über einen gemeinsamen transitiven Nenner darstellen. Die semantischen Annotationen folgen einer einheitlichen formalen Sprache, die entweder selbst definiert oder durch eine entsprechende plattformunabhängige Transformationssprache vordefiniert sein können, wie der Extensible Stylesheet Language Transformation (XSLT).

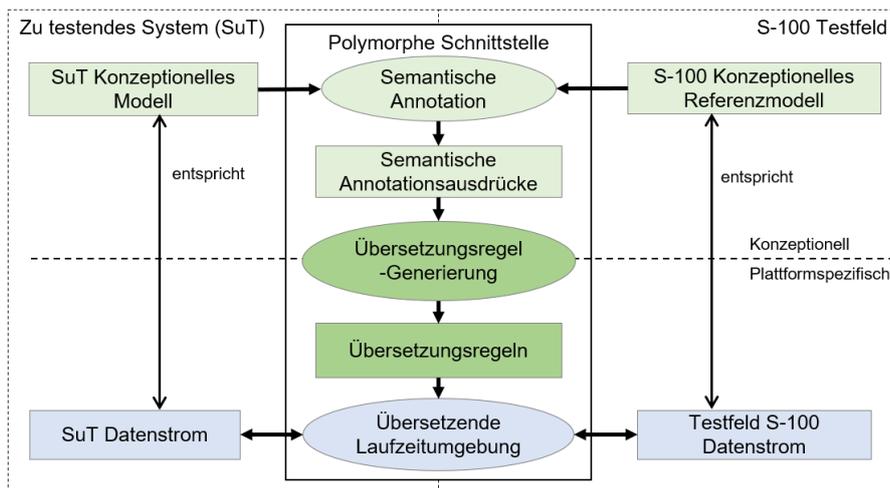


Abbildung 37: Methode der semantischen Transformation durch die polymorphe Schnittstelle nach [MiTa04, S.631]

Unter dem Einbezug eines Domänenexperten werden mit der formalen Sprache die lokalen Modelle der zu interoperierenden Systeme semantisch über das Referenzmodell annotiert.

Dieser Vorgang erfolgt indem der Domänenexperte die Strukturen, Attribute und Beziehungen des lokalen Modells analysiert und mit dem einheitlichen Datenmodell gemäß S-100 als Referenzmodell des physikalischen Testfeldes annotiert.

Ein wesentlicher Schritt der semantischen Annotation stellt die Analyse der Modelle der zu interoperierenden Systeme im Hinblick auf mögliche Interoperabilitätskollisionen (engl. „Interoperability Clashes“) und den Ausdruck dieser Kollisionen dar. Die Analyse und Bewertung der auftretenden Interoperabilitätskollisionen schaffen ein Bewusstsein und die Belegbarkeit des Informationsgehalts, der im Rahmen der Transformation existent ist. Interoperabilitätskollisionen werden in zwei Kategorien eingeteilt.

**Verlustfreie Kollision.** Die durch die semantische Annotation transformierbaren Interoperabilitätskollisionen in der transitiven Relation werden der Gruppe der verlustfreien Kollisionen zugeordnet. Ein eingehendes Datentupel wird in ein ausgehendes Datentupel transformiert, das von der Repräsentation zwar abweicht, den Informationsgehalt nach wie vor vollständig beinhaltet. Typischerweise fallen unterschiedliche Bezeichnungen oder abweichende Syntax in die Kategorie der verlustfreien Kollisionen, die entsprechend transformiert werden.

**Verlustbehaftete Kollision.** In diese Kategorie werden Interoperabilitätskollisionen zugeordnet, deren Transformation den Verlust von Informationen verursacht und dies nicht zu vermeiden ist. Insbesondere tritt dieser Fall auf, sofern unterschiedlich präzise Datentupel vorhanden sind oder in einem Datenmodell die Informationen des anderen nicht existieren.

Es ist zu berücksichtigen, dass eine völlig sichere Datenkommunikation lediglich für zwei lokale Modelle garantiert wird, die komplett verlustfreie Interoperabilitätskollisionen mit dem Referenzmodell aufweisen. Andernfalls ist eine Beeinflussung der Datenkommunikation in Form eines Informationsverlustes möglich. Tabelle 3 stellt die auftretenden Interoperabilitätskollisionen auf der Basis einer Vielzahl von praktischen Erfahrungen dar, die durch einen Domänenexperten analysiert und berücksichtigt werden müssen [MiTa04, S.624].

<i>Verlustfreie Kollisionen</i>	<i>Verlustbehaftete Kollisionen</i>
Kodierung: Verschiedene Formate von Daten oder Einheiten	Inhalt: Verschiedener Inhalt durch das selbe Konzept gekennzeichnet
Typisierung: Verschiedene Datentypen zur Repräsentation von Informationen	Umfang: Unterschiede der Anwesenheit/Abwesenheit von Informationen
Benennung: Verschiedene Namen für denselben Inhalt	Präzision: Genauigkeit der Informationen
Strukturierung: Verschiedene Strukturen desselben Inhalts	Abstraktion: Niveau der Spezialisierung in der Aufbereitung der Informationen
	Granularität: Niveau der Zerlegung in der Aufbereitung der Informationen
	Datenfeldlänge: Nummer der Bytes eines Attributfeldes

Tabelle 3: Interoperabilitätskollisionen und mögliche Verluste nach [MiTa04, S.624]

Auf der Basis der semantischen Annotationsausdrücke und den zu Grunde liegenden Modellen werden die Übersetzungsregeln generiert, wie die Abbildung 37 darstellt. Die Übersetzungsregeln repräsentieren in einer plattform- und programmiersprachenspezifisch interpretierbaren Weise die Zuordnungen der semantischen Annotationen und der Modelle. Dies ist beispielsweise durch ein Java-Softwaremodul möglich, welches die plattform- und programmiersprachenunabhängigen Annotationsausdrücke ausführt, die beispielsweise durch XSLT repräsentiert vorliegen. Die Übersetzungsregeln werden durch eine entsprechend übersetzende Laufzeitumgebung ausgeführt die in der Lage ist, diese zu interpretieren und somit die durch die Schnittstellen ausgetauschten Datenströme zu transformieren und den Datenfluss zwischen heterogenen Systemen zu ermöglichen [MiTa04].

### **4.5.3 Transformationsoperatoren des Kommunikationsverhaltens**

Der in Kapitel 4.5.2 vorgestellte Lösungsansatz auf der Basis der semantischen Annotation zweier unterschiedlicher Datenmodelle konzentriert sich auf die syntaktische und semantische Transformation der Datenobjekte. Zwar ist es hierdurch möglich ein Datenobjekt von einem Modell in ein anderes zu transformieren, jedoch ist eine Transformation des Kommunikationsverhaltens zweier unterschiedlicher Kommunikationsprotokolle hierdurch nicht möglich, wie den in Kapitel 2.1.3 beschriebenen NMEA0183, NMEA2000 oder IVEF. Dies ist jedoch notwendig sofern ein zu testendes System eine vom physikalischen Testfeld in einem Datenobjekt repräsentierte Nachricht in zwei Nachrichten übertragen benötigt. Veränderungen des Kommunikationsverhaltens umfassen weitergehend die zeitlich definierte Bereitstellung der Datennachrichten. Dies ist beispielsweise in dem Fall einer begrenzten Verfügbarkeit des Kommunikationskanals und entsprechender Vorhaltung der Daten in einer Warteschlange oder der Transformation von einem Pull-basiertem Kommunikationsprotokoll in ein Push-basiertes notwendig. Bezüglich des Kommunikationsverhaltens treten nach [Buse12, S.24] verschiedene Interoperabilitätskollisionen auf, wie das Hinzufügen zusätzlicher Nachrichten zu der eingehenden, das Erzeugen mehrere Nachrichten aus einer Nachricht oder das Rekonstruieren fehlender Daten durch das Anfragen an Datenquellen.

Für die Auflösung dieser Interoperabilitätskollision existieren etablierte Methoden für die Transformation des Kommunikationsverhaltens. Grundsätzlich erfolgt die Lösung des Problems durch die formale Beschreibung des Quellprotokolls und Zielprotokolls und der Kommunikationsmuster, auf dessen Basis die entsprechenden Transformationsoperatoren in einer Sequenz geordnet werden. Durch die formale Beschreibungen des Quellprotokolls und Zielprotokolls, beispielsweise durch die Prozessmodellierungssprache Business Process Execution Language (BPEL) oder durch UML Sequenzdiagramme, ist es im Rahmen der modellgetriebenen Softwareentwicklung möglich, entsprechende Komponenten für die Transformation des Kommunikationsverhaltens automatisiert zu erstellen [QPPS08], [RTTZ04].

Zunächst erfolgt die Konfiguration des jeweiligen Quellprotokolls und Zielprotokolls durch das Konfigurationsmodul für Pipelines in der polymorphen Schnittstelle. Für das

Quellprotokoll dient der eingehende Kommunikations-Handler und für das Zielprotokoll der ausgehende. Die Verhaltensänderung findet auf der Ebene der Kommunikations-Handler statt. Der Kommunikations-Handler besitzt hierfür eine Sequenz von Transformationsoperatoren gemäß den Anforderungen der inhärenten Schnittstelle des zu testenden Systems. Infolge der jeweiligen Interoperabilitätskollision zwischen Quellprotokoll und Zielprotokoll erfolgt die Bereitstellung der entsprechenden Operatoren, die im Folgenden erläutert werden und auf dem differenziellen Verhaltensmodell (engl. „Differential Behavior Model“) nach [Buse12] basieren. Die Auswahl dieses Lösungsansatzes liegt darin begründet, dass die vorgeschlagenen Operatoren und Modelle zur Beschreibung der Transformation des Kommunikationsverhaltens effizient mit der in Kapitel 4.5.1 beschriebenen Gestaltung der polymorphen Schnittstelle kombiniert werden können. Eine modulare und austauschbare Gestaltung ist hierdurch möglich. Alternative Ansätze wie [RTTZ04] und [MBMC07] verfügen über ähnliche Transformationsoperatoren zur Lösung der auftretenden Interoperabilitätskollisionen.

- **Hinzufügen.** Sofern eine neue Nachricht erzeugt wird erfolgt zunächst die Weiterleitung der eingehenden Nachricht ohne Änderungen. Für die neu erzeugte Nachricht wird das Ziel definiert, das entweder die Datenquelle, Datensenke oder eine weitere Zieladresse ist. Insbesondere ist die Spezifikation des Übertragungszeitpunkts möglich, um ggf. das Senden der hinzugefügten Nachricht zu verzögern beziehungsweise zeitlich abzustimmen.
- **Aufteilen.** In einigen Fällen ist es erforderlich, dass eine eingehende Nachricht in zwei oder mehrere ausgehende Nachrichten aufgeteilt wird.
- **Löschen.** Der Operator zum Löschen von Nachrichten bewirkt das Entfernen der eingehenden Nachricht oder einzelner Teile dieser.
- **Anfragen.** Einige Kommunikationsprotokolle erfordern das Anfragen zusätzlicher Nachrichten bei den Testfeld-Komponenten beziehungsweise weiterer Datenquellen. Wie bei dem Hinzufügen Operator erfolgt die Weiterleitung der eingehenden Nachricht ohne Änderungen. Ergänzend ereignen sich beliebig viele Anfrage-Nachrichten und die zeitlich definierte Vorhaltung der jeweiligen Antwort in dem ausgehenden Kommunikations-Handler. Durch eine weitere Nachricht ist es möglich die angefragte Nachricht an die jeweilige Zieladresse weiterzuleiten. Die Informationen, welche die eingehende Nachricht beinhaltet, können hierbei zur Generierung der Anfrage-Nachricht verwendet werden.
- **Speichern.** Der ausgehende Kommunikations-Handler speichert die Nachricht für eine definierte Zeit, bis diese entsprechend abgefragt wird.
- **Kombinieren.** Der Kombinations-Operator fasst mehrere Nachrichten zu einer einzelnen zusammen. Hierfür ist eine Nutzung des Speichern- oder Anfrage-Operator notwendig, um die benötigten Nachrichten zuvor für die Kombination vorzuhalten beziehungsweise die zu kombinierenden Nachrichten anzufragen.

Da insbesondere im Kontext der Testfeld-Architektur und automatisierten Schiffsführungssysteme drahtlose Kommunikationstechnologien für den Datenaustausch der verschiedenen

Sensoren und Systeme verwendet werden ist dem Übertragungszeitpunkt eine besondere Bedeutung beizumessen. Die Transformationsoperatoren des Hinzufügens, Kombinierens, Anfragens und Aufteilens verfügen daher über die Möglichkeit die Kommunikation über einen entsprechenden Zeitparameter zu definieren. Hierdurch sind Kommunikationsverhaltens-eigenschaften abbildbar, wie eine zeitliche Verzögerung oder zeitlich begrenzte Abschnitte zum Senden der Nachrichten [Buse12].

Die in diesem Kapitel beschriebenen Operatoren des Kommunikationsverhaltens dienen der Transformation zweier unterschiedlicher Kommunikationsprotokolle. In diesem Kontext ist es möglich verschiedene Operatoren auf die gleiche Nachricht anzuwenden und somit eine komplexe Transformationssequenz zu realisieren.

## 4.6 Zusammenfassung der Testfeld-Architektur

Mit dem Ziel die in Kapitel 1.3 beschriebene Zielsetzung zu erfüllen erfolgte die Erhebung von Anforderungen an eine Lösung in Kapitel 2.3. Die in Kapitel 4 aufgeführten Konzepte und Methoden des Testfeld-Architektur adressieren die Erfüllung dieser Anforderungen durch die Verankerung in den Konzepten des Zielsystems und der Systementwicklungsmethodik. Hierfür erfolgte die Systematisierung der generischen Testsystem-Architektur durch die Ausprägung anhand der Konzepte des Zielsystems in Form eines automatisierten Schiffsführungssystems, das als Anwendungsfall in Kapitel 4.1 beschrieben wurde. Die Testfeld-Architektur nutzt als Referenzpunkt für alle an der Kommunikation teilnehmenden Komponenten das einheitliche Datenmodell gemäß S-100. Eine nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform als Rückgrat des physikalischen Testfeldes integriert die verschiedenen des physikalischen Testfeldes. Über diese interne Kommunikationsinfrastruktur übertragen sämtliche Komponenten die Daten gemäß dem einheitlichen Datenmodell. Dies unterstützt insbesondere die Interoperabilität und die Funktionen der V+V, wie die Beobachtung durch das Monitoring und die Steuerung des physikalischen Testfeldes für Fehlerinjektionstests und Tests im Closed-Loop-Verfahren.

Eine polymorphe Schnittstelle dient als Kernelement der Erreichung von Generizität des physikalischen Testfeldes für eine nachhaltige und wiederverwendbare. Die polymorphe Schnittstelle nutzt verschiedene Methoden der Daten- und Kommunikationsverhaltenstransformation, um in modularer und konfigurierbarer Art und Weise die dynamische Integration der zu testenden System zu bewältigen. Hierdurch wird eine starke Entkopplung des Testfeldes und der jeweiligen zu testenden Systeme erreicht, in dem die interne IT-Infrastruktur des physikalischen Testfeldes abstrahiert wird.

Im Gegensatz zu den in Kapitel 3 vorgestellten physikalischen Testaufbauten der maritimen Domäne erlaubt die vorgestellte Testfeld-Architektur nicht nur die Durchführung eines im Vorfeld spezifizierten Tests durch einen starren Testaufbau. Vielmehr bietet sie die Reduktion der Aufwände für V+V für automatisierten Schiffsführungssysteme durch die Orientierung an den etablierten Systementwicklungsmethoden in nahtloser Ausprägung und

die Unterstützung sowohl von Tests im Open-Loop-Verfahren wie im Closed-Loop-Verfahren. Sämtliche Anforderungen wurden konzeptionell von der Testfeld-Architektur berücksichtigt. Im weiteren Verlauf der Arbeit dient die Realisierung der Testfeld-Architektur durch das physikalische Testfeld LABSKAUS der Darstellung der Anwendbarkeit und Bewertung der Zielerfüllung durch die Überprüfung zweier Hypothesen im Rahmen der Evaluation.

## 5 LABSKAUS – Labor für sicherheitskritische Analysen auf See

Dieses Kapitel beschreibt die Realisierung der entwickelten Testfeld-Architektur für ein physikalisches Testfeld zur Unterstützung der Systementwicklung von automatisierten Schiffsführungssystemen. Hierbei erfolgt die Umsetzung auf der Basis des physikalischen Testfeldes LABSKAUS im Kontext der eMIR Referenzplattform. Die IALA registrierte Testplattform eMIR (engl. „eMaritime Integrated Reference Platform“) kombiniert ein virtuelles und physikalisches maritimes Testfeld. eMIR ist ein von der deutschen Schifffahrtsindustrie initiiertes Projekt, um die Sicherheit und Effizienz des maritimen Transports zu erhöhen. Das Ziel von eMIR ist die Bereitstellung einer offenen und wiederverwendbaren Testumgebung, welche die Technologieentwicklung der maritimen Domäne unterstützt. Das physikalische Testfeld LABSKAUS ist in der deutschen Bucht verortet. Dennoch sind einige Komponenten transportabel und können ortsunabhängig genutzt werden [Hahn15]. LABSKAUS realisiert die in Kapitel 4 beschriebenen Elemente eines physikalischen Testfeldes durch die Nutzung der in Kapitel 5.1 beschriebenen Testfeld-Komponenten und folgt in der Umsetzung der beschriebenen Testfeld-Architektur.

Im Rahmen dieser Forschungsarbeit sind vier Komponenten zur Umsetzung der Testfeld-Architektur als Implementierungsbeitrag entstanden: Die Anpassung der vorherrschenden Kommunikationsinfrastruktur von LABSKAUS basierend auf einer nachrichtenorientierten Middleware mit einem Publish-Subscribe-Kommunikationsmuster und einheitlichem Datenmodell gemäß S-100 (Kapitel 4.3.1, 4.3.2 und 4.3.3), die polymorphe Schnittstelle zur dynamischen Integration beliebiger zu testender Systeme (Kapitel 5.4) und der Simulations-Adapter (Kapitel 5.5).

### 5.1 Testfeld-Komponenten

LABSKAUS besitzt Testfeld-Komponenten, deren Entwicklung im Rahmen von Forschungsprojekten und Promotionen erfolgte. Mit der Hilfe der Testfeld-Komponenten ist die Bereitstellung der notwendigen operativen Umgebung für zu testende Systeme möglich. Diese Testfeld-Komponenten werden im weiteren Verlauf der Arbeit gemäß der in Kapitel 4 beschriebenen Testfeld-Architektur in die Kommunikationsinfrastruktur integriert. Es ist zu beachten, dass diese Testfeld-Komponenten keinen Anspruch auf Vollständigkeit besitzen und je nach Testszenario ergänzende Testfeld-Komponenten benötigt werden. Aus diesem Grund berücksichtigt die Gestaltung der Testfeld-Architektur eine lose Kopplung, Konfigurierbarkeit und Erweiterbarkeit. Da die Implementierung der Testfeld-Komponenten kein Bestandteil dieser Arbeit ist und auf existierende Realisierungen zurückgegriffen wird, befinden sich in dieser Arbeit keine Implementierungsdetails. Für weitere Informationen wird auf die folgenden Publikationen verwiesen [Hahn15], [BrHa17].

**Stationäre und mobile NaviBox.** Die Testfeld-Komponente zur Integration von Sensoren und Aktuatoren in die Infrastruktur von LABSKAUS ist eine Sensorbox und trägt die Bezeichnung NaviBox. Die NaviBox ist ein kompakter Sensordaten-Hub, der Navigationsdaten an Bord eines Schiffes sowie Daten für landseitige maritime Überwachungssysteme zur Verfügung stellt. Die NaviBox bietet eine auf Ethernet, WLAN und LTE basierende Kommunikationsverbindung. Der minimale Aufbau besteht aus einem Radarsystem, einer Antenne des AIS und einem Windsensor. Sämtliche Sensoren auf der Basis von NMEA 0183, NMEA 2000 und Ethernet können ohne weitere Anpassungen angeschlossen werden. Im Falle eines abweichenden Kommunikationsstandards wird ein Adapter implementiert, um die Komponente in die NaviBox zu integrieren. Die NaviBox überführt sämtliche Daten durch einen statisch implementierten Adapter in das einheitliche Datenmodell gemäß S-100, wie näher in Kapitel 4.3.3 beschrieben. Weitergehend ist die NaviBox die Schnittstelle zur Teilnahme an der Kommunikation der nachrichtenorientierten Middleware mit Publish-Subscribe-Kommunikationsform.

**Referenzwasserstraße.** Die Referenzwasserstraße umfasst die deutsche Bucht, Elbe um Cuxhaven und den Nord-Ostsee-Kanal in der Nähe von Brunsbüttel. Eine stationäre maritime Überwachungsinfrastruktur bestehend aus den zuvor vorgestellten NaviBoxen und Breitbandkommunikation via LTE bietet eine Lagebilderfassung basierend auf AIS, Radar, Umweltinformationen und Videokameradaten. Die Referenzwasserstraße dient als experimentelle Plattform und für die Demonstration und Erprobung neuer maritimer Technologien durch ein beobachtbares Seegebiet ähnlich der Referenzstrecken der in Kapitel 3.2 beschriebenen Ansätze aus dem Automobilbereich. Neben dem Erproben von automatisierten Schiffsführungssystemen in einer kontrollierbaren Umgebung werden ebenfalls Übertragungen zwischen kooperierenden Schiffsassistenzsystemen durch die LTE Infrastruktur der Referenzwasserstraße ermöglicht. Die Referenzwasserstraße ist individuell durch stationäre NaviBoxen erweiterbar. Die Hauptaufgabe der Referenzwasserstraße ist einen komplett von Sensorik erfassten Seebereich für die landseitige und seeseitige Beobachtung zu installieren. Drei der stationären NaviBoxen sind in der Referenzwasserstraße auf den Dächern von Gebäuden in der Nähe der Küste stationiert.

**Forschungsboot ZUSE.** Seeseitig beinhaltet das Testfeld LABSKAUS das Forschungsboot ZUSE als Testträger welches modifiziert wurde, um als ein adaptiver Testträger für die V+V gemäß den in Kapitel 4.2 beschriebenen Komponenten eines automatisierten Schiffsführungssystemen zu dienen. Das für die Hochsee geeignete Forschungsboot ZUSE ist mit Sensoren und Aktorik Schnittstellen für Ruder und Motor ausgestattet, die über eine Software-schnittstelle bilateral nutzbar sind. Daneben existiert eine LTE-Schnittstelle, die von einem zu testenden System verwendet werden kann um Daten an andere seeseitige oder landseitige Systeme zur weiteren Kontrolle oder Verarbeitung zu senden. Ähnlich wie die Referenzwasserstraße ist eine mobile NaviBox auf dem Forschungsboot ZUSE installiert, welche sämtliche Sensordatenströme zusammenführt und auf die nachrichtenorientierte Middleware im einheitlichen Datenmodell gemäß S-100 sendet sowie Steuerbefehle hierüber entgegennimmt.

**Mobile Schiffsbrücke.** Für die Erprobung von Schiffsbrückenkomponenten im Labor und im Feld, wie beispielsweise Benutzerschnittstellen für die Navigation eines Schiffes, bietet LABSKAUS eine transportable und vielseitige mobile Schiffsbrücke. Drei Kästen beinhalten jeweils einen Computer und zwei Bildschirme mit Mehrfingergestenerkennung, die miteinander verbunden werden und als ein integriertes Navigationssystem dienen. In der Standardkonfiguration besitzt die mobile Schiffsbrücke ein proprietäres integriertes Navigationssystem sowie alternativ eine modifizierbare Variante als Open Source Software. Die Open Source Software in der Funktion einer ECDIS, trägt die Bezeichnung e-Navigation Prototype Display (EPD) in der schiffseitigen Ausprägung. Dies ergänzt eine Schiffsführungsanzeige, welche die Steuerung des Schiffes beispielsweise durch das Setzen einer Ruderlage ermöglicht und eine auf Schiffen übliche Radaranzeige. Die mobile Schiffsbrücke ist mit den weiteren Komponenten von eMIR verbunden, welche die notwendigen Daten und Kommunikationstechnologien der Testfeld-Infrastruktur für den Betrieb bereitstellen. Eine Gestaltungsentscheidung ist die Fähigkeit zum getrennten Betrieb der mobilen Schiffsbrückenelemente, um in begrenzten Räumlichkeiten betrieben zu werden, wie beispielsweise auf einem Schiff. Weitergehend ermöglicht die mobile Schiffsbrücke die Einrichtung von Experimenten, ohne auf die Installation eines Schiffes einzugreifen. Dies liegt darin begründet, dass die Daten entweder von einer NaviBox oder einem virtuellen Testfeld empfangen werden. Dies ermöglicht insbesondere die nahtlose Systementwicklung unter simulativen Bedingungen, bis zu einem späteren Test in der Realwelt in der Referenzwasserstraße beziehungsweise in Kombination mit der mobilen NaviBox an Bord des Forschungsbootes ZUSE.

**Maritime Kontrollstation.** Ein von einem Industrieunternehmen implementiertes VTS bereichert die Beobachtungsfunktionalität des physikalischen Testfeldes. Neben der Industrielösung existiert im physikalischen Testfeld eine weitere maritime Kontrollstation mit VTS-ähnlicher und ECDIS Funktionalität, die quelloffen verwendet und erweitert wird. Diese Variante basiert auf der küstenseitigen Variante der EPD, ähnlich wie die mobile Brücke. Die maritime Kontrollstation integriert Informationen über die Umgebung, beispielsweise auf der Basis der Datenquellen des DGPS, Radarsystems, SONAR, der Seekarten oder elektronische Navigationskarten und weitere schiffsspezifische Informationen. Für die Lagebilderzeugung nutzt die maritime Kontrollstation verschiedene durch die mobilen und stationären NaviBoxen bereitgestellte Daten, wie Radar, UKW Funk, AIS und Videokontrolle.

**Container-basierte Überwachungs- und Kontrollplattform.** Mit den Technologien der NaviBoxen und mobilen Schiffsbrücke ausgestattete mobile Container dienen der küstenseitigen und schiffseitigen Überwachung und Kontrolle. Diese Container entsprechen den Sicherheitsstandards des internationalen Übereinkommens über sichere Container (engl. „Container Safety Convention“). Die Container-basierte Plattform ist in die Kommunikationsinfrastruktur von LABSKAUS integriert und unterstützt in Kombination mit den weiteren Testfeld-Komponenten von LABSKAUS die nahtlose Systementwicklung unter Labor- und Realweltbedingungen.

## 5.2 Nachrichtenorientierte Middleware mit RabbitMQ

Die Umsetzung der Testfeld-Architektur durch LABSKAUS erfolgt unter dem Paradigma der losen Kopplung und Erweiterbarkeit und realisiert dies mit der Open Source Middleware RabbitMQ entsprechend des in Kapitel 4.3.2 beschriebenen Konzepts der Kommunikationsinfrastruktur. Hierbei knüpft diese Arbeit an eine existierende Implementierung in LABSKAUS an und erweitert diese um die in dieser Arbeit gestalteten Konzepte. RabbitMQ ist die technische Implementierung einer nachrichtenorientierten Middleware basierend auf dem Advanced Message Queuing Protocol (AMQP). AMQP ist ein offener Standard und binäres Netzwerkprotokoll auf der Anwendungsebene basierend auf einem Multi-Broker-Aufbau für eine nachrichtenorientierte Middleware. Ein oder mehrere zentrale oder dezentrale Broker verwalten als intermediäre Vermittler die Datenkanäle und vermitteln die Daten zwischen den Clients. Zudem organisieren die RabbitMQ Broker die Konfiguration der Middleware, damit die RabbitMQ Publisher und Subscriber die konfigurierte Middleware für den Datenaustausch verwenden können [ViWi12, S.1ff.]. Dieser Multi-Broker-Aufbau führt zu Skalierbarkeit und Flexibilität der Testfeld-Infrastruktur. RabbitMQ ist eine die folgenden Aspekte bietende Lösung.

- Der syntaktische beziehungsweise strukturelle und semantische Inhalt der Nachrichten ist für die Datenübertragung irrelevant. Dies fördert die Erweiterbarkeit und technologische Nachhaltigkeit der Testfeld-Infrastruktur, da eine Modifizierung des einheitlichen Datenmodells gemäß S-100 ohne eine Veränderung der Kommunikationsinfrastruktur möglich ist.
- Im Rahmen des CPS ist eine Gestaltungsentscheidung die Verwendung der losen Kopplung und Dezentralisierung. RabbitMQ unterstützt dies mittels der asynchronen Datenverarbeitung durch die Implementierung der Publish-Subscribe-Kommunikationsform. Die implementierten RabbitMQ Publisher beziehungsweise Subscriber senden und empfangen Daten über die RabbitMQ Broker unabhängig von einer Synchronisation untereinander, so dass beispielsweise der Ausfall einer sendenden Testfeld-Komponente nicht eine Blockade des Datenaustauschs innerhalb des physikalischen Testfeldes verursacht. In diesem Zusammenhang ist es ebenfalls möglich die Nachrichten an mehrere Datenempfänger zu verteilen, um mehrere Testszenarien gleichzeitig oder die Beobachtungsfähigkeit neben der Teststimulation zu unterstützen.
- RabbitMQ ist plattform- und programmiersprachenunabhängig, so dass eine lose Kopplung der Komponenten und die Zukunftsfähigkeit des Testfeldes unterstützt wird.

Gemäß der in Kapitel 4.3.5 beschriebenen physikalischen Gestaltung übernimmt ein zentraler RabbitMQ Broker als Intermediär die Verwaltungsaufgaben der Datenübertragung. Anhang D zeigt einen Bildschirmauszug der Weboberfläche des zentralen RabbitMQ Broker. Über diese Weboberfläche ist es möglich, die Kommunikationskanäle von LABSKAUS zu konfigurieren, überwachen und steuern. Ein produzierender RabbitMQ Publisher veröffentlicht die Nachrichten nicht unmittelbar an die Warteschlange (Queue) des RabbitMQ Broker, sondern sendet die Nachrichten zunächst an ein Vermittlungsregister (engl. „Exchange“).

Das Vermittlungsregister eines RabbitMQ Broker leitet die Nachrichten mit der Hilfe von Bindungen (engl. „Binding“) und Routingschlüsseln (engl. „Routing Keys“) zu den internen Zielwarteschlangen. Eine Bindung verknüpft eine Warteschlange und ein Vermittlungsregister. Ein Routingschlüssel stellt ein Nachrichtenattribut dar, das von einem RabbitMQ Publisher als Metadatum neben der Nachricht gesendet wird. Anhand dessen entscheidet das Vermittlungsregister des RabbitMQ Broker, auf welche der verknüpften Warteschlangen die Nachricht veröffentlicht wird.

RabbitMQ bietet die Möglichkeit den Empfang von Nachrichten zu bestätigen. Dies unterstützt das physikalische Testfeld durch die Flexibilität sowohl aktiv kommunizierende Komponenten mit hohen Anforderungen an die Zuverlässigkeit sowie Sensordatenströme ohne bestätigendes Kommunikationsprotokoll zu realisieren. RabbitMQ realisiert dies durch eine Unterscheidung des Zustands einer Nachricht in der Warteschlange als entweder bestätigt oder unbestätigt. Die Anzahl der unbestätigten Nachrichten in einer Warteschlange lässt sich durch einen Parameter begrenzen. Ebenso die Dauer der Datenvorhaltung, um die ausreichende Leistung im Rahmen der intensiven Datenstromverarbeitung zu gewährleisten.

### 5.3 Einheitliches S-100 Datenmodell mit EMF

Diese Arbeit greift auf eine existierende Umsetzung des einheitlichen Datenmodells gemäß S-100 von LABSKAUS zurück. Die Gestaltung des S-100 konformen Datenmodells erfolgt mit dem Modellierungswerkzeug Enterprise Architect. Mit der Hilfe von Enterprise Architect werden XSD Schemadateien erzeugt, deren Interpretation und Verarbeitung durch das Eclipse Modelling Framework (EMF) möglich ist. EMF ist ein quelloffenes Java-Framework, das automatisiert Quelltexte erzeugt. Das EMF generiert aus der XSD Schemadatei des S-100 Modells ein Ecore-Metamodell, welches die Datenobjekte in eine entsprechende Modellinstanz in Byte serialisiert überführt und die serialisierten Datenobjekte über RabbitMQ überträgt. Ecore ist das Metamodell des EMF und ermöglicht die Modellierung beliebiger Datenmodelle, wie jenes des Testfeldes LABSKAUS [Schä09].

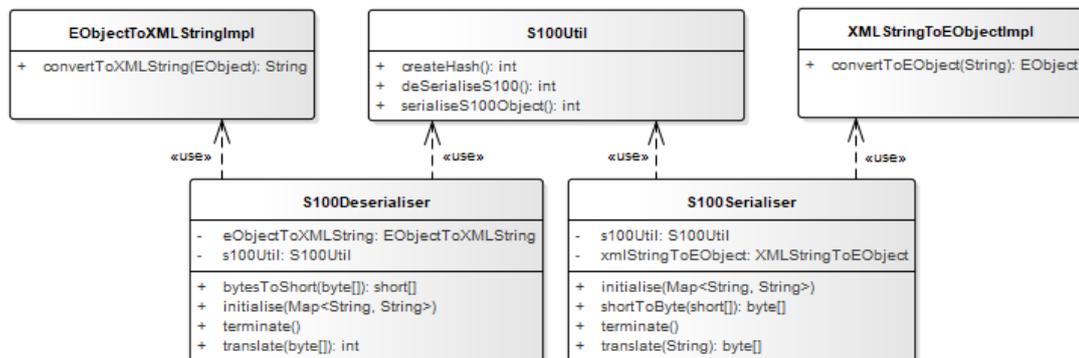


Abbildung 38: Serialisierung und Deserialisierung von S-100 Datenobjekten mit der Hilfe des EMF

Ein RabbitMQ Subscriber ist in der Lage den serialisierten Bytestrom zu empfangen und ebenfalls mit der Hilfe des Datenmodells gemäß S-100 eine Deserialisierung in eine

Ecore-Modellinstanz durchzuführen. Die Programmmodule für die Nutzung des einheitlichen Datenmodells gemäß S-100 liegen in der Programmbibliothek *eMIRLib* vor. Diese liefert Funktionen zum Serialisieren und Deserialisieren der gemäß dem modellierten Datenmodell übertragenen Datenobjekte. Abbildung 38 stellt den Zusammenhang der Klassen für die Nutzung des einheitlichen Datenmodells gemäß S-100 dar. Die polymorphe Schnittstelle basiert auf der Verwendung von XML für die notwendigen semantischen Transformationen. Daher erfolgt die Umwandlung einer Ecore-Modellinstanz eines S-100 Datenobjekts in XML durch die Funktionalität der EMF Programmbibliothek *XMLStringToObject* beziehungsweise *EObjectToXMLString* der Klassen *Resource* und *GenericXMLResourceFactoryImpl*. Eine Umwandlung in ein anderes Datenformat wie JSON ist alternativ durch entsprechende EMF Programmbibliotheken möglich.

Die Klasse *S100Util* der *eMIRLib* stellt die Funktionalität der Serialisierung und Deserialisierung gemäß der in eMIR implementierten Logik bereit. Hierbei erfolgt die Generierung einer Hashtabelle als Indexstruktur für die in Kapitel 4.3.1 beschriebenen S-100-Klassifizierer des Ecore-Modells. In diesem Zusammenhang findet das Ersetzen des konzeptionellen Datenmodells gemäß S-100 durch generierte Hashindizes als eindeutige Bezeichner statt, um eine effiziente Datenübertragung zu ermöglichen. Für die Übersetzung der über das Netzwerk übertragenen serialisierten Bytes einer S-100 Nachricht ist in den Klassen *S100Deserialiser* und *S100Serialiser* notwendig, durch den Aufruf der Methode *translate(final byte[] data)* die serialisierten Bytes in ein Array des Datentyps *short* zu überführen und umgekehrt. Dies begründet sich durch den Hashindex vom Datentyp *short*, um ausreichend viele Klassifizierer für die Elemente des S-100 Datenmodells abzubilden. Anschließend erfolgt im Falle einer Serialisierung der Aufruf *serializeS100Object(EObject obj)* durch Übergabe eines Datenobjekts vom Typ Ecore. Die Rückgabe der Methode ist ein Array des Datentyps *short*. Für das Deserialisieren findet umgekehrt die Übergabe des entsprechenden Arrays vom Datentyp *short* mit Bytedaten als Methodenaufruf an die Methode *deserializeS100(short[] serializedData)* statt, während das Resultat die Rückgabe eines Datenobjekts des Datentypen Ecore ist.

## 5.4 Polymorphe Schnittstelle

Die Implementierung der polymorphen Schnittstelle nutzt die Programmiersprache Java und die Auszeichnungssprache XML sowie die Programmiersprache XSLT für die Transformation der in XML ausgezeichneten Datenobjekte, wie in Abbildung 39 dargestellt.



Abbildung 39: Technologien zur Umsetzung der polymorphen Schnittstelle

Die polymorphe Schnittstelle besteht aus dem Paket *server*, welches die Klassen der polymorphen Schnittstelle zur Ausführung der Datenstromtransformation durch die Filter bein-

haltet, mit deren Hilfe die Datenströme transformiert werden. Um weitere mit der polymorphen Schnittstelle kompatible Filter zu erstellen bietet das Paket *server-api* die zu implementierenden Java-Schnittstellen an. Das Paket *communicationFilter* umfasst die implementierten Kommunikations-Handler, während das Paket *syntaxFilter* die Syntax-Handler beinhaltet. Das Paket *schemas* beinhaltet die in XSD vorliegenden Modellschemata. Das Paket *semanticLogic* bedient sich dieser Modellschemata, um sie mittels XSLT als plattform- und programmiersprachenunabhängige Technologie zu transformieren.

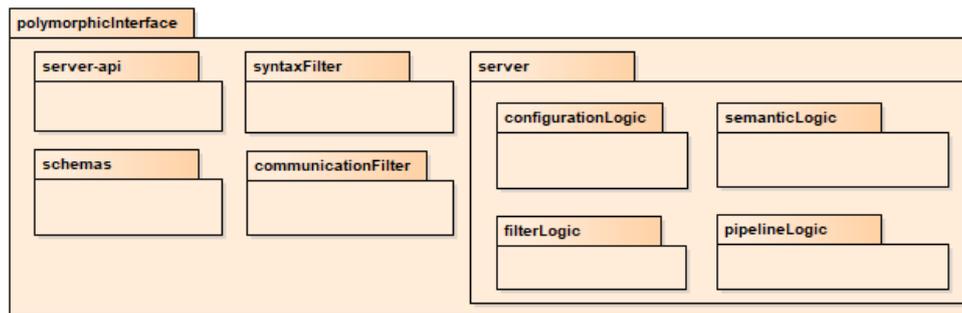


Abbildung 40: Paketdiagramm des Aufbaus der polymorphen Schnittstelle in LABSKAUS

**Laufzeitumgebung.** Die Laufzeitumgebung der polymorphen Schnittstelle befindet sich in den Unterpaket des Paketes *server* und ist in Java implementiert. Die Klassen des Paketes *server.filterLogic* dienen für das Laden und Initialisieren der Filter. Deren Konfiguration und Anordnung erfolgen gemäß der in XML ausgezeichneten Konfiguration einer Pipeline durch die Logik des Paketes *server.configurationLogic*. Das Paket *server.pipelineLogic* beinhaltet insbesondere die Logik zum Initialisieren einer Pipeline und den zur Kontrollierung des Datenflusses notwendigen Pipes, welche die Filter verbinden.

Die Software-Implementierung einer Filterkomponente liegt einer Instanz der polymorphen Schnittstelle als Java-Archiv (JAR) vor, das von der Laufzeitumgebung der polymorphen Schnittstelle eingebunden und ausgeführt wird. Das dynamische Einbinden der Objektinstanzen zur Laufzeit wird als Abhängigkeitsinjektion (engl. „Dependency Injection“) bezeichnet und erfolgt gemäß dem entsprechenden Softwareentwurfsmuster mit der Hilfe des Google Guice Frameworks. Google Guice nutzt Module als Beschreibung der zur Laufzeit zu ladenden Klassen, dessen Logik ein implementierter Filter erben muss. Durch die Verwendung des Softwareentwurfsmusters der Abhängigkeitsinjektion ist eine Form der polymorphen Schnittstelle möglich, in der die Abhängigkeiten zu einer konkreten Implementierung durch den Mechanismus der Coercion im Rahmen der Polymorphie zur Laufzeit aufgelöst wird, wie näher in Kapitel 4.5 erörtert [Vanb08, S.5ff.].

Wie in Abbildung 41 dargestellt existiert die Klasse *FilterJarFileImpl* im Paket *server.configurationLogic* für eine zu ladende Implementierung eines Filters in Form einer JAR. Die JAR-Datei enthält als notwendige Angaben für einen Filter den Pfad der JAR-Datei und den Klassennamen des Google Guice Moduls. Diese Angaben erfolgen über die Konfigurationsdatei der polymorphen Schnittstelle in XML. Eine Instanz der Klasse *FilterJarFileObjectFactoryImpl* besitzt Zugriff auf die Konfiguration der polymorphen Schnittstelle. Für jede konfigurierte Filterkomponente erfolgt der

Methodenaufruf *initModule(filterJarFile : FilterJarFile)*. Mit der Hilfe der Klasse *JarFileClassLoader* und des Methodenaufrufs *addJarFile(path : Path)* findet das Laden aller Filter der JAR-Dateien und das Durchsuchen dieser nach den konfigurierten Modulen von Google Guice statt. Diese Funktionalität ist in der Klasse *FilterJarFileObjectFactoryImpl* implementiert. Die identifizierten Module fasst die *Injector* Klasse als Bindung zusammen, die Bestandteil des Google Guice Frameworks ist. Ein *Injector* ermöglicht das Instanzieren von Objekten der Module. Abschließend erfolgt gemäß dem Softwareentwurfsmuster der Fabrikmethode der Methodenaufruf *getFilterInstance(className : String)* der *FilterJarFileObjectFactory* mit dem Klassennamen des zu ladenden Filters. Dieser Methodenaufruf liefert als Rückgabewert die jeweilige Klasse des *Injector* aus den zuvor geladenen Modulen. Anschließend steht eine Instanz der Implementierung des jeweiligen Filters dynamisch zur Laufzeit zur Verfügung und kann durch die Laufzeitumgebung der polymorphen Schnittstelle ausgeführt werden.



Abbildung 41: Stark vereinfachte Klassen für das Laden von Filtern in der polymorphen Schnittstelle

Für die Durchführung der verschiedenen Tests erfolgte die Implementierung vieler konfigurierbarer Filterkomponenten in der Programmiersprache Java auf der Basis der *server-api* beziehungsweise der Programmierung von XSLT Regelsätzen für die semantischen Transformationen. Ein Filter als Kommunikations-Handler beziehungsweise Syntax-Handler implementiert die jeweiligen Schnittstellen einer eingehenden oder ausgehenden Komponente zuzüglich der jeweils notwendigen Transformationsoperatoren, wie in Abbildung 42 dargestellt. Die Schnittstellenklassen der *server-api* bieten hierbei die grundlegende Logik als Gerüst, um durch die Laufzeitumgebung der polymorphen Schnittstelle ausgeführt werden zu können.

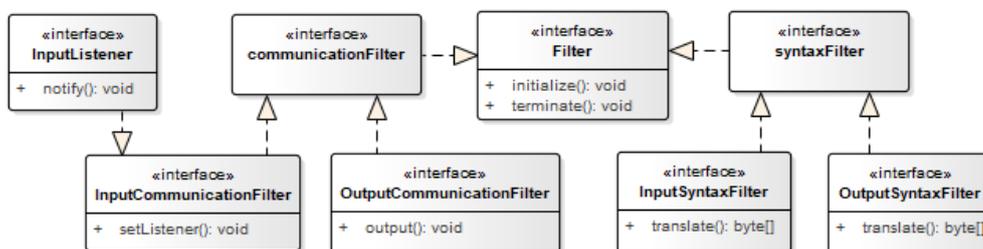


Abbildung 42: Implementierung von Filterkomponenten gemäß der API der polymorphen Schnittstelle

Wegen des begrenzten Rahmens dieser Arbeit erfolgt keine vollständige Beschreibung sämtlicher Implementierungsarbeiten der polymorphen Schnittstelle. Eine Übersicht der implementierten Filter befindet sich in Anhang E.

**Semantik-Handler.** Die Ausführung der semantischen Annotationen mit der Hilfe von XSD Schemadateien und Transformationsregelsätzen in XSLT erfolgt durch die Java-Klassen

des Pakets *polymorphic\_interface.server.semanticLogic*, die in Abbildung 43 dargestellt sind. Die hierin enthaltenen Klassen nutzen die Programmierschnittstelle von Java für die Prozessierung der XML (JAXP), um die implementierten XSLT Transformationsanweisungen auszuführen. Das *SemanticTransformationModule* stellt den von Google Guice injizierten Einstiegspunkt dar, der durch die Klasse *ModelTransformationImpl* die Transformationen der XSLT Regelsätze ausgeführt. Zunächst erfolgt die Überprüfung vor der Transformation durch die Klasse *ModelInstanceValidatorImpl* hinsichtlich der korrekten Entsprechung des Datenobjekts in XML gemäß der zu Grunde liegenden XSD Schemadatei.

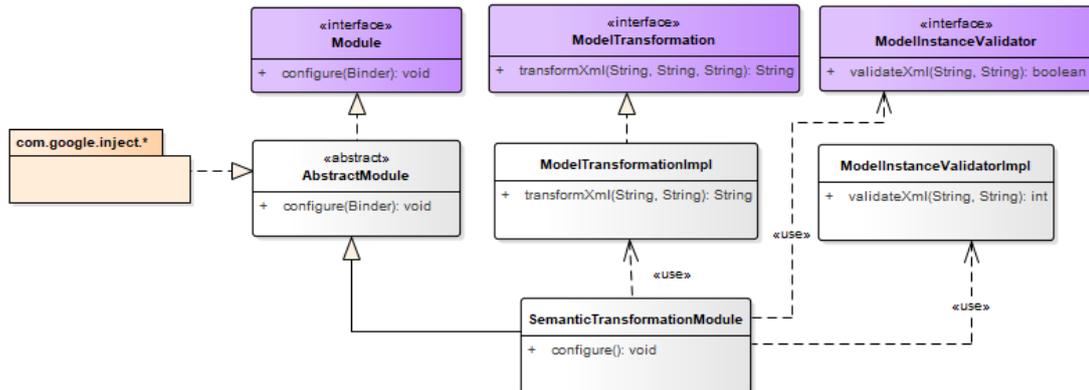


Abbildung 43: Klassendiagramm der semantischen Transformation durch die polymorphe Schnittstelle

**Konfiguration.** Die Konfiguration der Pipelines der polymorphen Schnittstelle erfolgt mit der Hilfe der in XML ausgezeichneten *configuration.xml*. Für die Konfiguration der Protokollierung der Transformationsausführung ist die *log4j2.xml* vorhanden. Die für die Konfiguration notwendigen Elemente sind gemäß der in Abbildung 44 dargestellten Baumstruktur zu deklarieren. In diesem Kontext erfolgt die Definition der Semantiken (*semantics*), in der die XSD Schemadateien mit dem Pfad und einer ID referenziert werden. Weitergehend beschreibt die Konfiguration die zu nutzenden Filter (*filter*) durch die Angabe zum einen der Ressourcen der JARs (*resources*) und zum anderen der Eigenschaften der zu instanzierenden Filter (*instances*). Hierbei sind je nach Filterimplementierung für die Kommunikations-Handler und Syntax-Handler beliebige Parameter durch eine Kombination von Schlüssel-Werte zu übergeben, wie Benutzername, Passwort und Host.

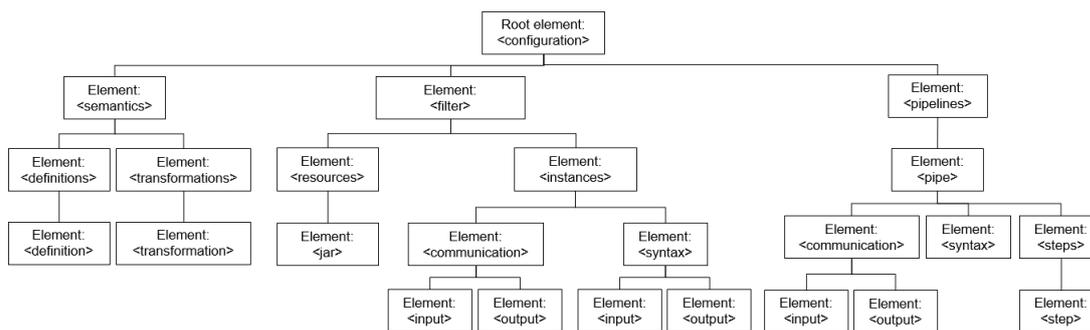


Abbildung 44: Aufbau der Konfiguration der polymorphen Schnittstelle

Es erfolgt die Unterscheidung zwischen eingehendem und ausgehendem Filter (*input* und *output*). Die Konfiguration beliebig vieler Pipelines erfolgt durch Bezeichnung der zu nutzenden Kommunikations-Handler (*communication*), Syntax-Handler (*syntax*) und den auszuführenden semantischen Transformationsschritten (*steps*). Ein Transformationsschritt beschreibt hierbei die unter der Definition der Semantiken deklarierten Modelle.

## 5.5 Simulations-Adapter

Für die gemischt virtuell-physikalische Testdurchführung zur Komplementierung der Realwelt benötigt das physikalische Testfeld die Integration von Simulatoren eines virtuellen Testfeldes, wie in Kapitel 4.4.2 beschrieben. Für diesen Zweck greift diese Arbeit auf das existierende virtuelle Testfeld HAGGIS von eMIR zurück, welches das Kapitel 4.4.2 beschreibt. Das folgende Kapitel erörtert die Implementierung des Simulations-Adapters für die Realisierung des bilateralen Datenaustausches zwischen den Kommunikationsinfrastrukturen von LABSKAUS und HAGGIS.

Analog zu dem Vorgehen des FMI ist die Implementierung einer Schnittstelle des virtuellen Testfeldes für den Austausch der Simulationsmodelldaten notwendig. Diese als „HAGGIS Protokollhandler“ bezeichnete Komponente ist in Java implementiert und durch das Paket *de.emir.service.codecs* in der Sensorsimulation des HAGGIS Entwicklungsprojektes lokalisiert. Das Datenmodell von HAGGIS basiert auf einem Datenmodell gemäß S-100, ähnlich wie das Datenmodell von LABSKAUS. Jedoch erfolgte wegen spezifischer Anforderungen von HAGGIS die Implementierung des Modellierungswerkzeugs uCore im Vorfeld dieser Arbeit. Das Modellierungswerkzeug uCore ähnelt stark dem Ecore des EMF, erweitert diese Technologie jedoch um die ergänzende Modellierungsfähigkeit der Beziehungen zwischen Entitäten. Dies ist im Rahmen der Simulationsumgebung HAGGIS notwendig und erfordert somit von einer Lösung die Übersetzung zwischen uCore und Ecore. Zwar ist HAGGIS in der Lage NMEA0183 konforme Datensätze und symbolische Radardaten zu erzeugen, jedoch basiert LABSKAUS auf XML. Der HAGGIS Protokollhandler überführt das intern durch uCore repräsentierte konzeptionelle Simulationsmodell in eine durch XML ausgedrückte Struktur und sendet die entstehenden Datenströme über das Kommunikationsprotokoll UDP an die Datensenke des Simulations-Adapters.

Für diese Funktionalität erfolgt die Implementierung des *UCoreXMLProtocolHandler*. Da der ausgehende Datenstrom des HAGGIS Simulationsmodells über eine erhebliche Größe verfügt und hierdurch nicht über UDP versendet werden kann, findet die Komprimierung des zu übertragenden Datenstroms durch die Programmbibliothek für Kompression *gzip* des Pakets *java.util.zip* statt. Anhang F illustriert die Konfiguration des Austausches der Modelldaten in HAGGIS, in diesem Fall den Export von Datenobjekten modelliert durch das uCore Datenmodell in XML über das UDP Kommunikationsprotokoll.

Der Simulations-Adapter übernimmt im Wesentlichen das Filtern der für das physikalische Testfeld unrelevanten Daten des aus HAGGIS extrahierten Datenobjekts, wie beispielsweise die Metadaten des Zustands der Simulation. Der Simulations-Adapter verfügt über die Klasse *UDPGZIPIInputAdapter* als eingehende Schnittstelle zu HAGGIS für den verbindungslosen Pakettransport über UDP, welche wiederum die Klasse *DatagramSocket* der *java.net* Programmbibliothek für den Datenempfang nutzt und diese mit der *gzip* Programmbibliothek dekomprimiert.

Auf der Ebene des Semantik-Handlers des Simulations-Adapters erfolgt zunächst die Überführung der gegebenenfalls nicht durch Ecore repräsentierbaren Artefakte des uCore Modells. Bei der Implementierung findet die Realisierung des Datenaustauschs für das statische und dynamische AIS und Radar statt. Diese semantische Interoperabilitätskollision ist verlustfrei zu überführen. Ähnlich wie in der polymorphen Schnittstelle in Kapitel 5.4 nutzt der Simulations-Adapter für die Transformation die XSLT und in der Klasse *Transformer* die Methoden *transformXML*, welche die übergebene XML Zeichenkette gemäß den ebenfalls in XSLT übergebenen Transformationsregeln zu einer Ziel-Zeichenkette in XML transformiert. Die Ziel-Zeichenkette verfügt anschließend über ein Ecore konformes S-100 Datenobjekt in XML. Ergänzend erfolgt das Hinzufügen eines Attributs *ReferenceID*, um zwischen simulierten und realen Daten bei der weiteren Verwendung durch das physikalische Testfeld LABSKAUS unterscheiden zu können. Auf der Ebene des Semantik-Handler ist der Filtermechanismus durch XSLT realisiert, um die nichtrelevanten Elemente des eingehenden Simulationsdatenobjekts aus HAGGIS zu entfernen. Ein Senden des gesamten Datenobjekts des Simulationsmodells in die nachrichtenorientierte Middleware des physikalischen Testfeldes LABSKAUS wäre wegen der Nachrichtengröße ineffizient und ist für die Testdurchführung nicht notwendig.

Um an der Kommunikation mit der nachrichtenorientierten Middleware des physikalischen Testfeldes teilzunehmen wird analog zu den Testfeld-Komponenten ein produzierender RabbitMQ Client implementiert. Hierfür wird die in Kapitel 5.4 vorgestellte Implementierung der Programmlogik des *RabbitMQOutputFilter* wiederverwendet und konfiguriert.

## 5.6 Zusammenfassung der Implementierung

Dieses Kapitel beschreibt die Realisierung der zuvor entwickelten physikalischen Testfeld-Architektur durch das physikalische Testfeld LABSKAUS. LABSKAUS unterstützt das Testen der aus dem Drei-Schichten-Modell eines automatisierten Schiffsführungssystems

abgeleiteten Funktionalität durch die Bereitstellung der notwendigen Testfeldfunktionen unter Berücksichtigung der etablierten Methoden der V+V.

Für die Bereitstellung der operativen Umgebung dienen Testfeld-Komponenten wie die Referenzwasserstraße oder das Forschungsboot ZUSE als Testträger. Diese produzieren Daten und empfangen sie über die nachrichtenorientierte Middleware, welche durch RabbitMQ implementiert wurde. Hierfür erfolgt die Umsetzung durch eine Publish-Subscribe-Kommunikationsform. Sämtliche an der Kommunikation teilnehmenden Komponenten zur Funktionsausführung des zu testenden Systems und die Werkzeuge der V+V nutzen statische Adapter. Diese statischen Adapter integrieren die Komponenten durch definierte Transformationsoperatoren, welche die verschiedenartigen Schnittstellen der Komponenten in das einheitliche Datenmodell gemäß S-100 überführen und Daten entweder veröffentlichen oder abonnieren. Die besondere Herausforderung der dynamischen Integration der zu testenden Systeme begegnet die polymorphe Schnittstelle, welche mit der Hilfe einer konfigurierbaren Anordnung verschiedener Transformationsoperatoren durch das Softwarearchitekturmuster der Abhängigkeitsinjektion eine wiederverwendbare Lösung darstellt. Der Simulations-Adapter integriert LABSKAUS und das virtuelle Testfeld HAGGIS, um insbesondere Testszenarien im Closed-Loop-Verfahren umzusetzen. Hierdurch ist die Anreicherung eines Realweltszenarios durch ergänzende virtuelle Daten möglich.

## 6 Evaluation der entwickelten physikalischen Testfeld-Architektur

Die vorherigen Kapitel stellen den Lösungsansatz einer Testfeld-Architektur und deren Implementierung durch das physikalische Testfeld LABSKAUS vor. Mit deren Hilfe geschieht die Beantwortung der Forschungsfrage „Wie muss ein physikalisches Testfeld gestaltet sein, um die Verifikation und Validierung in der Systementwicklung von automatisierten Schiffsführungssystemen effizient zu unterstützen?“. In dieser Arbeit erfolgt die begründete Überführung der aus den theoretischen Grundlagen und dem Stand der Technik erhobenen Anforderungen in die Gestaltung der Testfeld-Architektur. LABSKAUS setzt diese Anforderungen an ein physikalisches Testfeld um und dient in diesem Kapitel für die Evaluation der Anwendbarkeit der Testfeld-Architektur im Rahmen verschiedener Einsatzszenarien. Die Evaluation überprüft zwei Hypothesen, deren Verifikation oder Falsifikation die Fähigkeit des Lösungsansatzes zur Erreichung der in Kapitel 1.3 definierten Zielsetzung zeigt oder Lücken aufdeckt. Die folgenden Hypothesen dienen dieser Evaluation.

1. Die Testfeld-Architektur ist für die Verifikation und Validierung von heterogenen automatisierten Schiffsführungssystemen geeignet.
2. Die Testfeld-Architektur realisiert die bisher von maritimen Testaufbauten überwiegend unzureichend erfüllten Anforderungen.

Die Anwendbarkeitsüberprüfung der technologischen Artefakte anhand der relevanten Einsatzszenarien ermöglicht die Beurteilung der Validität der Systemarchitektur [RoVe08]. Für die Untersuchung der Eignung der Testfeld-Architektur für die Umsetzung unterschiedlicher Methoden der V+V für heterogene automatisierte Schiffsführungssysteme finden repräsentative Tests mit LABSKAUS statt. Diese müssen zum einen eine Testumgebung für die verschiedenen Komponenten des in Kapitel 4.2 vorgestellten Drei-Schichten-Modells zum Führen eines Schiffes bieten. Zum anderen ist die Umsetzung unterschiedlicher Methoden der V+V gemäß den Anforderungen der Systementwicklungsmethodik durch LABSKAUS notwendig. Die folgenden Kapitel untersuchen die Anwendbarkeit der Testfeld-Architektur durch die Implementierung und Nutzung entlang des Systementwicklungsprozesses durch die in Kapitel 2.2.5 identifizierten Einsatzszenarien eines Testfeldes. Diese sind:

- Die explorative Datenerhebung für die frühe modellbasierte Systementwicklung und Unterstützung des Front-Loading der V+V. Dies geschieht durch den empirischen Aufbau einer Szenarien-Datenbank für kritische Schiffsbegegnungen, denen das physikalische Testfeld die entsprechenden Beobachtungsdaten zur Verfügung stellt.
- Der Test im Open-Loop-Verfahren für einen szenariobasierten Systemtest beziehungsweise die Konformitätsprüfung und die dynamische Integration des zu testenden Systems. Das zu testende System ist ein Teilautomatisierungssystem für die Kollisionsvermeidung, das eine proprietäre Schnittstelle besitzt.

- Der Test im Closed-Loop-Verfahren während der Systementwicklung und die Komplementierung der IT-Infrastruktur von LABSKAUS durch einen integriert virtuell-physikalischen Testfeld-Aufbau und die Integration des zu testenden Systems. Das zu testende System ist ein landseitig und seeseitig verortetes hochautomatisiertes Schiffsführungssystem, das eine dem NMEA 2000 Standard entsprechende Schnittstelle bereitstellt.

Die zweite Hypothese verifiziert die Zielerfüllung durch die Überprüfung der Anforderungserfüllung und die Bewertung der durch LABSKAUS realisierten Testfeld-Architektur, wie Abbildung 45 darstellt. Die Argumentation der Verifikation der Systemarchitektur erfolgt auf der Basis der zuvor beschriebenen Anwendungen von LABSKAUS anhand der dargelegten Einsatzszenarien [Wass05, S.697ff.].

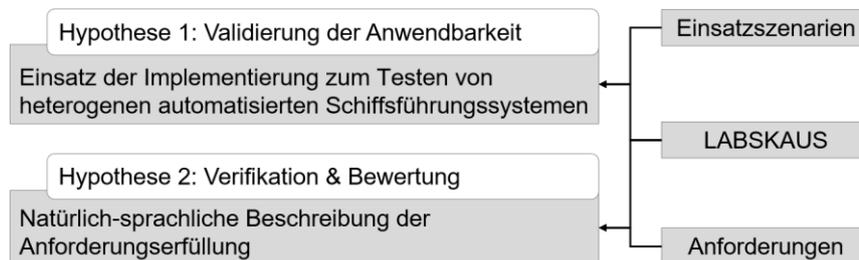


Abbildung 45: Vorgehen für die Evaluation der Testfeld-Architektur

Um die Eignung der Testfeld-Architektur für die vorgestellten Einsatzszenarien zu überprüfen erfolgt zunächst in Kapitel 6.1 die umfangreiche Leistungsdatenerhebung der implementierten Testfeld-Infrastruktur hinsichtlich auftretender Latenzen und Datenübertragungsraten.

## 6.1 Leistungsdatenerhebung von LABSKAUS

Das physikalische Testfeld LABSKAUS besteht aus integrierten Komponenten, die geographisch verteilt über teilweise drahtlose Kommunikationstechnologie kommunizieren. Hierbei erfolgen diverse Datenverarbeitungsschritte, die additiv entgegen des operativen Betriebs die Leistung beeinflussen. Um die operative Zuverlässigkeit des physikalischen Testfeldes und dessen Grenzen zu beurteilen, erfolgt in diesem Kapitel die Leistungsdatenerhebung von LABSKAUS unter den Aspekten der praktischen Anwendbarkeit. Daher stellt dieses Kapitel die Messung der relevanten Leistungsmerkmale der Latenz und Datenübertragungsraten der Infrastruktur-Komponenten von LABSKAUS vor.

Abbildung 46 illustriert zusammenfassend die einzelnen Komponenten von LABSKAUS, die einen Einfluss auf die Datenstromverarbeitung nehmen. Hierbei stellen unterschiedliche Systemkomponenten die operative Testumgebung für ein zu testendes System bereit, wie Sensorschnittstellen, Kommunikationstechnologie, Datenstromverarbeitungssoftware oder die ausführenden Recheneinheiten. Die Latenz bezeichnet den Zeitraum der Datenstromverarbeitung von der Datenquelle bis zur Datensenke, während die Datenübertragungsrate die über ein Kommunikationsmedium übertragene Datenmenge pro Zeit angibt.

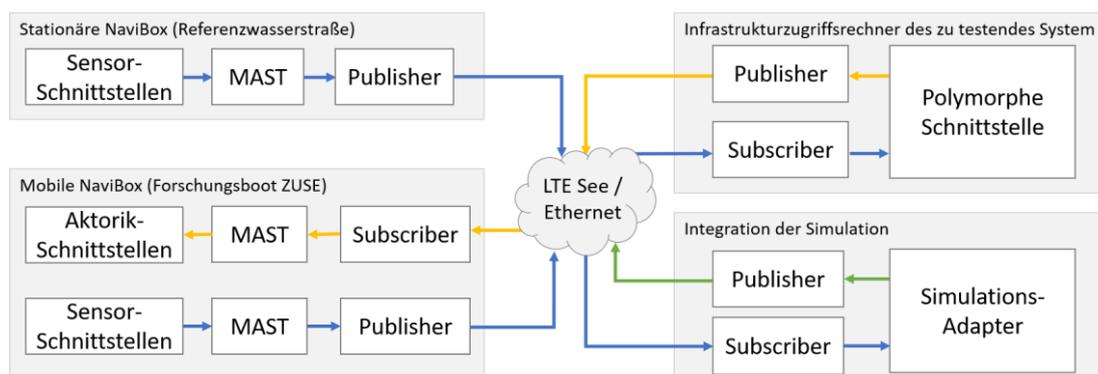


Abbildung 46: Datenströme für die Messung der Latenz und des Datendurchsatzes der Infrastruktur von LABSKAUS

Wie in Abbildung 46 dargestellt, lassen sich die zu messenden Datenströme aus den Testfeld-Komponenten und der Testfeld-Infrastruktur zusammen auf der Basis der in Kapitel 4.3.5 vorgestellten physikalischen Verteilung der Testfeld-Architektur identifizieren. MAST (engl. „Maritime Automatic Sensor Distribution Tool“) bezeichnet die auf den stationären und mobilen NaviBoxen installierte eigenentwickelte Software, welche neben der Bereitstellung der Schnittstellentreiber der Sensorik und Aktorik sämtliche Datenverarbeitungsschritte ausführt. Weitergehend findet die statische Transformation in das einheitliche Datenmodell gemäß S-100 und die Kommunikation mit der nachrichtenorientierten Middleware RabbitMQ von LABSKAUS statt.

Der gelbe Datenstrom realisiert die Steuerbefehle eines zu testenden Systems, die zunächst durch die polymorphe Schnittstelle in das einheitliche Datenmodell gemäß S-100 transformiert und anschließend über die Kommunikationsinfrastruktur gesandt werden. Die Datensenke ist die Schnittstelle zur Ausführung durch die Aktorik in der mobilen NaviBox an Bord des Forschungsbootes ZUSE. Die blau dargestellten Datenströme extrahieren Sensordaten zur Wahrnehmung der Umwelt und Eigenschiffsinformationen. Der grüne Datenstrom stellt die simulative Erzeugung und Einspeisung von Daten dar. Die Leistungsfähigkeit dieser Datenstromverarbeitung messen die folgenden Kapitel quantitativ.

### 6.1.1 Latenz

Ein Rechner mit einem Intel Core i7 7020HQ Prozessor und 16 GB PC3-17000 DDR3 Arbeitsspeicher misst die Latenz der Komponenten von LABSKAUS. Für die Messung der Latenz der verschiedenen datenstromverarbeitenden Komponenten sind Zeitstempel an die Datennachrichten gesetzt, mit deren Hilfe im Anschluss die Auswertung der Differenz stattfindet. Die Messungen der Latenz findet durch die Ermittlung des Maximalwerts, Median und Durchschnittes statt, wie in Tabelle 4 dargestellt.

Messobjekt	Stichprobengröße	Latenz in Millisekunden (ms)		
		Max	Median	Schnitt
Schiffseitige Sensordatenextraktion	10.857	532	38	56,14
Schiffseitiger Befehlsempfang	9.381	26	12	11,58
RabbitMQ lokal	5.542	4	2	2
See-landseitige Datenübertragung	31.455	99.151	153	10.615
See-landseitige Datenübertragung (ohne Randfälle)	18.786	1.000	73	107,63
Polymorphe Schnittstelle	14.539	22	2	2,9
	44.611	50	2	2,5
Simulations-Adapter	14.508	196	21	22,8

Tabelle 4: Leistungsdaterhebung hinsichtlich der auftretenden Latenz von LABSKAUS

**Polymorphe Schnittstelle.** Die Latenz der polymorphen Schnittstelle bestimmt die Zeitspanne der Datenverarbeitung zwischen dem Eingang der Daten sowie der anschließenden Transformation und dem ausgehenden Übertragungszeitpunkt an die Datensinke. Daher erfolgt der Messaufbau für die Latenz der polymorphen Schnittstelle durch einen Zeitstempel bei dem Eingang einer Datennachricht in dem eingehenden Kommunikations-Filter und einem Zeitstempel nach der erfolgten Transformation in dem ausgehenden Kommunikations-Filter. Durch einen an jede eingehende Datennachricht angehängten Identifikator ist die deren eindeutige Identifikation möglich. Die Identifikation der durchschnittlichen Nachrichtenanzahl pro Sekunde findet durch die Auswertung von 563.347 Datennachrichten statt, welche die NaviBoxen über einen Zeitraum von 3.686 Sekunden beziehungsweise circa 61 Minuten produziert. Im Durchschnitt erzeugt die NaviBox alle sechs Millisekunden eine Datennachricht. Zwei Testszenarien werden durchgeführt, in denen 14.539 Datennachrichten durch eine mobile NaviBox an Bord des Forschungsbootes ZUSE als Datenquelle und 44.611 Datennachrichten durch drei mobile NaviBoxen erzeugt werden. Die Größe der durch die NaviBox erzeugten Datennachrichten liegt bei durchschnittlich 215,9 Byte. Die mobile NaviBox besitzt die folgenden Datenquellen: Motorsteuereinheit, Ruderwinkelindikator, Triducer, DGPS, AIS, IMU und Radarsystem.

Die Messung ergibt wie in Abbildung 47 dargestellt, dass bei dem Start der polymorphen Schnittstelle die ersten 500 eingehenden Nachrichten eine höhere Latenz aufweisen als sämtliche folgenden. Dies führt zu dem dargestellten Ausschlag der Transformationsverarbeitung, die maximal bis zu vier Sekunden Latenz verursacht. Die Interpretation dieser beobachteten Korrelation begründet sich durch die gewählte Programmiersprache Java, denn die ausführende Java Laufzeitumgebung mit der Bezeichnung *Hotspot* erkennt dynamisch zur Laufzeit häufig genutzte Programmabschnitte und optimiert diese durch eine Just-in-time-Kompilierung und adaptive Optimierung [PaVC00]. Sofern die ersten 200 Datennachrichten bei dem Start der polymorphen Schnittstelle aus der Messung entfernt werden, tritt während der gesamten Laufzeit der Transformation in 99% der Fälle eine maximale Latenz von zehn Millisekunden auf. Die durchschnittliche Latenz der polymorphen Schnittstelle beträgt 2,9 Millisekunden. Hierbei ist zu erkennen, dass diese Latenz nicht mit der Anzahl der produzierenden Sensorsystemknoten korreliert. Die Begründung liegt in den parallelisierten Ausführungssträngen (engl. „Thread“) in der Implementierung der polymorphen Schnittstelle.

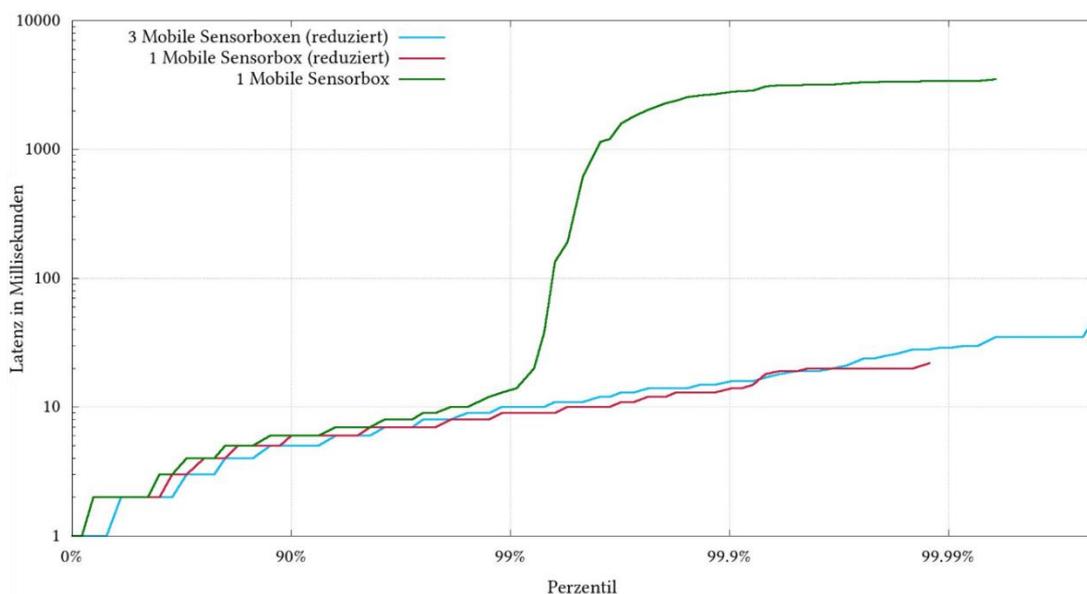


Abbildung 47: Latenzen der polymorphen Schnittstelle mit einer beziehungsweise drei schiffseitigen NaviBoxen als Dateneingabe. Die reduzierten Eingaben stellen die polymorphe Schnittstelle während des laufenden Betriebs exklusive der anfänglichen Latenz dar, welche durch die Laufzeitumgebung *Hotspot* von Java verursacht wird.

Da die polymorphe Schnittstelle einen XSLT Regelsatz für die Transformation verwendet, beeinflusst die Datennachrichtengröße die Verarbeitungszeit und somit Latenz, wie in Anhang G dargestellt. Es ist eine Korrelation zwischen der Datennachrichtengröße sowie der Verarbeitungszeit durch den XSLT Prozessor zu erkennen. Die Stichprobengröße beträgt 14.859 Datennachrichten. Die kleinen Datennachrichten mit einer Größe von kleiner als 78 Byte und einem Anteil von 11% an der Gesamtmenge erzeugen im Durchschnitt eine Latenz von 2,43 Millisekunden, einen Median von 2 Millisekunden und einen Maximalwert von 16 Millisekunden. Die großen Datennachrichten mit einer Größe von 288 Byte und einer Gesamtmenge von 68% resultieren in einer durchschnittlichen Latenz von 4,39 Millisekunden, einem Median von 3 Millisekunden und einem Maximalwert von 21 Millisekunden. Es ist somit eine durchschnittlich höhere Latenz von 1,96 Millisekunden bei der Verarbeitung von großen Datennachrichten zu messen.

**RabbitMQ.** Unabhängig von der zur Verfügung stehenden Datenübertragungsrate der genutzten Kommunikationstechnologie erfolgt zunächst die Messung der Latenz der genutzten Datenübertragungstechnologie RabbitMQ. Der Testaufbau besteht aus zwei lokal als JAR ausgeführte Programme, die jeweils einen veröffentlichenden RabbitMQ Client und abonnierenden RabbitMQ Client implementieren. Es werden wie zuvor die Datennachrichten der mobilen NaviBox mit einem Intervall zwischen zwei Datennachrichten von sechs Millisekunden übertragen. Die Latenz berechnet die Differenz des zu sendenden Zeitstempels und dem Zeitstempel nach dem Empfang durch den RabbitMQ Client. Das Ergebnis dieser Messung ohne Berücksichtigung einer Kommunikationstechnologie wie LTE, WLAN oder Funk ist die auftretende Latenz der Nachrichtenübertragung mittels RabbitMQ im Bereich von unter vier Millisekunden. Im Durchschnitt benötigt eine Datennachricht zwischen einem

veröffentlichenden RabbitMQ Client und einem abonnierenden RabbitMQ Client zwei Millisekunden.

**Sensordatenextraktion und Befehlsempfang.** Die in Abbildung 48 dargestellte Latenz der Sensordatenextraktion bezeichnet die zeitliche Differenz zwischen der Erfassung eines Sensordatums durch eine proprietäre Komponente und der Bereitstellung von diesem über die IT-Infrastruktur des physikalischen Testfelds an die Datensenke, beispielsweise ein zu testendes System oder Werkzeug für die V+V. Umgekehrt stellt die Latenz des Befehlsempfangs die Zeitspanne zwischen der ausgehenden Schnittstelle eines zu testenden Systems und der Übermittlung der Befehlsdatenobjekte an die Aktorik dar. Daher nutzt der Messaufbau für die Feststellung der Latenz dieser Datenströme einen Zeitstempel für die Sensordatenextraktion vor der statischen Transformation an der eingehenden Schnittstelle in MAST und einen Zeitstempel in dem ausgehenden Kommunikations-Handler der polymorphen Schnittstelle. Analog befindet sich für die Messung des Befehlsempfangs ein Zeitstempel bei dem eingehenden Kommunikations-Handler der polymorphen Schnittstelle und vor dem Datenversand über eine UDP Kommunikationsendpunkt von MAST als Übergabepunkt für die Ausführung durch die Aktorik.

Die etwa achtzehn minütige Messung der schiffseitigen Sensordatenextraktion während der Fahrt mit dem Testträger und einer Stichprobengröße von 10.857 ergibt einen Mittelwert für die Latenz der schiffseitigen Sensordatenbereitstellung über Ethernet von 38 Millisekunden und für maximale Randfälle von 532 Millisekunden, wie in Abbildung 48 dargestellt. Die hohen Latenzen als Randfälle der Messung treten wegen des auf der mobilen NaviBox installierten Betriebssystems Windows 10 und der fehlenden Echtzeitfähigkeit auf. Hierdurch resultiert die Priorisierung der Aufgaben des Betriebssystems auf gleicher Ebene mit denen der Sensordatenextraktion. Dennoch ist in 99,9% der Fälle eine Latenz von weniger als 1.000 Millisekunden messbar, wie in Anhang H dargestellt.

Sofern eine Datenübertragung zwischen dem landseitigen Server der zentralen nachrichtenorientierten Middleware und dem Testträger Schiff auf See über LTE stattfindet, tritt ein unähnliches Verhalten auf. Die Synchronisierung der Zeiten in dem verteilten Systemverbund mit paketbasiertem Kommunikationsnetz realisiert der Network Time Protocol (NTP) Standard. Der Mittelwert der Latenz während der achtzehn minütigen Fahrt mit dem Testträger und einer Stichprobengröße von 31.455 Datennachrichten beträgt bei der Datenübertragung von See auf Land und sämtlicher datenverarbeitender Aktivitäten 153 Millisekunden. Es sind jedoch Ausschläge wegen einer zu geringen Übertragungsgeschwindigkeit der LTE Kommunikationsinfrastruktur resultierend aus dem Netzausbau zu verzeichnen, die zu einer schiffseitigen Vorhaltung der Daten durch RabbitMQ führen (siehe Kapitel 6.1.2). Aus diesen aufsteigenden und nach Erreichen des Scheitelpunktes wieder absteigenden Latenzen resultiert ein Maximalwert in den Randfällen der Latenz von 99,2 Sekunden. Zur Darstellung des üblichen Betriebes stellt die Tabelle 4 eine gefilterte Messung der Datenübertragung zwischen See und Land dar. Ein Filter mit einem Schwellwert von 1.000 Millisekunden entfernt die durch fehlenden Netzausbau begründeten Latenzen der Datenübertragung. Der übliche

Betrieb der Datenübertragung zwischen See und Land weist über LTE einen Mittelwert von 73 Millisekunden auf.

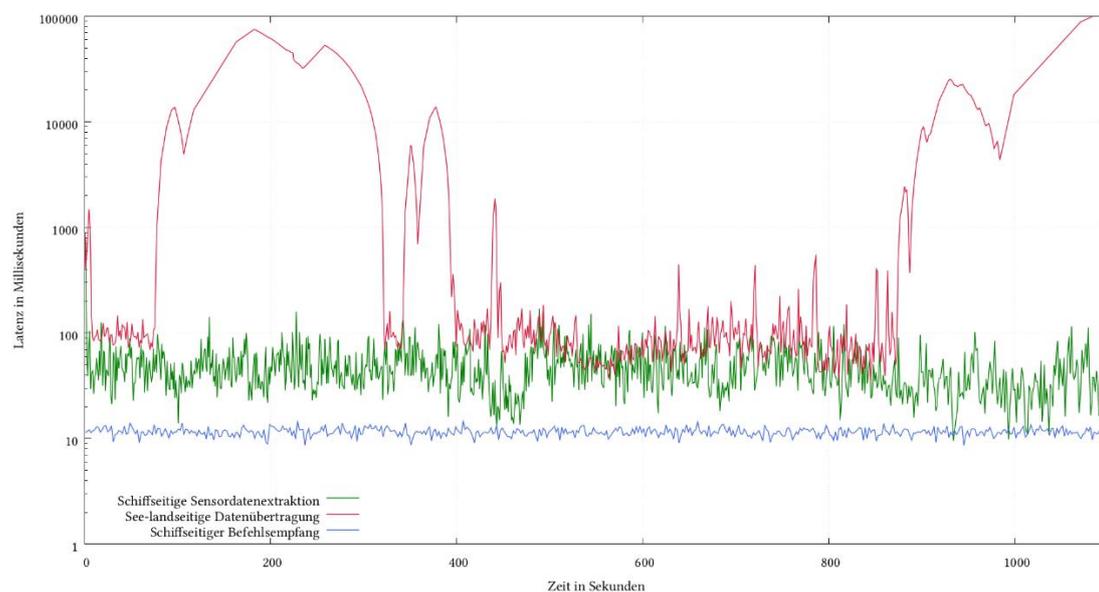


Abbildung 48: Latenzen der Sensordatenextraktion und des Befehlsempfangs der mobilen NaviBox auf dem Forschungsboot ZUSE über die Zeit

Die Messung der Latenz des schiffseitigen Befehlsempfangs mit einer Stichprobengröße von 9.381 über einen Zeitraum von achtzehn Minuten ergibt einen Mittelwert von 12 Millisekunden und einen Maximalwert von 26 Millisekunden.

**Simulations-Adapter.** Der Simulations-Adapter wird für die Anreicherung der Realwelt von LABSKAUS durch simulativ erzeugte AIS und Radardaten durch das virtuelle Testfeld HAGGIS genutzt und weist als Leistungsindikator die Latenz auf. Der Messaufbau des Simulations-Adapters entspricht wegen der Ähnlichkeiten dem der polymorphen Schnittstelle. Die Differenz des Zeitstempels einer Datennachricht in dem eingehenden Kommunikations-Handler und des Zeitstempels nach der erfolgten Transformation und dem Datenversand an einen RabbitMQ Broker durch den ausgehenden Kommunikations-Handler bemisst die Latenz. Die Messung ergibt eine durchschnittliche Latenz des Simulations-Adapters von 22,8 Millisekunden und einen Median-Wert von 21 Millisekunden, wie in Anhang I illustriert. Der Maximalwert der Latenz liegt zwar bei 196 Millisekunden, dieser wird jedoch in 99,99% der Fälle nicht erreicht.

## 6.1.2 Datenübertragungsrate

Das folgende Kapitel beschreibt die gemessene Datenübertragungsrate und das Datenaufkommen der mobilen und stationären NaviBoxen. Wegen der besonderen Relevanz der drahtlosen Datenübertragung findet die Messung der Übertragungsgeschwindigkeit der verwendeten LTE Kommunikationsinfrastruktur sowohl der stationären NaviBoxen wie der mobilen NaviBox an Bord des Testträgers statt. Die hierfür verwendete LTE Technologie wird seitens eines Drittanbieters zur Verfügung gestellt, somit unterliegt die Verfügbarkeit

und Datenübertragungsrate dem regionalen Netzausbau und ist stark schwankend. Die drahtlose Kommunikationsverbindung bildet die wesentliche Engstelle für die Datenübertragungsleistung des physikalischen Testfeldes.

**Datenaufkommen der NaviBoxen.** Ein Rechner mit einem Intel Core i7 7020HQ Prozessor und 16 GB PC3-17000 DDR3 Arbeitsspeicher misst die Datenübertragungsrate und das Datenaufkommen der stationären und mobilen NaviBoxen von LABSKAUS. Abbildung 49 zeigt die Messung der Datenübertragungsrate der mobilen NaviBox auf dem Forschungsboot ZUSE als Testträger für eine Dauer von 61,4 Minuten. Es werden durchschnittlich 152,8 Datennachrichten pro Sekunde erzeugt, welche eine durchschnittliche Größe von 215,9 Byte besitzen. Wenngleich die Datenübertragungsrate während eines Stillstands des Schiffes begründet durch die IMU zu erkennen ist, produziert die mobile NaviBox eine durchschnittliche Datenübertragungsrate von 33.007 Byte/s (33 Kilobyte/s) während der Fahrt.

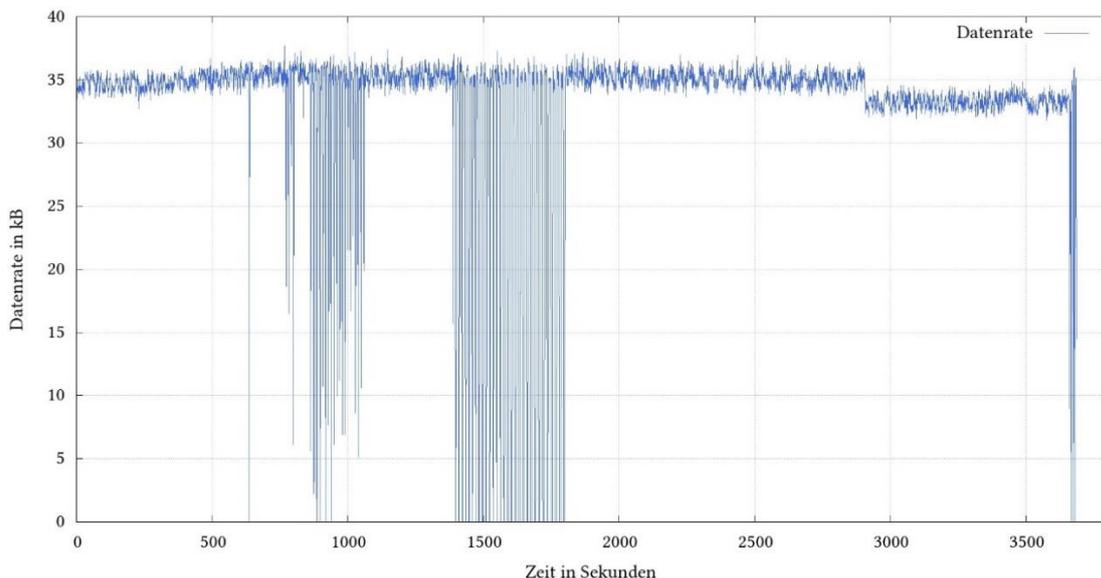


Abbildung 49: Datenübertragungsrate der mobilen NaviBox auf dem Forschungsboot ZUSE

Neben der Messung der Datenübertragungsrate der mobilen NaviBox auf dem Forschungsboot ZUSE findet die Messung der Datenübertragungsrate einer stationären NaviBox der Referenzwasserstraße in Brunsbüttel statt. Die stationäre NaviBox besitzt einen Windsensor, ein AIS und Radarsystem. Abbildung 50 illustriert das wesentlich konstantere Datenaufkommen der stationären NaviBox im Gegensatz zu der mobilen NaviBox. Die Datenpakete bemessen eine Größe zwischen weniger als einem Kilobyte und etwa 8,5 Kilobyte.

Über eine Dauer von 43,4 Minuten produziert die stationäre NaviBox 39.969 Datennachrichten mit einer durchschnittlichen Größe eines Datums von 273,1 Byte. Die durchschnittliche Anzahl der Datennachrichten pro Sekunde fällt geringer als bei der mobilen NaviBox aus, denn es werden lediglich 15,4 Nachrichten pro Sekunde erzeugt. Die Messung ergibt eine durchschnittliche Datenübertragungsrate von 4.194,7 Byte/s (4,2 Kilobyte/s) während des stationären Betriebes.

Im Vergleich über einen Zeitraum von 10 Minuten erzeugt eine stationäre küstenseitige NaviBox in Brunsbüttel eine Datenübertragungsrate von durchschnittlich 5 Kilobyte/s, während eine mit Motorsteuereinheit, Ruderwinkelindikator, Triducer, DGPS, AIS, IMU und Radarsystem ausgestattete mobile NaviBox an Bord des Forschungsbootes ZUSE eine Datenübertragungsrate von durchschnittlich 33 Kilobyte/s aufweist.

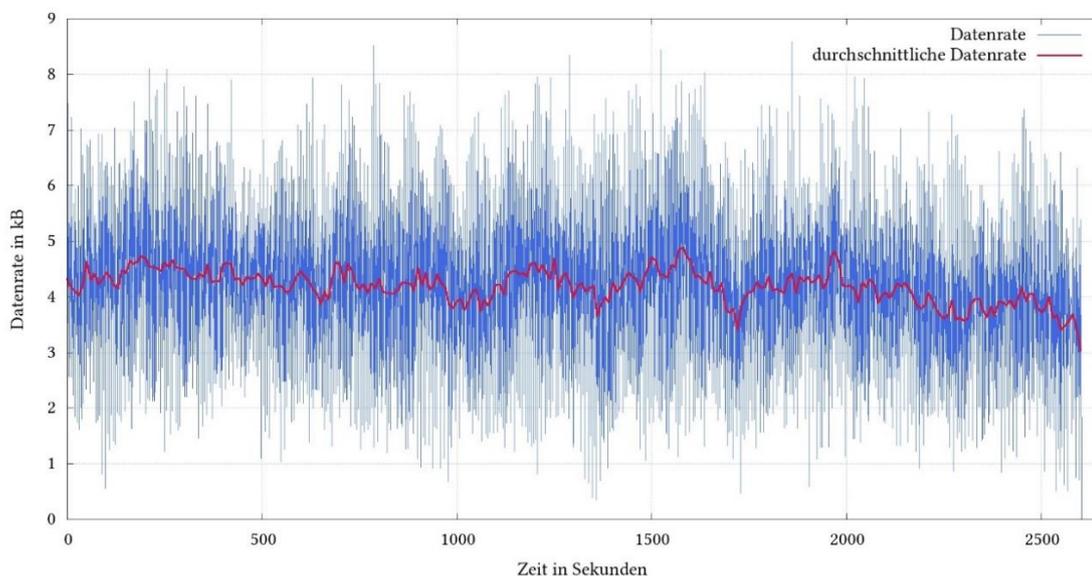


Abbildung 50: Datenübertragungsrate einer stationären NaviBox der Referenzwasserstraße in Brunsbüttel in Kilobyte/s

**Übertragungsgeschwindigkeit der LTE Kommunikationsinfrastruktur.** Die LTE Technologie ist theoretisch in der Lage eine Datenübertragungsrate im Download von bis zu 300 Mbit/s und im Upload von bis zu 50 Mbit/s bereitzustellen. Abhängig von dem Netzausbau misst eine Software die tatsächliche Übertragungsgeschwindigkeit der LTE Kommunikationsinfrastruktur, welche kontinuierlich die Datenübertragungsrate des Downloads und Uploads während der Übertragung einer Datei zu einem Server feststellt.

Abbildung 51 illustriert das Messergebnis für die stationären NaviBoxen in Brunsbüttel und Cuxhaven. Während des zehn minütigen Messzeitraumes treten schwach periodische und stark schwankende Datenübertragungsraten auf. Die durchschnittliche Datenübertragungsrate der stationären NaviBox in Cuxhaven über LTE im Upload beträgt 17.711 Megabit/s und minimal 9.773 Megabit/s, während diese im Download durchschnittlich 11.643 Megabit/s und minimal 4.754 Megabit/s betragen. Die durchschnittliche Datenübertragungsrate des Upload über LTE der stationären NaviBox in Brunsbüttel weist 4.218 Megabit/s und minimal 2.226 Megabit/s auf, während diese im Download durchschnittlich 10.412 Megabit/s und minimal 4.671 Megabit/s betragen. Wegen des wesentlich geringeren festgestellten Datenaufkommens der stationären NaviBoxen ist eine ausreichende LTE Kommunikationsinfrastruktur für die Datenübertragung der Referenzwasserstraße durch LABSKAUS vorhanden. Die Schwankungen der Datenübertragungsrate nehmen keinen negativen Einfluss, da die minimale Übertragungsgeschwindigkeit des LTE um ein Vielfaches höher als das Datenaufkommen ist.

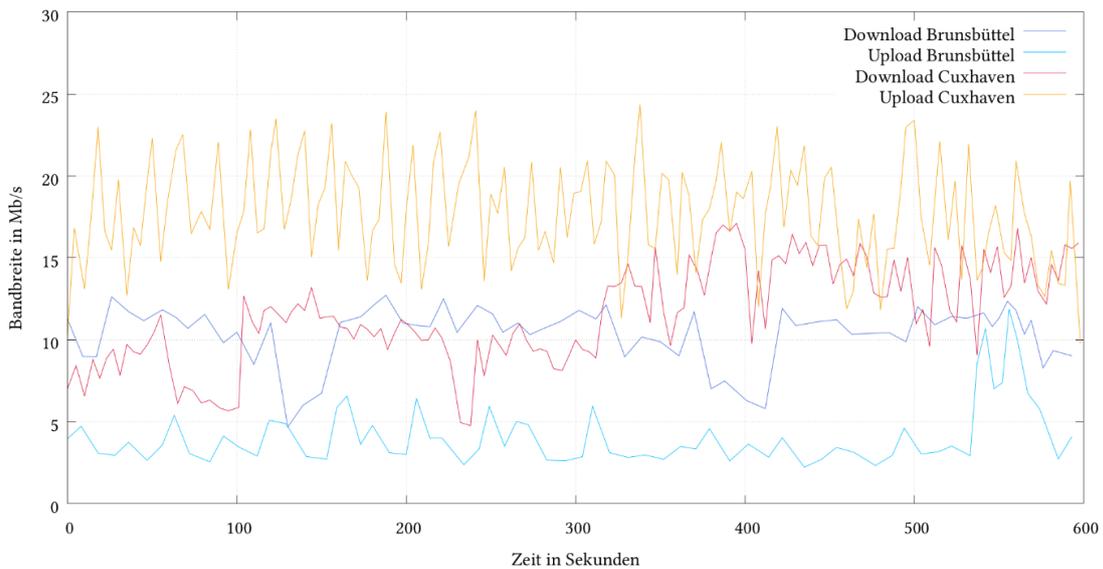


Abbildung 51: Übertragungsgeschwindigkeit der LTE Kommunikationsverbindung der stationären NaviBoxen der Referenzwasserstraße von LABSKAUS

Abbildung 52 stellt das Messergebnis für die mobile NaviBox an Bord des Forschungsbootes ZUSE während der Messung in Fahrt im Bereich der Innenjade dar. Während des 33-minütigen Messzeitraumes treten nicht periodische und stark schwankende Datenübertragungsraten über LTE auf.

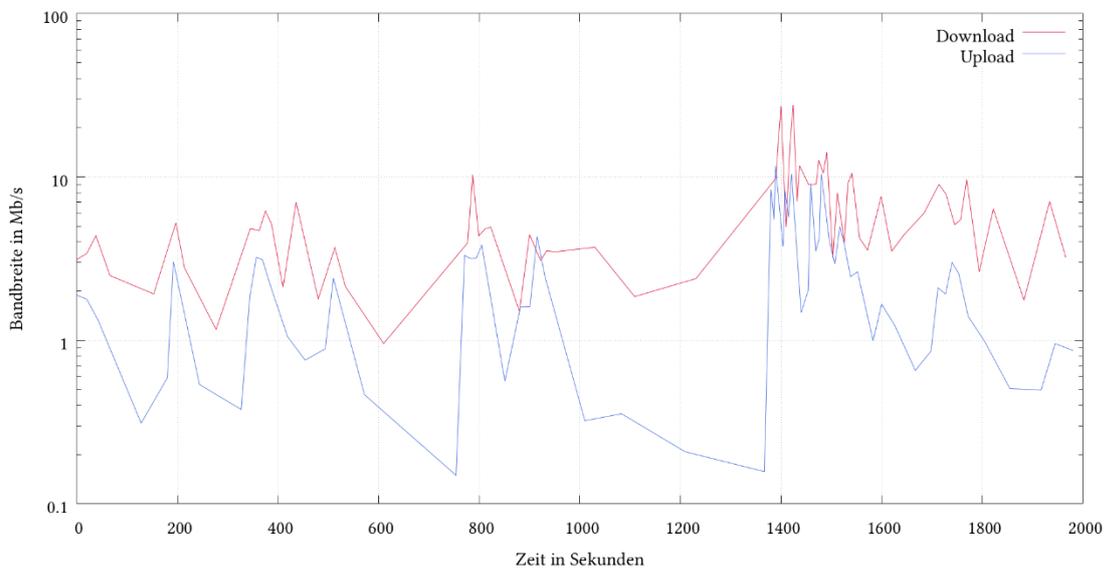


Abbildung 52: Übertragungsgeschwindigkeit der LTE Kommunikationsverbindung der mobilen NaviBoxen auf dem Forschungsboot ZUSE

Die durchschnittliche Datenübertragungsrate der mobilen NaviBox an Bord des Forschungsbootes ZUSE über LTE beträgt im Upload 2,71 Megabit/s und minimal 0,15 Megabit/s. Während der Messung beträgt die Datenübertragungsrate im Download durchschnittlich 6,38 Megabit/s und minimal 0,96 Megabit/s. Zwar ist die LTE Kommunikationsinfrastruktur überwiegend in der Lage das festgestellte Datenaufkommen der mobilen NaviBox an Land

zu übertragen. Dennoch treten vereinzelt Situationen auf, in denen dies wegen des unzureichenden Netzausbaus nicht möglich ist.

## 6.2 Explorative Datenerhebung für Szenarien-Generierung

Das Kapitel 2.2.5 stellt das Einsatzszenario eines physikalischen Testfeldes für die explorative Datenerhebung vor. Die explorierten Daten werden als Teststimulation des zu testenden Systems durch Simulatoren für einen Test im Closed-Loop-Verfahren oder Open-Loop-Verfahren verwendet, oder für die Modellbildung während der Definition und Gestaltung eines automatisierten Schiffsführungssystems. Das folgende Evaluationsexperiment setzt dieses Einsatzszenario durch die Verwendung von LABSKAUS für die Beobachtung des maritimen Verkehrssystems um. In diesem Rahmen dienen die Testfeld-Komponenten von LABSKAUS als Datenquelle zur Sammlung von Beobachtungsdaten des Verkehrs für die weitere Verwendung in einer Szenarien-Datenplattform.

Die Szenarien-Datenplattform dient der Speicherung und Bereitstellung von gemessenen Realweltdaten für Verwendung innerhalb der szenariobasierten V+V. Es erfolgt die Identifizierung von Szenarien und Speicherung dieser als Testszenarien für die zukünftige Anwendung durch die Bereitstellung für das Testen und Erforschen, beispielsweise die repräsentativen Testszenarien einer kritischen Verkehrssituation. Durch die Kombination einer Datenbank und Datenanalysekomponente erfolgt die Analyse und Identifikation der relevanten Testszenarien auf der Basis durch LABSKAUS beobachteter historischer Verkehrsdaten. Für die Analyse und Identifikation der Testszenarien dienen verschiedene Methoden der Wissensgenerierung. Weitergehend findet die Verifikation des entwickelten Algorithmus für die Identifizierung der Szenarien von kritischen Schiffsmanövern unter der Zuhilfenahme des Forschungsbootes ZUSE und der kontrollierten Ausführung und Messung der relevanten Manöver statt. Diese Verifikation der vermeintlich identifizierten Szenarien ist notwendig, da durch das reine Beobachten der Referenzwasserstraße nicht eindeutig bestimmbar ist, inwiefern ein kritisches Manöver zur Kollisionsvermeidung oder ein üblicher Richtungswechsel in einer unkritischen Situation vorliegt [LaHa17].

### 6.2.1 Evaluationsziel

In dem folgenden Einsatzszenario findet die Untersuchung der erfolgreichen Erfüllung einiger grundlegender Anforderungen an ein physikalisches Testfeld durch die Testfeld-Architektur statt. Zum einen erfolgt die Überprüfung der Datenbereitstellung der Schiffsumwelt (A1) und des Testträger-Schiffes (A2) gemäß der Testfeld-Architektur durch zu verwendende schiffseitige und küstenseitige Testfeld-Komponenten. Die erfassten Daten überträgt die bereitgestellte Kommunikationsinfrastruktur von LABSKAUS über LTE an die landseitige Szenarien-Datenplattform (A4). Aufgrund der Nutzung von drei stationären NaviBoxen

und den auf NMEA 2000 basierenden Sensoren entsteht eine erwartete gesamte Datenübertragungsrate von maximal 750 Kilobyte/s (A5), (A6). Die auftretende Latenz durch die Datenverarbeitung ist in diesem Einsatzszenario der Erhebung historischer Daten zu vernachlässigen. Durch den Aufbau lässt sich das erfasste Umweltszenario beobachten und eine Modellbildung sowie die Identifikation von Annahmen und Bedingungen zur Unterstützung der Systementwicklung realisieren (A8). Um ein interpretierbares Gesamtlagebild für die Beobachtung zu erhalten müssen die von den proprietären Teilkomponenten erfassten Daten innerhalb der Testfeld-Infrastruktur interoperabel aufbereitet werden (A19). Die lose Kopplung wird untersucht, indem mobile und stationäre Sensorsystemknoten in verteilter Position ein geografisch weiträumiges Gebiet der Nordseeküste erfassen. Dies erfolgt lose gekoppelt in dem die Daten der Szenario-Datenplattform über die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform zur Verfügung gestellt werden (A21), (A23).

Die verschiedenen notwendigen Sensoren generieren Datenströme, welche durch die ausgewählte Systemarchitekturgestaltung zentral übertragen und somit vielfältige Daten bereitgestellt werden, wie die Positionen, Windgeschwindigkeit und Windrichtung, Videodaten und schiffsbezogene Daten (A22). Das folgende Evaluationsexperiment zielt auf die Untersuchung der Fähigkeit der Testfeld-Architektur ab, die Datenströme durch die nachrichtenorientierte Middleware konfigurierbar zu gestalten (A25). Hierbei erfolgt die Konfiguration der Kommunikationsinfrastruktur, um die verteilten NaviBoxen als einzelne Systemknoten so zu konfigurieren, dass die Szenario-Datenplattform einen zusammengefassten und der Testkonfiguration entsprechenden Datenstrom erhält. Weitergehend findet die Konfiguration der Testfeld-Komponenten gemäß den Anforderungen des Evaluationsexperiments hinsichtlich der entsprechend gewünschten Daten und Aufbereitungsparameter statt (A26). Da eine langfristige Datenexploration vorgenommen wird und parallel weitere Anwendungen von LABSKAUS stattfinden, zeigt dieses Evaluationsexperiment die mögliche logische Trennung verschiedener Tests mit der Hilfe einer parallelen Nutzung des physikalischen Testfeldes (A28).

Für die Evaluation der Einsatzfähigkeit von LABSKAUS für die explorative Datensammlung dient der in Abbildung 53 dargestellte Aufbau des physikalischen Testfeldes und der Szenarien-Datenplattform. Durch dieses Einsatzszenario für den Testfeldaufbau für die explorative Datenerhebung durch LABSKAUS erfolgt die Überprüfung einer wesentlichen Teilmenge der zu erfüllenden Anforderungen.

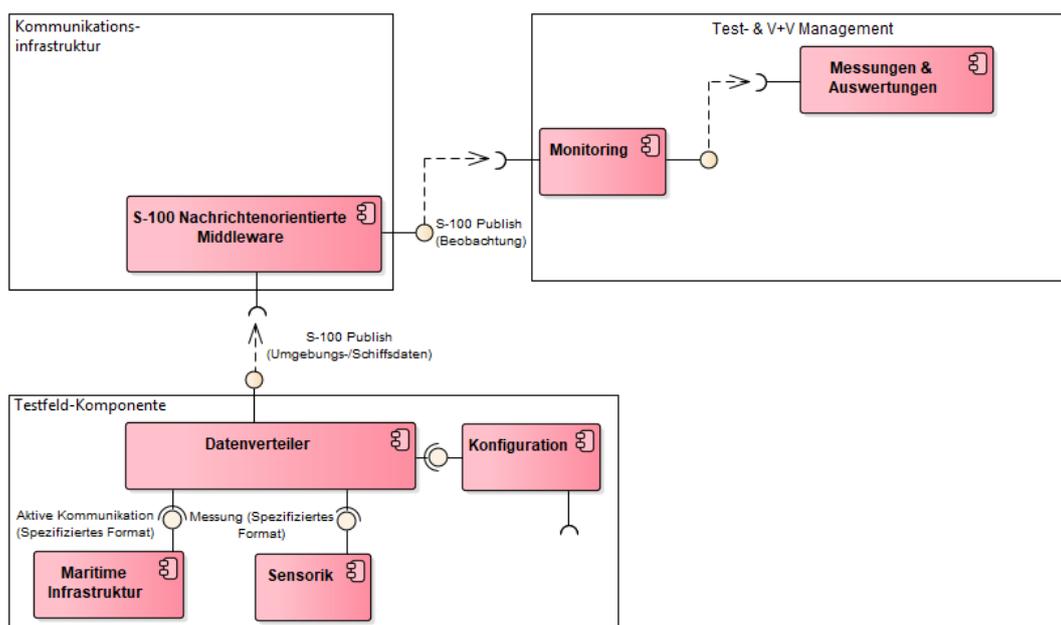


Abbildung 53: Aufbau des physikalischen Testfeldes für die explorative Datensammlung zur Unterstützung der Beobachtung von Annahmen, Bedingungen und Modellbildung für frühe Systementwicklungsphasen

## 6.2.2 Durchführung

Die Realisierung dieses Experiments durch LABSKAUS stellt der Szenarien-Datenplattform die notwendigen Beobachtungsdaten des maritimen Verkehrssystems durch die Referenzwasserstraße und das Forschungsboot ZUSE bereit. Die Referenzwasserstraße deckt die hochfrequent befahrene Seestrecke der Elbmündung des Nord-Ostsee-Kanals und der Jade ab. Zwei stationäre NaviBoxen in Cuxhaven und Brunsbüttel sammeln Beobachtungsdaten durch AIS Empfänger, Umweltsensoren und eine Radarerfassung, wie in Abbildung 54 dargestellt. Ergänzend sind zwei AIS Empfänger mit weitreichenden Antennen auf dem Dach der Jade Hochschule in Wilhelmshaven verbunden, die Wilhelmshaven, Bremerhaven bis Cuxhaven und die ostfriesischen Inseln bis Helgoland erfassen.

Die von einem AIS Empfänger vom Typ Digital Yacht AIS200N2K erfassten navigations- und schiffsbezogenen Daten der Referenzwasserstraße umfassen dynamische Schiffsdaten, statische Schiffsdaten und Reisedaten der mit einem AIS ausgestatteten Schiffe in einem von der Höhe der jeweiligen Antenne abhängigen Erfassungsbereich von 35-100 Kilometer. Eine Simrad 4G Radarantenne (Dauerstrichradar) erkennt und ortet ebenfalls die mit AIS ausgestatteten Schiffe und zusätzlich Schiffe ohne AIS in dem es einzelne Objektpositionen und aufeinanderfolgende Radarbeobachtungen des gleichen Ziels (engl. „Track“) bereitstellt. Die auf dem Industrie-PC einer jeden NaviBox installierte Radarerfassungssoftware ist Cambridge Pixel SPx. Ein advanSea S400 Windsensor misst die Windgeschwindigkeit und Windrichtung.



Abbildung 54: Stationäre NaviBoxen der Referenzwasserstraße von LABSKAUS in Cuxhaven (links) und Brunsbüttel (rechts)

Die stationären NaviBoxen der Referenzwasserstraße stellen den proprietären Sensorkomponenten sämtliche notwendige Schnittstellen auf der Basis von NMEA 2000 (AIS), dem proprietären Datenformat SPx Enhanced Data (Radar) sowie NMEA 0183 (Windsensor) bereit und verarbeiten die anfallenden Datenströme. Hierfür beherbergt eine NaviBox den Industrie-PC, der die Sensordatenstromverarbeitung durch eine eigens implementierte Software in C++ bewältigt. Gemäß der in Kapitel 4.3.3 beschriebenen Testfeld-Architektur findet die Konfiguration der NaviBoxen als Testfeld-Komponenten dezentral statt. Dies erfolgt über Konfigurationsbefehle, welche mit Hilfe der nachrichtenorientierten Middleware RabbitMQ über eine LTE Verbindung gesendet werden.

Die auf dem Industrie-PC installierte und in C++ eigenentwickelte Software besitzt eine statische Transformationslogik, welche die empfangenen Daten der inhärenten Schnittstellen der jeweiligen Datenquellkomponente in Instanzen der Datenobjekte entsprechend des einheitlichen Datenmodells gemäß S-100 der Testfeld-Architektur überführt, wie in Kapitel 5.3 erläutert. Anschließend erfolgt die Serialisierung der Datenobjekte durch einen hashbasierten Algorithmus für die Serialisierung und anschließender Veröffentlichung durch den sich auf dem Industrie-PC befindenden RabbitMQ Client (Publisher). Die drahtlose Datenübertragung erfolgt über LTE an einen in Oldenburg beim OFFIS e. V. verorteten Server. Dieser Server hostet den zentralen RabbitMQ Broker, der ein Vermittlungsregister *ServiceBusData* besitzt. An den zentralen RabbitMQ Broker veröffentlichen sämtliche RabbitMQ Clients (Publisher) der stationären NaviBoxen die Daten. Die RabbitMQ Warteschlange *DataBase* stellt die für das Anwendungsszenario der explorativen Datenerhebung über einen langen Zeitraum benötigten Beobachtungsdaten bereit.

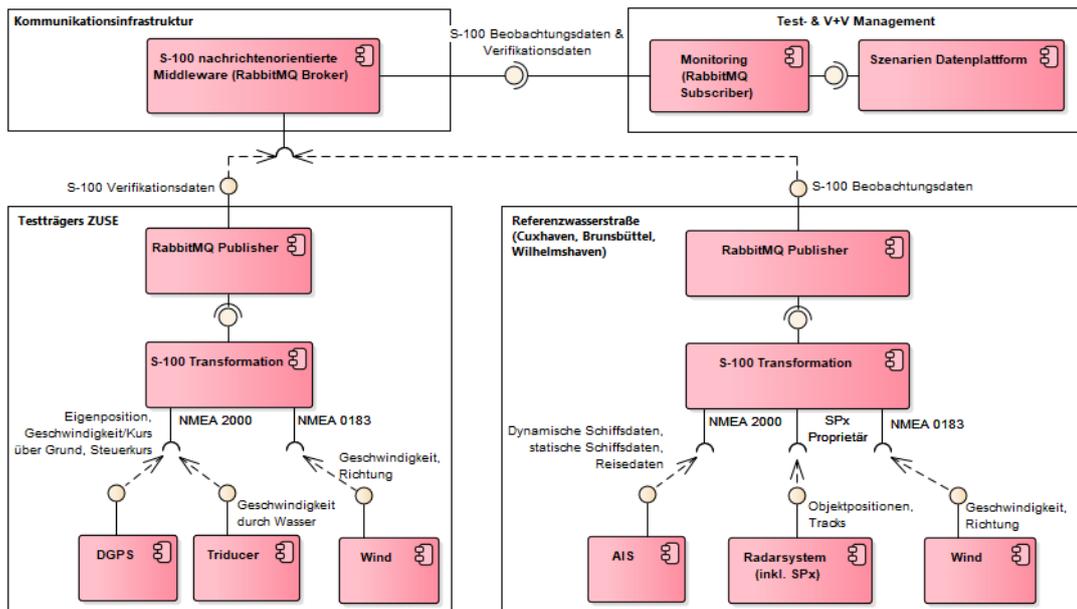


Abbildung 55: Testfeld-Aufbau von LABSKAUS für die explorative Datenerhebung für die Szenarien-Datenplattform

Die stationären NaviBoxen produzieren je Systemknoten eine konstante Datenübertragungsrate von 5 Kilobyte/s, während die mobile NaviBox an Bord des Forschungsbootes ZUSE eine Datenübertragungsrate von circa 33 Kilobyte/s aufweist, wie in Kapitel 6.1.2 gemessen. Weitergehend dienen Testfahrten mit dem Forschungsboot ZUSE als Verifikation der von der Szenarien-Datenplattform erkannten Manöver und deren Plausibilität, wie in Abbildung 56 dargestellt.



Abbildung 56: Forschungsboot ZUSE (links) und mobile NaviBox an Bord der ZUSE (rechts) für die Verifikation der durch die Szenarien-Datenplattform identifizierten Manöver

Hierfür erfolgt die manuelle Ausführung der Manöver durch einen Kapitän des Forschungsbootes ZUSE und währenddessen die Erfassung der schiffseigenen Daten durch eine mobile NaviBox an Bord der ZUSE. Die mobile NaviBox unterscheidet sich insofern von einer stationären NaviBox, als dass die schiffseigenen Informationen erfasst werden und daher die verwendeten Sensoren abweichen. Ein JRC JLR-21 DGPS misst die Eigenposition, Geschwindigkeit über Grund, den Kurs über Grund und den Steuerkurs des Testträgers. Ein Airmar DST800 Triducer ist in der Lage, die Geschwindigkeit der ZUSE durch das Wasser zu messen. Beide Sensoren sind ebenfalls über eine NMEA 2000 Schnittstelle an die mobile NaviBox

angeschlossen, welche die Sensordatenströme analog zu der stationären NaviBox in das einheitliche Datenmodell gemäß S-100 transformiert.

Entsprechend zu dem Veröffentlichen der Daten durch die stationären NaviBoxen erfolgt die Bereitstellung der schiffseigenen Daten der ZUSE ebenfalls über einen RabbitMQ Client (Publisher) an das Vermittlungsregister *ServiceBusData*. Der auf der mobilen NaviBox installierte RabbitMQ Broker leitet die Daten an die eigens für das Schiff ZUSE konfigurierte Warteschlange *ZUSE* an den zentralen RabbitMQ Broker in Oldenburg weiter. Ein Infrastruktur-zugriffsrechner mit einem installierten RabbitMQ Client (Subscriber) empfängt sowohl die Beobachtungsdaten der Referenzwasserstraße sowie die Verifikationsdaten von der RabbitMQ Warteschlange *DataBase* in Form von serialisierten Datenobjekten in einer S-100 entsprechenden Repräsentation über den zentralen RabbitMQ Broker des Servers in Oldenburg. Es folgt analog zu der Serialisierung eine Deserialisierung der empfangenen Datenobjekte in eine XML Datenstruktur. Die zu bedienende Szenarien-Datenplattform erhält die Datenobjekte im S-100 Format und verarbeitet diese intern weiter, um sie in einer PostgreSQL Datenbank zu archivieren und beispielsweise durch eine Benutzeroberfläche darstellen zu können, wie in Abbildung 57 dargestellt.

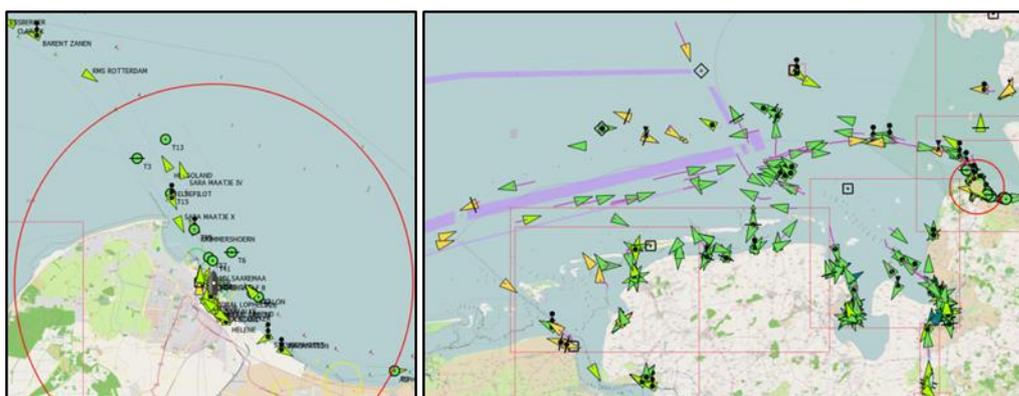


Abbildung 57: Ausschnitt der explorierten Daten des AIS und Radarsystems der LABSKAUS Infrastruktur in der deutschen Bucht für die Szenario-Datenplattform

Durch die in diesem Kapitel beschriebene Gestaltung des physikalischen Testfeldes LABSKAUS war die Anwendung der Testfeld-Architektur für das Einsatzszenario einer explorativen Datenerhebung vollumfänglich möglich. Die Ergebnisse des Einsatzes von LABSKAUS aus der Anwendungssicht der Szenarien-Datenplattform sind unter [LaHa17] nachvollziehbar.

### 6.2.3 Ergebnisse des Einsatzes für explorative Datenerhebung

Durch die von LABSKAUS bereitgestellte Infrastruktur ist es möglich der Szenario-Datenplattform für das Einsatzszenario der explorativen Datenerhebung die notwendigen Studienbedingungen zur Verfügung zu stellen. Hierfür erfasst ein verteilter Systemverbund aus stationären NaviBoxen und eine mobile NaviBox auf dem Forschungsboot ZUSE die Umgebung und Eigenschiffdaten. Anschließend überträgt die zentrale nachrichtenorientierte Middleware diese Datenströme logisch getrennt von weiteren Tests für die Weiterverarbeitung

durch das Zielsystem. Die stationären NaviBoxen erzeugen je Sensorsystemknoten eine kontinuierliche Datenübertragungsrate von 5 Kilobyte/s, während die mobile NaviBox eine Datenübertragungsrate von 33 Kilobyte/s erzeugt. Diese Daten sind im einheitlichen Datenmodell gemäß S-100 modelliert. Die für die drahtlose Übertragung eingesetzte LTE Kommunikationstechnologie besitzt eine minimale Datenübertragungsrate für den Upload von 2.226 Megabit/s der stationären NaviBox in Brunsbüttel und 9.773 Megabit/s der stationären NaviBox in Cuxhaven, wie in Kapitel 6.1.2 beschrieben. LABSKAUS ist somit in der Lage das anfallende Datenaufkommen der stationären NaviBoxen an das Zielsystem zu übertragen.

Das einheitliche Datenmodell gemäß S-100 zeigt während des Evaluationsexperiments die Fähigkeit, die unterschiedlichen Daten und Schnittstellen der proprietären Komponenten in interoperabler Weise abbilden zu können. Hierfür übernehmen die Testfeld-Komponenten die statischen Transformationen durch die in Kapitel 4.3.3 beschriebenen Operatoren. Die Interoperabilitätsmechanismen von S-100 führen zu der Fähigkeit der Modellierung sämtlicher für das Evaluationsexperiment notwendiger Sensoren in einem interoperablen Datenmodell, wie in Kapitel 4.3.1 beschrieben.

Die Konfiguration und Verteilung der verschiedenen Datenströme ist durch die nachrichtenorientierte Middleware in Form von RabbitMQ mit einem zentralen Broker in Oldenburg möglich. Diese Systemarchitekturentscheidung erweist sich insbesondere als positiv für die ganzheitliche Beobachtbarkeit der Referenzwasserstraße und des Testträgers. Die Konzepte der Warteschlangen und Vermittlungsregister der nachrichtenorientierten Middleware sowie verbundene Publisher und Subscriber führen zu einer hohen Konfigurierbarkeit verschiedener Datenströme. Insbesondere zeigt diese Gestaltungsentscheidung einer klaren Trennung der Kommunikationsinfrastruktur und des Datenmodells die Fähigkeit der Erweiterbarkeit und losen Kopplung. Veränderungen des Datenmodells oder einzelner Testfeld-Komponenten resultieren nicht in notwendigen Veränderungen anderer Komponenten oder der Kommunikationsinfrastruktur.

Der in dieser Arbeit vorgestellte Ansatz der Testfeld-Architektur ist fähig, mehrere geographisch verteilte Systemknoten zu einer ganzheitlichen Testfeld-Infrastruktur zusammenzuschließen, wie beispielhaft im Bereich der deutschen Nordseeküste dargestellt. Insbesondere erfolgt während der Anwendung der Szenarien-Datenplattform die Integration weiterer Systemknoten in Form des Forschungsbootes ZUSE und einer mobilen NaviBox an dessen Bord. Die Ausführung der kontinuierlichen Datenbereitstellung durch die stationären NaviBoxen wird hierdurch nicht beeinflusst. Die nachrichtenorientierte Middleware mit einem Publish-Subscribe-Kommunikationsmuster bietet die geforderte Flexibilität des physikalischen Testfeldes mit Fokus auf Erweiterbarkeit und Wiederverwendung in unterschiedlichen Testszenarien. Insbesondere für automatisierte Schiffsführungssysteme, deren technologischen Konzepte und Infrastrukturen gegenwärtig nicht vollständig entwickelt sind, ermöglicht die vorherrschende Generizität der entwickelten Testfeld-Architektur eine nachhaltige Grundlage für zukünftige Testszenarien. Die von der IALA geforderte technologische Unbeschränktheit durch aktuelle Architekturen, Datenstrukturen oder Prozesse wird durch den in dieser Arbeit vorgestellten Lösungsansatz erfüllt [Iala17].

Die Szenarien-Datenplattform ist durch die Bereitstellung der maritimen Verkehrsaktivitäten in der Lage, Muster innerhalb der maritimen Schifffahrt auf der Basis einer großen Datengrundlage zu identifizieren und diese mit der Hilfe der Eigendaten des Forschungsbootes ZUSE erfolgreich zu verifizieren. Das Testfeld LABSKAUS als Realisierung der in dieser Arbeit vorgestellten Lösung durch die Testfeld-Architektur erreicht die geforderten Ergebnisse zur Unterstützung der Systementwicklung durch die Anwendbarkeit für das Einsatzszenario der explorativen Datenerhebung. Die identifizierten Verkehrsszenarien werden im Rahmen der Forschung und Entwicklung für die Bildung von Modellen genutzt, die in der simulativen Erprobung die Realweltszenarien nachbilden oder komplementieren. LABSKAUS bietet somit eine wiederverwendbare Testumgebung für die Bildung von Annahmen, Bedingungen und Modellen zur Unterstützung der explorativen Modellbildung in frühen Systementwicklungsphasen.

### **6.3 Open-Loop-Test eines Kollisionsvermeidungssystems**

Der Test im Open-Loop-Verfahren bezeichnet eine Testmethode, in der keine Rückkopplung der aktuellen Zustandsdaten des physikalischen Testfeldes zur Steuerung des Testszenarios durch manipulierte, emulierte oder simulierte Teststimuli erfolgt, wie in Kapitel 2.2.5 beschrieben. Dieses Einsatzszenario eines physikalischen Testfeldes dient insbesondere dem dynamischen Testen der Anforderungs- und Eigenschaftenerfüllung des zu testenden Systems gemäß Konformität im Rahmen eines Realwelttests, in dem ein zu testendes System stimuliert wird. Das folgende Evaluationsexperiment realisiert dieses Einsatzszenario durch den Einsatz von LABSKAUS als Testinfrastruktur für die Bereitstellung der notwendigen operativen Umgebung und Beobachtung sowie Auswertung der Testdurchführung.

Die Entwicklung des maritimen Teilautomatisierungssystems für die Kollisionsvermeidung mit der Bezeichnung Maritime Traffic Collision Avoidance System (MTCAS) erfolgt im Zeitraum 2016-2018 und wird durch das Bundesministerium für Wirtschaft und Energie gefördert. An der Umsetzung ist das OFFIS e. V. in Oldenburg beteiligt. Entsprechend des ähnlichen in der Luftfahrt existierenden TCAS nutzt MTCAS alle einem Schiff zur Verfügung stehenden Informationen, um frühzeitig und möglichst exakt ein Lagebild unter anderem durch die Vorhersage potenzieller konfliktärer Schiffsbegegnungen durchzuführen. Gemäß den in Kapitel 2.1 dargestellten Stufen für automatisierte Schiffsführungsfunktionen stellt MTCAS ein Teilautomatisierungssystem dar. Hierdurch erfolgt eine verbesserte Wahrnehmung der Umgebung und Kollisionsvermeidung für die ein Schiff steuernde Besatzung und ein küstenseitiges VTS. Anwendungsszenarien existieren sowohl in Form verschiedener Schiffsausstattungen (mit oder ohne MTCAS) sowie durch die Berücksichtigung der unterschiedlichen Verkehrshinweise und Ausweichempfehlungen des MTCAS während der Schiffsbegegnungen [DBFH16]. Der in diesem Evaluationsexperiment dargestellte Test im Open-Loop-Verfahren nutzt das Forschungsboot ZUSE um die zum Test in der operativen

Umgebung von MTCAS notwendigen Daten als Teststimuli bereitzustellen und durch LABSKAUS die Testausführung zu überwachen.

### 6.3.1 Zielsetzung

Das Kapitel 6.2 untersucht die Erfüllung einiger grundlegender Anforderungen an ein physikalisches Testfeld zur Beurteilung der Testfeld-Architektur. Die folgende Evaluation stellt zwar die gleichen Anforderungen an die Testfeld-Architektur, erweitert die Zielsetzung der Evaluation jedoch durch ein zu testendes System im Gegensatz zu der explorativen Datenerhebung. Neben den untersuchten Anforderungen der Datenbereitstellung der Stimulationsdaten der Schiffsumwelt und des Testträger-Schiffes erfolgt bei dem in diesem Kapitel durchgeführten Evaluationsexperiment die Erweiterung des Betrachtungsraums. Es finden die Beobachtung der Reaktion des zu testenden Systems und dessen dynamische Integration durch die polymorphe Schnittstelle statt. Da MTCAS ein Teilautomatisierungssystem ist, das hochfrequente Sensordatenströme verarbeitet, stellt dieses Evaluationsexperiment Anforderungen an LABSKAUS bezüglich der notwendigen Latenz zwischen der Datenerfassung und Bereitstellung von maximal einer Sekunde (A5). Als Teststimuli dienen die mobile NaviBox und zwei stationäre NaviBoxen für die Beobachtung der Testdurchführung. Hierbei erzeugt die mobile NaviBox mit auf NMEA 0183 beziehungsweise NMEA 2000 basierender Sensorik eine maximale Datenübertragungsrate von 250 Kilobyte/s, die als Teststimuli an das MTCAS übertragen werden. Weitergehend erzeugen die drei stationären NaviBoxen, welche ebenfalls auf NMEA 0183 beziehungsweise NMEA 2000 basierende Sensorik beherbergen, für die Testdurchführung eine über drahtlose Kommunikationsinfrastruktur des Testfeldes zu transferierende Datenübertragungsrate von maximal 750 Kilobyte/s (A6, A4).

Ein ergänzender Aspekt dieses Evaluationsexperiments ist die ganzheitliche Beobachtungsfähigkeit der Testausführung hinsichtlich der Umwelt, Eingabe- und Ausgabedaten sowie systeminterne Parameter des zu testenden Systems und des Zustandes der Testfeld-Infrastruktur. Aus diesem Grund muss der Testbeobachtungsraum dahingehend erweitert werden, als dass neben den Eingangstimuli und Ausgabereaktionen weitergehend die Rahmenbedingungen des Testszenarios beobachtet werden (A8, A9, A10, A11). Auf der Basis einer ganzheitlich beobachteten Testausführung und der Testrahmenbedingungen ist die Evaluation der Messung gemäß der erwarteten Funktionalität der MTCAS Testspezifikation auszuwerten. Hierdurch erfolgt der Systemtest des MTCAS in Hinblick auf die Erfüllung der Konformität und korrekten Funktionalität gemäß den Testspezifikationen (A12).

Das folgende Evaluationsexperiment stellt die Fähigkeit des physikalischen Testfeldes dar, zu testende Systeme flexibel durch die polymorphe Schnittstelle integrieren zu können ohne die Infrastruktur von LABSKAUS oder das zu testende System zu modifizieren. MTCAS besitzt eine proprietäre Schnittstelle, die ohne die polymorphe Schnittstelle durch die verschiedenen Komponenten des physikalischen Testfeldes nicht ohne weiteres interoperabel zu bedienen ist. Für die notwendige Integration des MTCAS in die Testfeld-Infrastruktur erfolgt der Einsatz der polymorphen Schnittstelle (A20). Für die Evaluation der aufgeführten Anforderungen bietet sich das Einsatzszenario des Tests im Open-Loop-Verfahren an. Daher

wird der in Abbildung 58 dargestellte Testfeld-Aufbau für das dynamische Testen gemäß dem Open-Loop-Verfahren genutzt.

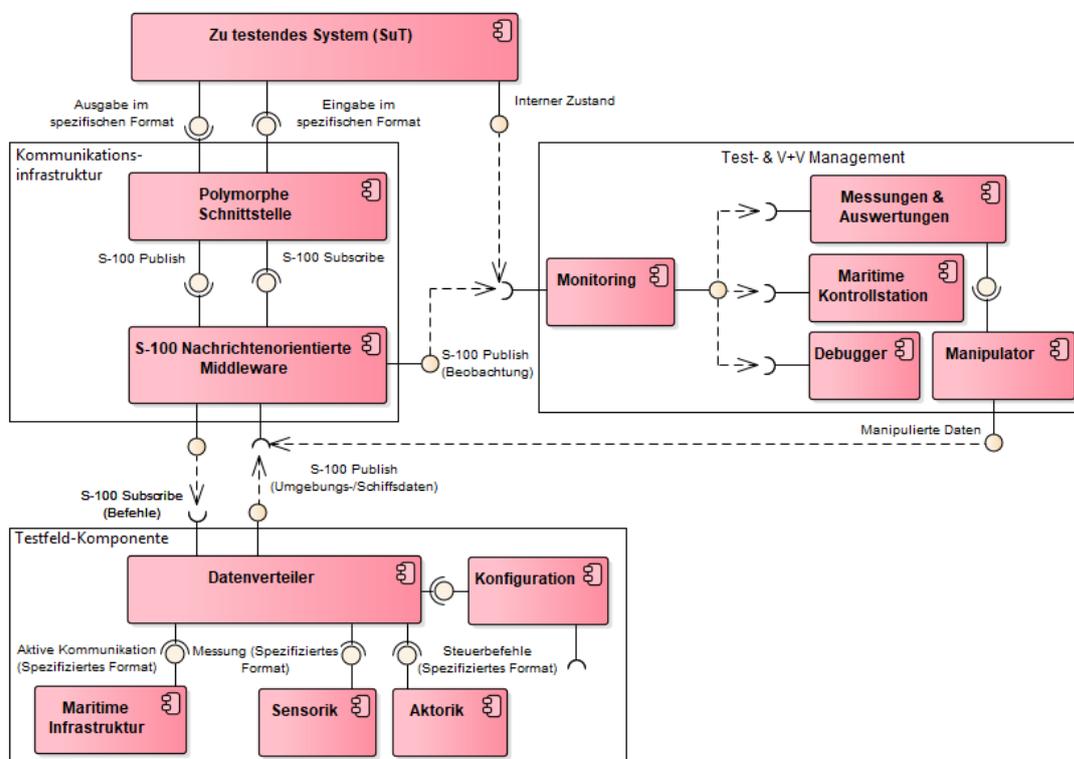


Abbildung 58: Aufbau des physikalischen Testfeldes für das dynamische Testen im Open-Loop Verfahren

### 6.3.2 Durchführung

Der Schwerpunkt dieses Evaluationsexperiment liegt in der dynamischen Integration des MTCAS in die in Kapitel 6.3.1 dargestellte Testfeld-Infrastruktur von LABSKAUS durch die polymorphe Schnittstelle. LABSKAUS stellt für die Bereitstellung der vom MTCAS benötigten Teststimuli das Forschungsboot ZUSE zur Verfügung. Hierdurch stellt LABSKAUS dem MTCAS als zu testendes System die notwendige Testumgebung bereit, ohne dass ein individueller Testaufbau oder eine Anpassung des MTCAS notwendig ist. Die Durchführung der finalen Erprobung des MTCAS erfolgte vom 03. Juli 2018 bis zum 06. Juli 2018 und vorherige Einzeltests der Verhaltensprädiktion gemäß im Vorfeld definierter Testszenarien und mit in Absprache mit den Systementwicklern des MTCAS ausgewählten Testmethoden. Für die Durchführung des finalen Systemtests auf der Innenjade wird das nicht zu LABSKAUS gehörende Testträger-Schiff Senckenberg als Fremdschiff für Kollisionsszenarien einbezogen. Für den Test der Verhaltensprädiktion dient der Schiffsverkehr auf der Innenjade und für die dynamische Integration des zu testenden Systems die polymorphe Schnittstelle.

**Testfeld-Aufbau.** Die vom MTCAS an LABSKAUS gestellten Anforderungen variieren in einigen Aspekten zu dem in Kapitel 6.2 vorgestellten Evaluationsexperiment, weshalb eine

Erweiterung der Sensorausstattung des Forschungsbootes ZUSE notwendig ist. Wie in Abbildung 59 dargestellt, ergänzt die Installation eines Ruderwinkelindikators der Firma Volvo Penta den Aufbau, dessen Daten nach einer proprietären Digitalisierungseinheit über eine NMEA 2000 Schnittstelle in das einheitliche Datenmodell gemäß S-100 von LABSKAUS transformiert werden. Hiervon abgesehen liegt die gleiche Installation des im vorherigen Kapitel beschriebenen Aufbaus des Forschungsbootes ZUSE vor (siehe Kapitel 6.2.2).

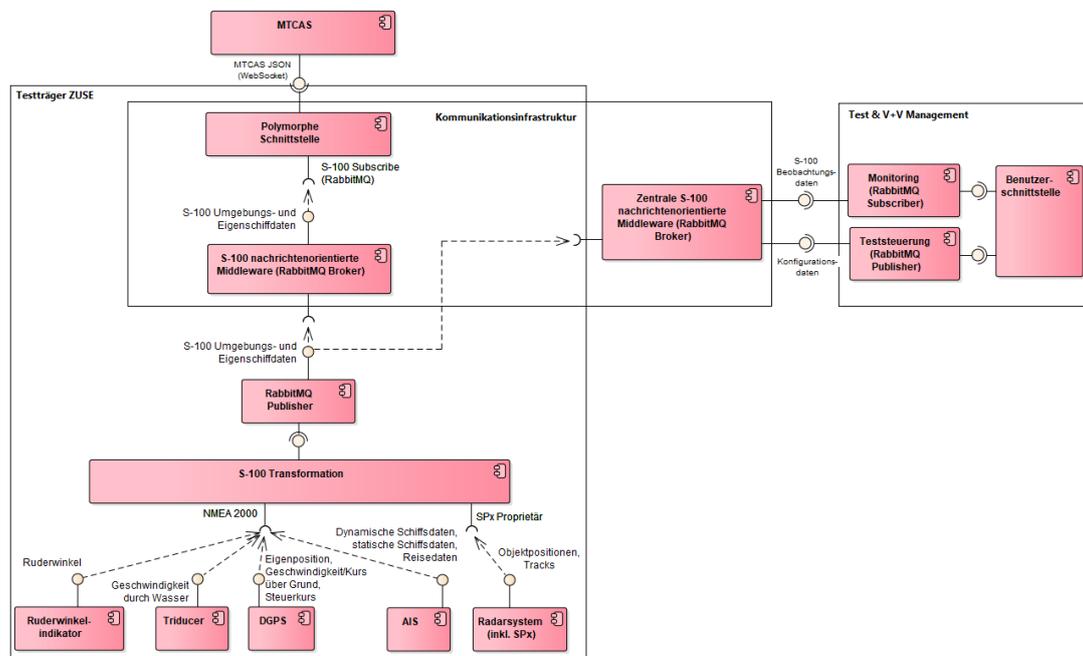


Abbildung 59: Testfeld-Aufbau von LABSKAUS für den Test im Open-Loop-Verfahren des MTCAS

Da das MTCAS ein schiffseitiges System ist existiert auf dem Industrie-PC der mobilen NaviBox ein RabbitMQ Broker, welcher die veröffentlichten Daten des RabbitMQ Publisher entgegennimmt und den konsumierenden Systemen über die zentrale nachrichtenorientierte Middleware gemäß S-100 bereitstellt. Im Fall des MTCAS basiert die Schnittstelle auf dem kabelgebundenen Datennetz Ethernet. Sowohl die schiffseitig installierte polymorphe Schnittstelle für die dynamische Integration des MTCAS sowie die Benutzeroberfläche für das Monitoring zur Testbeobachtung und Testauswertung abonnieren diesen Datenstrom der Warteschlange MTCAS der zentralen nachrichtenorientierten Middleware implementiert mittels eines RabbitMQ Brokers durch entsprechende RabbitMQ Clients (Subscriber).

Die mobile NaviBox veröffentlicht durch einen RabbitMQ Publisher die Datenströme für die zentrale nachrichtenorientierte Middleware drahtlos über LTE, um komplementierend durch die Referenzwasserstraße das ganzheitliche Lagebild für die Beobachtung bereitzustellen. Die zentrale nachrichtenorientierte Middleware wird durch einen in Oldenburg bereitgestellten Server realisiert, der einen RabbitMQ Broker für die Verwaltung sämtlicher Datenströme bietet.

**Integration des zu testenden Systems.** Abbildung 60 stellt die Konfiguration der polymorphen Schnittstelle für die dynamische Integration des MTCAS in die Infrastruktur von

LABSKAUS dar. Die Pipeline zur Transformation der Datenströme benötigt fünf Filterkomponenten, welche über die Konfigurationsdatei der polymorphen Schnittstelle in XML spezifiziert werden. Der eingehende Kommunikations-Handler baut die Verbindung zu dem RabbitMQ Broker der nachrichtenorientierten Middleware mit Datenmodell gemäß S-100 auf, um entsprechende Daten von der Warteschlange über das Kommunikationsprotokoll AMQP zu erhalten. Dieser Kommunikations-Handler erhält die Konfigurationsparameter, um sich mit der für den Test des MTCAS konfigurierten Warteschlange *MTCAS* zu verbinden. Der eingehende Syntax-Handler empfängt die Datenobjekte der nachrichtenorientierten Middleware und wandelt diese in ein Array vom Datentyp *short* um. Dieses umgewandelte Array wird anschließend in ein EObject des durch das Ecore-Metamodell modellierten einheitlichen Datenmodells gemäß S-100 deserialisiert. Anschließend transformiert der eingehende Syntax-Handler das S-100 Datenobjekt in eine XML Repräsentation. Die Übergabe dieses S-100 Datenobjekts im XML Format an den Semantik-Handler ermöglicht die semantische Transformation in das proprietäre MTCAS Format. Nach erfolgter Schemavalidierung des Ecore S-100 und XSD MTCAS Datenmodells transformiert der XSLT-Prozessor der Java-Bibliothek *java.xml.transform* das Datenobjekt semantisch vom Typ S-100 in ein Datenobjekt vom Typ MTCAS proprietär.

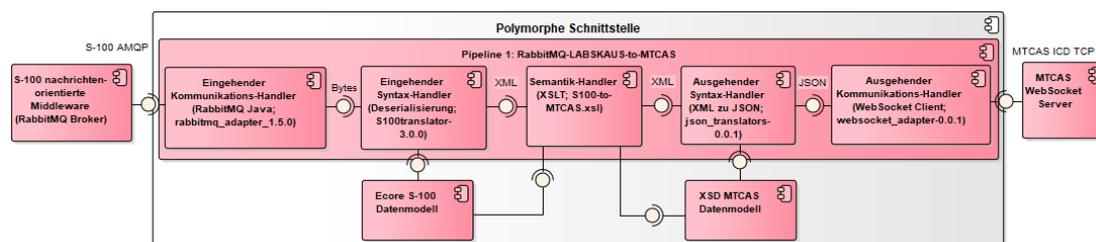


Abbildung 60: Integrationskonfiguration der polymorphen Schnittstelle für MTCAS

Der ausgehende Syntax-Handler überführt die XML Repräsentation des Datenobjekts vom Typ MTCAS proprietär in eine JSON Repräsentation gemäß der inhärenten Schnittstelle des MTCAS. Hierbei sind einige strukturelle Anpassungen vorzunehmen, wie das Entfernen der bei der XSLT Transformation gesetzten Elemente `<root>`, `</root>` und `<root/>` sowie `<myItem>`, `</myItem>` und `<myItem/>`. Andernfalls würden sie im JSON Format als Objekte interpretiert werden. Für die syntaktische Transformation dient die Programmbibliothek *org.json*, die mit der Methode *org.json.XML.toJSONObject(String xml)* eine entsprechende Transformationslogik bietet. Der Rückgabewert wird in einen Bytestrom transformiert und an den ausgehenden Kommunikations-Handler weitergeleitet. Der ausgehende Kommunikations-Handler sendet gemäß der Netzwerkkonfiguration des MTCAS WebSocketServer die Daten an das zu testende System, in dem die *org.java\_websocket* Programmbibliothek genutzt wird. Die Steuerung des Kommunikationsprotokolls erfolgt durch die Methoden der *org.java\_websocket* Programmbibliothek, wie das Bestätigen einer eingegangenen Datenrichtung oder ein eventuelles Nachsenden weiterer Datenrichtungen.

**Testmethoden.** Der durch das physikalische Testfeld LABSKAUS realisierte Test des MTCAS zielt auf die Durchführung eines funktionalen Systemtests im Feld anhand der zuvor

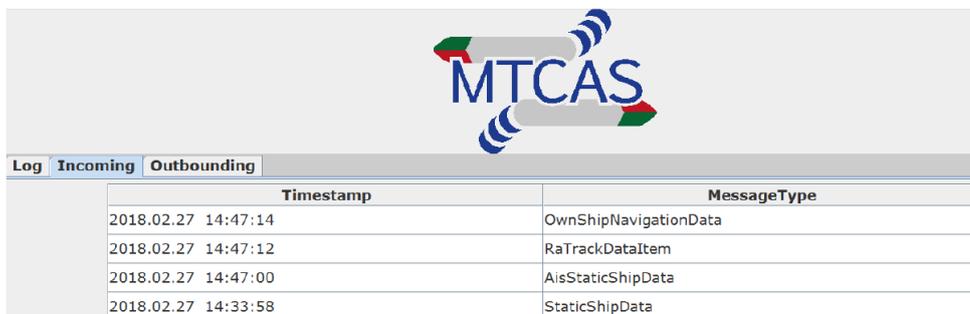
von den Systementwicklern identifizierten Testszenarien und Testfälle ab. Dieser Test erfolgt in dynamischer Ausführung des MTCAS in der Realwelt unter operativen Bedingungen. Hierdurch findet die Überprüfung der Konformität der Systemspezifikationen und somit der Anforderungserfüllung des MTCAS statt. Insgesamt existieren neun Testfälle in drei Testszenarien, die mit der Hilfe von LABSKAUS überprüft werden. Die Testszenarien des Systemtests definieren maritime Szenarien, für die das manuelle Nachstellen spezieller Situationen der Schiffsbegegnung notwendig sind. Die ZUSE als Testträger erfasst gemäß dem jeweiligen Testfall die relevanten Umgebungs- und Eigenschiffdaten und stimuliert somit das MTCAS während der Testausführung.

Zunächst erfolgt ein Systemintegrationstest des MTCAS und LABSKAUS. Der Systemintegrationstest verifiziert die korrekte Funktionsweise der polymorphen Schnittstelle hinsichtlich einer erfolgten Kommunikationsverbindung sowie der syntaktischen Korrektheit der Daten. Anschließend beurteilt ein Systemexperte des MTCAS die semantische Korrektheit einiger empfangener Live-Daten der mobilen NaviBox, die dem MTCAS durch LABSKAUS bereitgestellt und ein plausibles Ergebnis anhand der vorliegenden Situation der Realwelt validiert wird.

Im Anschluss an den Systemintegrationstest findet ein funktionaler Systemtest im Black-Box-Verfahren statt, bei dem das MTCAS im operativen Betrieb ausgeführt wird. Die durch MTCAS spezifizierten Testszenarien orientieren sich an den im Rahmen der Systementwicklung definierten Anforderungen. Nach der erfolgten Funktionsausführung des MTCAS findet die manuelle Überprüfung und Beurteilung der funktionalen Korrektheit durch zwei Systementwickler des MTCAS im Vier-Augen-Prinzip statt. Hierfür werden die Benutzeroberfläche des MTCAS, Werkzeuge der Systementwickler für V+V und teilweise eine Benutzeroberfläche von LABSKAUS für das Monitoring zur ganzheitlichen Beobachtung der Referenzwasserstraße verwendet. Eine Aussage über die Korrektheit findet anhand der Plausibilitätsprüfung der vom MTCAS errechneten Parameter statt, wie der kritischen Schiffsposition (engl. „Critical Ship Pose“) oder den Berechnungsergebnissen der MTCAS Komponente für die Verhaltensprädiktion (engl. „Behaviour Predictor“).

**Testdurchführung.** Zunächst erfolgt der Systemintegrationstest des MTCAS und des Testträgers ZUSE in dem das Umweltlagebild im Hafengebiet und die Eigenschiffdaten der ZUSE an das MTCAS gesandt werden. Die Schnittstelle des MTCAS ist in der Lage, die syntaktische und strukturelle Korrektheit der eingehenden Daten zu verifizieren und protokolliert auftretende Fehler. Das MTCAS besitzt hierfür eine Diagnose-Software, die in einem Ausführungsprotokoll die Korrektheit oder auftretende Fehler der empfangenen Daten darstellt, wie in Abbildung 61 illustriert. Für den Systemintegrationstest dienen 563.347 Datensätze der mobilen NaviBox, die durch LABSKAUS fehlerfrei dem MTCAS über die polymorphe Schnittstelle zur Verfügung gestellt werden. Ein Experte des MTCAS validiert ergänzend die erfolgreiche Integration durch die Überprüfung der semantischen Korrektheit der Daten in einem Vergleich mit dem Realweltszenario vor Ort. Anschließend werden in dem durch die Referenzwasserstraße erfassten Testbereich der Innenjade die Testszenarien der Schiffs-

begegnungen durchgeführt, um die Detektion und Verhaltensprädiktion des MTCAS zu testen. Hierfür validiert ein Experte die korrekte Funktionalität der Verhaltensprädiktion und der Eskalationszustände des MTCAS in einem Test im Open-Loop-Verfahren während eines Überholmanövers sowie entgegengesetzter und kreuzender Kurse.



Log		Incoming	Outbounding
		Timestamp	MessageType
2018.02.27	14:47:14		OwnShipNavigationData
2018.02.27	14:47:12		RaTrackDataItem
2018.02.27	14:47:00		AisStaticShipData
2018.02.27	14:33:58		StaticShipData

Abbildung 61: Diagnose-Software des MTCAS für den Systemintegrationstest in LABSKAUS

Neben der Beurteilung der Testausführung auf der Basis der Experteneinschätzung des MTCAS findet die Beobachtung der Testdurchführung teilweise durch eine Benutzeroberfläche für das Monitoring von LABSKAUS mit der Bezeichnung MATEControl und weiterer Werkzeuge der Systementwickler für die V+V des MTCAS statt. Die Java Anwendung MATEControl (Maritime Test and Experimental Platform Control) ist an die zentrale nachrichtenorientierte Middleware mit einheitlichem Datenmodell gemäß S-100 gekoppelt und ermöglicht die ganzheitliche Beobachtung des Tests auf der Basis der stationären und mobilen NaviBoxen ohne einen Einfluss auf die Testdurchführung zu nehmen.

Abbildung 62 stellt die Durchführung eines Testszenarios zur Überprüfung eines Testfalls dar, in dem die Verhaltensprädiktion des MTCAS für das fremde Schiff auf der Basis der durch den Testträger ZUSE bereitgestellten Teststimuli erfolgt.

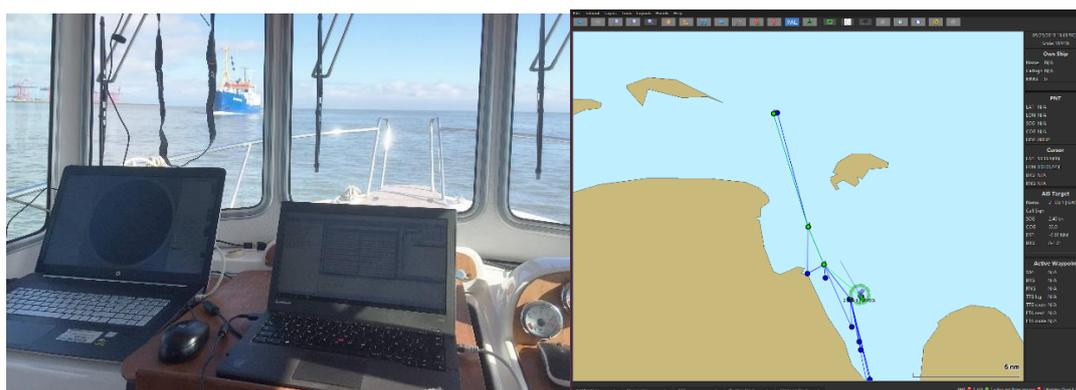


Abbildung 62: MTCAS Anwendung während des szenariobasierten Systemtests im Open-Loop-Verfahren auf dem Testträger ZUSE. Experte des MTCAS überprüft die korrekte Funktionserfüllung während der Testträger die notwendigen Teststimuli durch eine mobile NaviBox erfasst und an das MTCAS sendet.

Die grüne Linie der elektronischen Seekarte ist die vorhergesagte Trajektorie des fremden Schiffes während die blaue Linie die eigene Trajektorie darstellt. Das AIS des Schiffes ZUSE erfasst die dynamischen und statischen Daten der Fremdschiffe während der Ruderwinkel-Indikator, Triducer und das DGPS die Eigenschiffdaten der ZUSE bereitstellen.

Auf der Basis dieser Sensordaten erfolgt die Berechnung der Trajektorien und Ausweichempfehlungen. Der Systemexperte überprüft anhand der Testspezifikation das Systemverhalten des MTCAS und beurteilt basierend auf den eingehenden Daten und Ausführungsergebnissen die korrekte Funktionsfähigkeit gemäß dem jeweiligen Testfall. Für eine neutrale Beurteilung des vorliegenden Szenarios dienen die vom MTCAS unabhängigen Testwerkzeuge zur Beobachtung.

Der durch LABSKAUS durchgeführte szenariobasierte Systemtest der Verhaltensprädiktion des MTCAS im Open-Loop-Verfahren führt zu dem Ergebnis, dass das MTCAS in allen Testfällen ein erfolgreiches Systemverhalten aufweist. Durch die in diesem Kapitel dargestellte Gestaltung des physikalischen Testfeldes LABSKAUS ist es vollumfänglich möglich, das Einsatzszenario des Testens im Open-Loop-Verfahren zu erfüllen. Die Anforderungen des Teilautomatisierungssystems für die automatisierte Schiffsführung hinsichtlich der operativen Umgebung sind ebenso durch LABSKAUS umsetzbar wie die Durchführung der für die Systementwicklung erforderlichen Testmethoden und die dynamische Integration des MTCAS zur Laufzeit ohne eine Modifikation der Testfeld-Infrastruktur.

### **6.3.3 Ergebnisse des Einsatzes für Open-Loop-Tests**

Mit der Hilfe von LABSKAUS ist es möglich, die notwendige Testumgebung für ein automatisiertes Schiffsführungssystem zur Durchführung eines szenariobasierten Systemtests im Open-Loop-Verfahren zu realisieren. Hierfür erfolgt die Bereitstellung der entsprechenden Stimulationsdaten für das zu testende System unter ähnlichen Bedingungen wie im vorherigen Einsatzszenario der explorativen Datenerhebung. Es dient der Testträger ZUSE und eine hierauf installierte mobile NaviBox für die Erfassung der Umgebungsdaten sowie Eigenschiffsdaten und der Bereitstellung entsprechender Teststimulationen an das zu testende System. Das einheitliche Datenmodell gemäß S-100 erweist sich wiederholt als die Interoperabilität fördernd, wie in Kapitel 6.2.3 diskutiert. Ebenso bietet die durch RabbitMQ implementierte nachrichtenorientierte Middleware eine hochkonfigurierbare Testfeld-Infrastruktur für die lose gekoppelte Integration und den Datenaustausch der verteilten Sensorsystemknoten. Dieses Evaluationsexperiment bestätigt die in Kapitel 6.2.3 beschriebenen Ergebnisse.

Als ein ergänzender Aspekt untersucht dieses Evaluationsexperiments die Fähigkeit der polymorphen Schnittstelle dynamisch zur Laufzeit ein zu testendes System so in LABSKAUS zu integrieren, das eine Änderung der Infrastruktur des Testfeldes nicht notwendig ist. Die polymorphe Schnittstelle zeigt hierbei die Fähigkeit ohne softwaretechnische Modifikation der Testfeld-Infrastruktur oder des zu testenden Systems eine entsprechende Schnittstelle zu bieten und einen interoperablen Datenaustausch bereitzustellen. Hierdurch passt sich LABSKAUS polymorph dem zu testenden System an und abstrahiert die internen Technologien des physikalischen Testfeldes zur Erhöhung der Generizität.

Die Anforderungen des MTCAS an die Latenz und die Datenübertragungsrate erfüllt die technische Infrastruktur von LABSKAUS. Das MTCAS wurde basierend auf Ethernet in LABSKAUS integriert und benötigte die Teststimuli einer mobilen NaviBox an Bord der

ZUSE mit einer Latenz von maximal einer Sekunde. Der Mittelwert der Latenz der Teststimulation während der Testdurchführung betrug 14 Millisekunden und maximal 532 Millisekunden. Die mobile NaviBox erzeugt eine konstante Datenübertragungsrate von circa 33 Kilobyte/s, die über die LTE Kommunikationsinfrastruktur drahtlos für die Testauswertung über LTE übertragen werden konnten, wie in der Leistungsdatenerhebung von LABSKAUS in Kapitel 6.1.2 beschrieben. Die temporär nicht ausreichende Datenübertragungsrate der LTE Kommunikationsinfrastruktur wird durch den Mechanismus der Warteschlange von RabbitMQ ausgeglichen. Diese Maßnahme vermeidet einen möglichen durch unzureichenden Netzausbau verursachten Datenverlust. Weitergehend erfüllen die stationären NaviBoxen zur Beobachtung die notwendige Datenübertragungsrate bei einer minimalen Übertragungsgeschwindigkeit über LTE von 2.226 Megabit/s wodurch das anfallende Datenaufkommen von je circa 4,2 Kilobyte/s uneingeschränkt an die zentrale nachrichtenorientierte Middleware übertragen wird.

Sämtliche datenerfassenden NaviBoxen übermitteln die Datenströme an die zentrale nachrichtenorientierte Middleware. Dies umfasst zum einen die Umweltparameter des Test szenarios und zum anderen die Teststimulation an das zu testende System MTCAS. Die übermittelten Daten enthalten somit neben den Teststimuli für das MTCAS ebenfalls die Beobachtungsdaten des vorliegenden Umweltszenarios des Tests zur weiterführenden Beurteilung der Testausführung. Hierdurch ist die Beobachtbarkeit des Systemverhaltens des zu testenden Systems in Korrelation zu dem vorliegenden Umweltszenario möglich. Für die Beobachtbarkeit der systeminternen Parameter des MTCAS existiert eine speziell implementierte Benutzeroberfläche, mit derer diese angezeigt und durch einen Systemexperten des MTCAS beurteilt werden. In Kombination mit der einheitlichen Datenmodellierung gemäß S-100 ist die ganzheitliche Beobachtung des Testszenarios möglich.

Für die Beurteilung der Konformität des MTCAS dienen die durch die Systementwickler definierten Testspezifikationen, welche die Testschritte und erwarteten Referenzwerte vorgeben. Durch LABSKAUS ist es möglich die notwendigen Evaluationsparameter der Eingabedaten und Ausgabedaten des zu testenden Systems zu erfassen und die korrekte Funktionsfähigkeit im Vergleich zwischen tatsächlichem und erwartetem Ergebnis der Systemreaktion zu beurteilen. LABSKAUS und die Testfeld-Architektur erfüllen die Anforderungen des Tests eines automatisierten Schiffsführungssystems im Open-Loop-Verfahren und bieten eine wiederverwendbare und generische Testumgebung.

## **6.4 Closed-Loop-Test eines hochautomatisierten Schiffsführungssystems**

Das OFFIS entwickelt in enger Kooperation mit der Universität Oldenburg verschiedene Systeme für die hochautomatisierte Führung von Schiffen. In diesem Zusammenhang wurde ein automatisiertes Schiffsführungssystem in Form eines kombiniert küstenseitig und seeseitig lokalisierten Systems entwickelt, welches die manuelle Steuerung und das automatisierte

Abfahren einer Route durch ein Schiff unter Berücksichtigung der Kollisionsvermeidung ermöglicht. Dieses automatisierte Schiffsführungssystem ist ein komplexer Systemverbund, denn es besteht aus den folgenden kooperierenden Teilsystemen.

- Ein Autopilot für die Bewegungsregelung und das Verfolgen einer Trajektorie des zu führenden Schiffes [AbFH15].
- Ein küstenseitiger Leitstand für die statische Routenplanung und Routenüberwachung implementiert in dem EPD.
- Eine schiffseitige Pfadplanung inklusive der Kollisionsverhütung.
- Eine statische Objekterkennung basierend auf digitalen Seekarten.
- Eine dynamische Objekterkennung basierend auf einem AIS, Radarsystem und einer optischen Kamera.

Diese Teilsysteme vereinen ihre Fähigkeiten, um als ein komplexer Systemverbund die automatisierte Schiffsführungsaufgabe zu übernehmen. Gemäß der in Kapitel 2.1 vorgestellten Taxonomie für automatisierte Schiffsführungsfunktionen, stellt dieser Systemverbund ein hochautomatisiertes Schiffsführungssystem dar. Hierfür visualisiert die Benutzeroberfläche EPD die unmittelbare Umgebung und Eigendaten des Schiffes durch Daten der entsprechenden Sensorik und maritimen Infrastruktur zur Überwachung der Umgebung und sendet Routen über eine drahtlose LTE Kommunikationsverbindung an die schiffseitigen Komponenten des automatisierten Schiffsführungssystems. Das schiffseitige Pfadplanungssystem detektiert statische und dynamische Objekte und erstellt einen geeigneten und sicheren zu folgendem Pfad. Hierfür berechnet das Pfadplanungssystem den Pfad für einen Streckenabschnitt unter Berücksichtigung der vorgegebenen Route des küstenseitigen Leitstands sowie der Kollisionsverhütung und sendet die berechneten Wegpunkte und eine Zielgeschwindigkeit an den Autopiloten. Der Autopilot berechnet die für die Bewegungsregelung notwendigen Steuerbefehle auf der Basis der durch Sensorik wahrgenommenen Schiffsdynamik, sowie der hydrodynamischen Umgebung.

Für die Erprobung dieses komplexen automatisierten Schiffsführungssystems, bietet sich ein Test im Closed-Loop-Verfahren an. Daher nutzt das folgende Einsatzszenario von LABSKAUS eine Simulation für die Steuerung des Testszenarios, um die benötigte Umweltsituation zur Erprobung von Schiffsbegegnungen gemäß der Testspezifikation in der Realwelt sicher bereitstellen zu können. In diesem Zusammenhang ist durch die Integration eines virtuellen und physikalischen Testfeldes die Komplementierung der real vorherrschenden Situation unter sicheren Testausführungsbedingungen möglich. Hierfür werden durch eine Simulation erzeugte virtuelle Schiffe gemäß dem bereits in den Verfahren MIL, SIL und HIL durchgeführten Testszenario verwendet, statt der Herleitung der Schiffsbegegnungen für die Kollisionsvermeidung mit realen Schiffen. Diese Testmethode vereint die Effizienz der zuvor virtuell erprobten Testszenarios durch eine nahtlose Überführung des zu testenden Systems für den Test in der Realwelt.

### 6.4.1 Zielsetzung

Das Kapitel 6.3 untersucht die Anforderungserfüllung der Testfeld-Architektur im Einsatzszenario des Tests im Open-Loop-Verfahren durch LABSKAUS. Bereits dieser Test im Open-Loop-Verfahren erweitert den Anforderungsraum entgegen des in Kapitel 6.2 vorgestellten Einsatzszenarios einer explorativen Datenerhebung. Der in diesem Kapitel vorgestellte Test im Closed-Loop-Verfahren vergrößert diesen erneut und stellt die vollumfängliche Anforderungsabdeckung der in Kapitel 2.3 aufgestellten Anforderungen an ein physikalisches Testfeld dar. In diesem Evaluationsexperiment erfolgt die Unterstützung der Systementwicklung des hochautomatisierten Schiffsführungssystems des OFFIS durch die nahtlose Überführung des zu testenden Systems von durchgeführten frühen MIL bis zu Tests in der Realwelt. Diese simulativen Tests fanden im Vorfeld während der Systementwicklung mit der Hilfe des in Kapitel 4.4.2 vorgestellten virtuellen Testfeld HAGGIS statt. Dieses Kapitel beschreibt den nahtlos an die simulative Erprobung anknüpfenden Systemtest mit der Hilfe von LABSKAUS, in dem das zu testende System gemäß der in HAGGIS genutzten inhärenten Schnittstelle und des dort definierten Testszenarios in die Realwelt überführt und dort getestet wird (A15).

Um in der realen Umgebung effizient und sicher das Kollisionsverhütungsszenario zu erproben reichert das virtuelle Testfeld den physikalischen Testfeld-Aufbau von LABSKAUS durch simulierte Fremdschiffe als dynamische Hindernisse an (A17). Auf die Abweichungen des vorherrschenden Realweltszenarios entgegen der Testspezifikation muss in der Simulation des virtuellen Testfeldes reagiert werden. Hierfür erfolgt die Rückkopplung der vorherrschenden Umweltbedingungen in das virtuelle Testfeld (A18). Das hochautomatisierte Schiffsführungssystem als zu testendes System empfängt nicht nur Umgebungs- und Eigenschiffdaten, sondern erzeugt im Regelkreis die Steuerbefehle an die Aktoren des Testträgers (A3).

Da das zu testende System des OFFIS ein Hochautomatisierungssystem ist, das hochfrequente Sensordatenströme verarbeitet, stellt dieses Evaluationsexperiment die Anforderung an die drahtlose und drahtgebundene Kommunikationsinfrastruktur von LABSKAUS hinsichtlich der notwendigen bilateralen Latenz zwischen der Datenerfassung und Bereitstellung von maximal einer Sekunde (A5). Für die Testdurchführung dienen die mobile NaviBox als Teststimuli sowie zwei stationäre NaviBoxen für die Beobachtung der Testdurchführung. Hierbei erzeugt die auf NMEA 0183 beziehungsweise NMEA 2000 basierende Sensorik der mobilen NaviBox eine maximale Datenübertragungsrate von 250 Kilobyte/s, die als Teststimuli an das System des OFFIS übertragen werden. Weitergehend erzeugen die drei auf NMEA 0183 beziehungsweise NMEA 2000 basierenden stationären NaviBoxen eine über drahtlose Kommunikationsinfrastruktur des Testfeldes zu übertragende Datenübertragungsrate von maximal 750 Kilobyte/s (A6, A4).

Für die Fehlertoleranz und Notfallmechanismen im Falle einer unvorhergesehenen und womöglich sicherheitskritischen Situation muss die Gestaltung des Testträgers einen sicheren Zustand sowohl für Ausrüstung wie Menschen und weiterer Verkehrsteilnehmer errei-

chen (A7). Das automatisierte Schiffsführungssystem benötigt vielfältige Daten unterschiedlicher Formate, wie optische Videodaten für die dynamische Objekterkennung und georäumliche sowie navigationsbezogene Daten durch ein AIS oder Radarsystem. Weiterführend sendet das automatisierte Schiffsführungssystem die Steuerbefehle des Schiffes an den Testträger (A24). Wie in dem in Kapitel 6.3 beschriebenen Einsatzszenario integriert die polymorphe Schnittstelle das zu testende System und stellt somit die Fähigkeit der Generalisierbarkeit durch die Entkopplung von LABSKAUS und dem zu testenden System dar. Im Rahmen des in diesem Kapitel beschriebenen Einsatzszenarios ist die Erprobung der Robustheit und Zuverlässigkeit des hochautomatisierten Schiffsführungssystems des OFFIS notwendig, da es ein sicherheitskritisches System ist. Daher findet die Manipulation der Datenströme statt, um die jeweiligen in den Testfällen definierten Fehlerparameter als Teststimulation an das zu testende System zu leiten. Das durch das Realszenario bereitgestellte Testszenario ist nicht ohne weiteres in der Lage die Fehlerzustände herzuleiten. Aus diesem Grund wird die steuerbare Infrastruktur von LABSKAUS hierfür genutzt (A16).

Die gemäß den normativen Standards geforderte Dokumentation der Testausführung umfasst nicht nur die Ausführung des Tests und eine Aussage über den Erfolg. Vielmehr erfolgt die nachvollziehbare Speicherung sämtlicher Beobachtungsdaten des Testszenarios des OFFIS Systems durch LABSKAUS (A13). Da eine oder mehrere Wiederholungen des Systemtests stattfinden und verglichen werden stellt dieses Evaluationsexperiment die Anforderung an das Testfeld, eine Reproduzierbarkeit durch die nachvollziehbare Dokumentation der Testausführungsdaten und des Umweltszenarios zu gewährleisten (A14). Dieses Evaluationsexperiment benötigt für die Beobachtung und Steuerung des Tests eine Benutzeroberfläche mit deren Hilfe die Testfeld-Infrastruktur gemäß dem Testszenario konfiguriert wird und die relevanten Testparameter während der Testdurchführung durch die Referenzwasserstraße beobachtet werden (A27).

Für das szenariobasierte Testen von hochautomatisierten Schiffsführungssystemen entlang des Systementwicklungsprozesses wird die in Abbildung 63 dargestellte Systemarchitektur eines Testaufbaus für das Testen im Closed-Loop-Verfahren verwendet. Das folgende Evaluationsexperiment ersetzt einen oder mehrere physikalische Sensoren, Aktoren oder Komponenten der maritimen Infrastruktur durch die Simulatoren des virtuellen Testfeldes, wie in Kapitel 2.2.4 erläutert. Die vom virtuellen Testfeld durch Sensorsimulation, Verkehrssimulation und Dynamiksimulation erzeugten Testdaten stimulieren das sich in der Referenzwasserstraße befindende zu testende System, welches kontinuierlich eine reagierende Ausgabe im Regelkreis erzeugt.

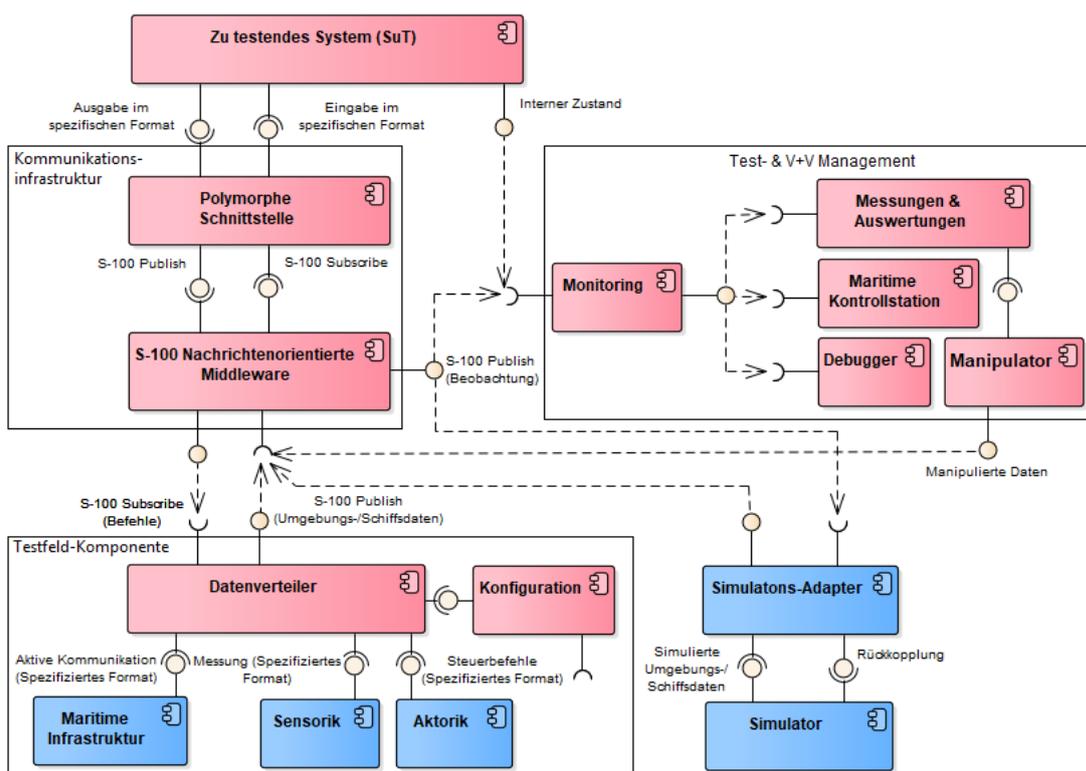


Abbildung 63: Aufbau des physikalischen Testfeldes für das dynamische Testen im Closed-Loop-Verfahren

Wie in dem in Kapitel 6.3.1 beschriebenen Aufbau des physikalischen Testfeldes für den Test im Open-Loop-Verfahren erfolgt die Beobachtung, Auswertung und Dokumentation der Testausführung durch das Monitoring des physikalischen Testfeldes. Hierfür stellt die zentrale nachrichtenorientierte Middleware mit einheitlichem Datenmodell gemäß S-100 die Ausgabedaten des zu testenden Systems sowie die Umgebungs- und Schiffsdaten der Referenzwasserstraße bereit. Die in der Abbildung 63 blau markierten Komponenten stellen wahlweise physikalische oder durch die Simulatoren des virtuellen Testfeldes ersetzte Komponenten dar. Neben der simulativen Anreicherung des Testträgers findet die gezielte Untersuchung des zu testenden Systems hinsichtlich der Robustheit und Zuverlässigkeit statt. Hierfür definiert eine Testspezifikation die Fehlerzustände, die durch eine manipulierbare Testfeld-Infrastruktur zur Unterstützung von Fehlerinjektionstests das zu testende System stimulieren.

## 6.4.2 Durchführung

Dieses Evaluationsexperiment eines Tests im Closed-Loop-Verfahren legt einen Schwerpunkt auf die dynamische Integration des hochautomatisierten Schiffsführungssystems des OFFIS in die Testfeld-Infrastruktur von LABSKAUS. LABSKAUS nutzt als Testträger das Schiff ZUSE um die vom zu testenden System des OFFIS benötigten Teststimuli bereitzustellen und im Regelkreis die Steuerbefehle auszuführen. Für diesen Zweck findet erneut die polymorphe Schnittstelle Verwendung, die entsprechende Schnittstellen zur Verfügung

stellt, um die Testfeld-Infrastruktur von dem zu testenden System zu entkoppeln. Dieses Einsatzszenario fokussiert weitergehend die Integration von LABSKAUS und dem virtuellen Testfeld HAGGIS, um die Herleitung der entsprechenden Kollisionsszenarien gemäß den spezifizierten Testszenarien durch virtuell repräsentierte Schiffe zu realisieren und somit die Realwelt durch LABSKAUS anzureichern. Der Test wurde am 12. Juli 2018 durchgeführt.

**Testfeld-Aufbau.** Bereits das Evaluationsexperiment des Tests im Open-Loop-Verfahren in Kapitel 6.3 nutzt die ZUSE als Testträger, um das zu testende System MTCAS über die polymorphe Schnittstelle dynamisch in die Testfeld-Infrastruktur zu integrieren. Während das MTCAS ein Teilautomatisierungssystem für die Kollisionsvermeidung gemäß der ersten Stufe der Taxonomie der automatisierten Schiffsführungsfunktionen darstellt, stellt das in diesem Kapitel erprobte hochautomatisierte Schiffsführungssystem des OFFIS weitere Anforderungen an LABSKAUS. Um diese zu erfüllen ist LABSKAUS wie in Abbildung 64 dargestellt aufgebaut.

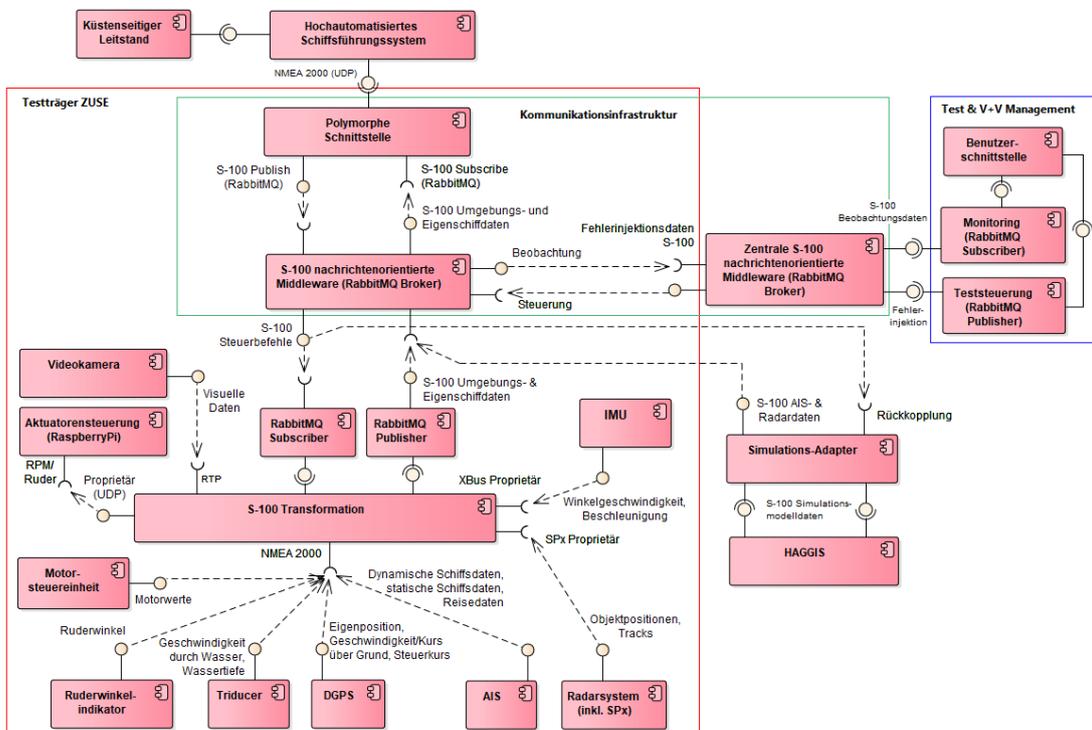


Abbildung 64: Testfeld-Aufbau von LABSKAUS für die Erprobung des hochautomatisierten Schiffsführungssystems des OFFIS im Closed-Loop-Verfahren

LABSKAUS setzt als Testträger das mit der mobilen NaviBox ausgerüstete Forschungsboot ZUSE ein, für die Bereitstellung der Daten und dem Empfang von Steuerbefehlen des zu testenden Systems. Die mobile NaviBox nimmt die Umgebung durch Sensorik wahr und stellt die für die Testdurchführung notwendigen Daten über die Warteschlange *PGMATE* der RabbitMQ Kommunikationsinfrastruktur als zentrale nachrichtenorientierte Middleware mit einheitlichem Datenmodell gemäß S-100 bereit.

Neben den in Kapitel 6.2.2 und 6.3.2 vorgestellten Komponenten von LABSKAUS zur Wahrnehmung der Umwelt und der Eigenschiffdaten ergänzen eine IMU vom Typ MTi-10

des Herstellers Xsens und eine Motorsteuereinheit der Firma Volvo Penta mit der Bezeichnung Head Control Unit den Testaufbau. Die Motorensteuereinheit ist in der Lage die Motordrehzahl zu messen und über NMEA 2000 zu übertragen. Der Einplatinencomputer Raspberry Pi dient als Schnittstelle zwischen der mobilen NaviBox und dem Ruder sowie der Motorensteuereinheit über die Module für die Eingabe und Ausgabe. Für die Lenkung des Forschungsbootes ZUSE ist eine hydraulische Pumpe parallel zu dem hydraulischen Lenkrad installiert, um das Schiff elektronisch durch eine Pulsweitenmodulation als Analog-Digital-Umsetzer zu steuern. Für die dynamische Objekterkennung benötigt das zu testende System neben dem AIS und Radar weitergehend durch eine Kamera erfasste visuelle Daten der Umgebung. Hierfür ist eine Kamera in Fahrtrichtung auf die Meeresoberfläche und den Horizont ausgerichtet. Die visuellen Datenströme der Kamera wurden über das Real-Time Transport Protocol (RTP) und einer Publish-Subscribe Kommunikationsform gemäß der Testfeld-Architektur dem zu testenden System über eine zentrale Middleware zur Verfügung gestellt.

Das Motorensteuergerät wurde ergänzt, um analoge Signale von dem Raspberry Pi entgegenzunehmen und auszuführen. Der Raspberry Pi empfängt von der mobilen NaviBox die Steuerbefehle in Form von RPM und Ruderwinkel über das Kommunikationsprotokoll UDP. Abgesehen von den beschriebenen Änderungen liegt die gleiche Installation des in dem Kapitel 6.3.2 beschriebenen Aufbaus des Forschungsbootes ZUSE vor.

Der in Kapitel 5.5 beschriebene Simulations-Adapter kommt in diesem Evaluationsexperiment zum Einsatz, um das vorherrschende Szenario der Realwelt durch simulierte Daten der Fremdschiffe anzureichern. Der Simulations-Adapter nutzt für die Datenversorgung den gleichen RabbitMQ Kommunikationskanal wie die realen Sensordaten der mobilen NaviBox. Mit der Hilfe des Szenario-Editors von HAGGIS erfolgt die Definition des Testszenarios. Nach dem Start der Simulation von HAGGIS findet die Erzeugung und der Datenversand statischer und dynamischer Daten des AIS und Radarsystems an die zentrale nachrichtenorientierte Middleware von LABSKAUS statt.

Abbildung 65 illustriert die physikalische Verteilung von LABSKAUS und des zu testenden schiffseitigen und küstenseitigen Systemverbunds, dessen Komponenten jeweils auf einem küstenseitigen und schiffseitigen Laptop installiert sind. Das hochautomatisierte Schiffsführungssystem wird schiffseitig auf der ZUSE kabelgebunden über Ethernet in die Kommunikationsinfrastruktur von LABSKAUS angeschlossen. Da das hochautomatisierte Schiffsführungssystem des OFFIS in erster Linie ein schiffseitiges System mit Führungsfunktionalität ist, existiert ein wie in Kapitel 6.2.2 beschriebener Aufbau der Kommunikationsinfrastruktur ausgeprägt durch die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform.

Zwar besitzt der Systemverbund wie LABSKAUS einen küstenseitigen Leitstand, dieser ist jedoch ein Bestandteil des zu testenden Systems und besitzt für die dynamische Integration in LABSKAUS keine dedizierte Schnittstelle. Über die von LABSKAUS bereitgestellte LTE Kommunikationsinfrastruktur ist es dem schiffseitigen hochautomatisierten Schiffsführungssystem möglich, mit dem landseitig installierten küstenseitigen Leitstand Daten auszutauschen und somit als Systemverbund die Schiffsführungsaufgabe zu übernehmen. Die

polymorphe Schnittstelle ist über die zentrale nachrichtenorientierte Middleware implementiert durch RabbitMQ bilateral verbunden und dient der Kopplung von LABSKAUS mit dem zu testenden System.

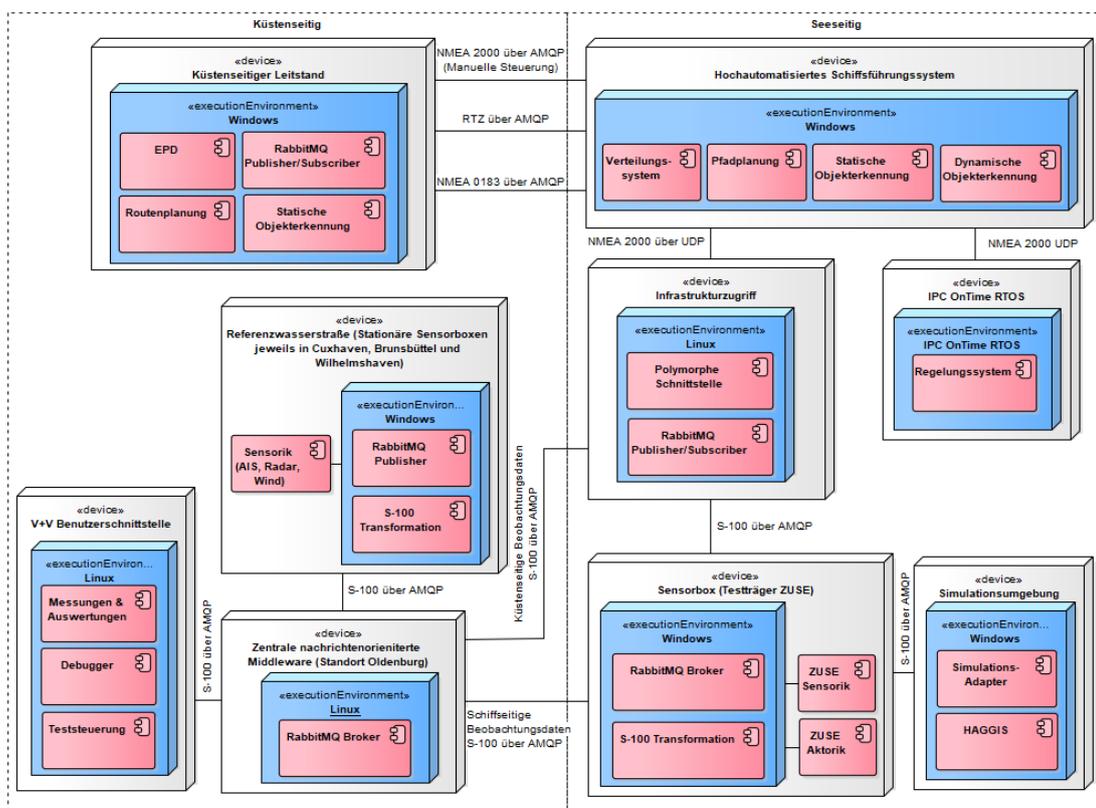


Abbildung 65: Physikalische Verteilung des szenariobasierten Systemtests des hochautomatisierten Schiffsführungssystems im Closed-Loop-Verfahren

Weitergehend senden die mobile NaviBox sowie die stationären NaviBoxen die gemessenen Daten an die zentrale nachrichtenorientierte Middleware für die Beobachtung. Ergänzend zu der in Kapitel 6.3.2 beschriebenen Installation findet der bilaterale Datenaustausch durch jeweils einem RabbitMQ Subscriber und Publisher statt, die auf der mobilen NaviBox implementiert sind.

Als Benutzerschnittstelle für die V+V dient die Java Anwendung MATEControl (Maritime Test and Experimental Platform Control), welche an die zentrale nachrichtenorientierte Middleware mit einheitlichem Datenmodell gemäß S-100 gekoppelt ist. Zum einen erfolgt das Monitoring durch den Empfang der Daten, zum anderen die Steuerung des Testszenarios zur Fehlerinjektion durch eine Komponente für die Manipulation der Datenströme. Die Benutzeroberfläche für die V+V speichert sämtliche Beobachtungsdaten des Testszenarios in dem binären Datenformat Base64 für die Dokumentation, Replikation und weitergehende Analyse. Die Kommunikationsinfrastruktur ergänzt entgegen der vorherigen Evaluations-experimente die Teststeuerung für die Fehlerinjektion zur Erprobung der Robustheit und Zuverlässigkeit des zu testenden Systems. Dies geschieht ebenfalls über die Benutzeroberfläche für die V+V implementiert durch MATEControl, die eine entsprechende Funktionalität auf der Basis des einheitlichen Datenmodells gemäß S-100 durch Eingabemöglichkeiten für einen Systemtester bereitstellt.

**Integration des zu testenden Systems.** Abbildung 66 illustriert die Konfiguration der polymorphen Schnittstelle für die dynamische Integration des hochautomatisierten Schiffsführungssystems des OFFIS. Die inhärente Schnittstelle dieses Systems besteht aus einer schiffseitigen, durch NMEA 2000 spezifizierten bilateralen Schnittstelle und dem Netzwerkprotokoll UDP. Über diese Schnittstelle findet sowohl die Datenextraktion, sowie das Senden von Steuerbefehlen statt. Die zwei Pipelines für die Transformation der bilateralen Datenströme benötigen jeweils fünf Filterkomponenten, welche über die Konfigurationsdatei in XML definiert werden.

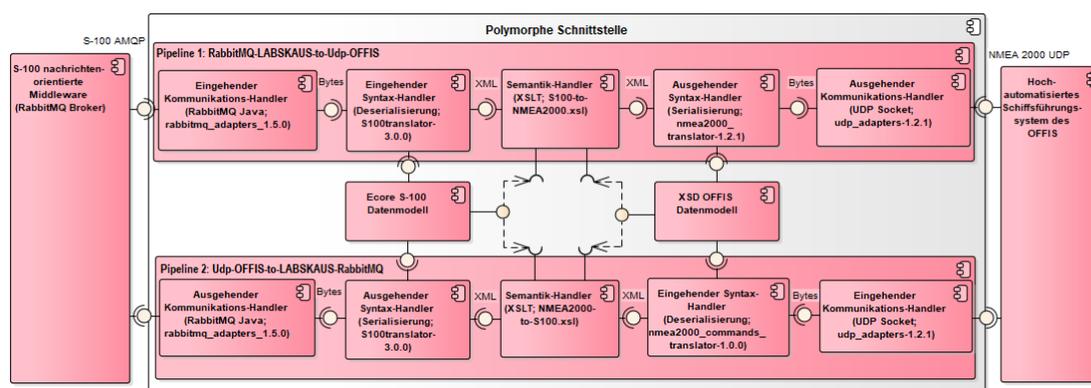


Abbildung 66: Konfiguration der polymorphen Schnittstelle für die dynamische Integration des OFFIS Systems

Die eingehenden Kommunikations-Handler und Syntax-Handler sind gemäß den in Kapitel 5.4 beschriebenen Implementierungen wiederverwendbar, da sich die Infrastruktur von LABSKAUS nicht ändert. Für die Datenextraktion existiert die RabbitMQ Warteschlange *PGMATE*, die von dem eingehenden Kommunikations-Handler abonniert wird. Hierüber werden die Daten der mobilen NaviBox des Testträgers bezogen. Der Syntax-Handler deserialisiert die S-100 Datenströme und überführt sie in eine XML Repräsentation für die durchzuführende semantische Transformation.

Der Semantik-Handler, der die strukturelle und semantische Transformationsaufgabe von dem einheitlichen Datenmodell gemäß S-100 in NMEA 2000 vornimmt, wird entgegen des Evaluationsexperiments des MTCAS modifiziert. Durch die polymorphe Schnittstelle ist die flexible Ausführung eines in XSLT vorliegenden Transformationsregelsatzes durch den Semantik-Handler möglich. Hierfür erfolgte zunächst die semantische Annotation der zwei unterschiedlichen Datenmodelle, um auf dieser Basis ausführbare Transformationsregelsätze in XSLT zu erzeugen. Die semantischen Kollisionen der zwei unterschiedlichen Datenmodelle waren ohne Informationsverlust transformierbar. Für die ausgehende syntaktische Transformation dient die Java Programmibliothek *com.thoughtworks.xstream*, mit deren Hilfe eine Serialisierung in XML erfolgt. *XStream* ermöglicht das Instanzieren der NMEA 2000 konformen Java-Klassen in XML und umgekehrt, gemäß der Schnittstelle des hochautomatisierten Schiffsführungssystems des OFFIS. Dieses gibt ebenso das Netzwerkprotokoll UDP vor, so dass der ausgehende Kommunikations-Handler die transformierten Datennachrichten über eine Implementierung der Klassen des Paketes *java.net.DatagramSocket* an das zu testende System sendet.

Eine zweite Pipeline existiert für die Steuerbefehle, welche das Forschungsboot ZUSE als Testträger durch die bereitgestellte Aktorik ausführt. Sämtliche Filterimplementierungen mit Ausnahme des Semantik-Handler werden wiederverwendet und als eigene datenverarbeitende Instanzen in der zweiten Pipeline instanziiert, wie in Abbildung 66 illustriert. Der Semantik-Handler umfasst einen XSLT Transformationsregelsatz, der die vom zu testenden System gesandten Steuerbefehle für RPM und den Ruderwinkel in NMEA 2000 modelliert entgegennimmt und semantisch in das einheitliche Datenmodell gemäß S-100 überführt. Anschließend veröffentlicht der ausgehende Kommunikations-Handler die S-100 Datenströme auf der RabbitMQ Warteschlange *PGMATE\_Steuerung*, welche durch die zentrale nachrichtenorientierte Middleware und die mobile NaviBox abonniert wird.

**Testmethoden.** Der durch LABSKAUS realisierte Test des hochautomatisierten Schiffsführungssystems des OFFIS stellt einen anforderungsbasierten Systemtest im Feld dar, anhand dessen die funktionale Korrektheit des zu testenden Systems gemäß der zuvor von den Systementwicklern identifizierten Testfällen in den Testszenarien möglich ist. Dieser Test erfolgt in dynamischer Ausführung des OFFIS Systems unter operativen Bedingungen. Im Vorfeld erfolgte eine simulationsunterstützte Systementwicklung mit der Hilfe des virtuellen Testfeldes HAGGIS. Der in diesem Kapitel beschriebene Test überführt das zu testende System nahtlos in eine Realweltumgebung, in der die simuliert erprobten Testszenarien physikalisch ausgeführt werden. Die Durchführung umfasst die fünf Testszenarien des automatisierten Abfahrens einer Route und die küstenseitige Überwachung, ein Überholmanöver durch das automatisiert zu führende Schiff, ein Fremdschiff mit kreuzendem Kurs von rechts und von links, sowie ein sich entgegengesetzt näherndes Fremdschiff. Anhand dieser fünf Testszenarien wird die Erfüllung von 126 Testfällen inklusive der Fehlerinjektionstests überprüft.

Zunächst findet ein Systemintegrationstest des hochautomatisierten Schiffsführungssystems des OFFIS und LABSKAUS statt. Hierdurch erfolgt die Verifikation der korrekten Funktionsweise der polymorphen Schnittstelle hinsichtlich der erfolgten Kommunikationsverbindung und syntaktischen Korrektheit der Daten. Nach dem erfolgreichen Verbindungsaufbau und einem interoperablen Datenaustausch beurteilt ein Systemexperte des OFFIS Systems die semantische Korrektheit der empfangenen Live-Daten der mobilen NaviBox von LABSKAUS. Die Plausibilität der empfangenen Daten wird anhand des vorliegenden Szenarios der Realwelt durch den Systemexperten validiert.

Nach dem erfolgreichen Systemintegrationstest findet ein funktionaler Systemtest statt, bei dem das OFFIS System im operativen Betrieb gemäß den Testszenarien ausgeführt wird. Entgegen der in Kapitel 6.3.2 beschriebenen manuellen Nachstellung der spezifizierten Situationen der Schiffsbegegnung erfolgt das Bereitstellen der Sensordaten für die Fremdschiffe im Closed-Loop-Verfahren durch die Simulation von HAGGIS. Das Forschungsboot ZUSE als Testträger erfasst durch die mobile NaviBox die gemäß dem jeweiligen Testfall relevanten Umgebungs- und Eigenschiffdaten ohne Kenntnis über deren Herkunft aus der Realwelt oder Simulation zu besitzen. Da das hochautomatisierte Schiffsführungssystem ein sicherheitskritisches System ist erfolgt dessen Überprüfung hinsichtlich der Zuverlässigkeit und

Robustheit durch das Provozieren von Fehlerzuständen und der Bereitstellung der resultierenden Daten als Teststimulation. Hierfür dient die steuerbare Kommunikationsinfrastruktur von LABSKAUS, deren Datenströme mit der Hilfe der Teststeuerungskomponente als Bestandteil der MATEControl die entsprechende Benutzeroberfläche für Fehlerinjektionstests bietet.

Während der Durchführung des Testszenarios und der Funktionsausführung des OFFIS Systems überprüft und beurteilt ein Systementwickler die funktionale Korrektheit des zu testenden Systems. MATEControl als Benutzeroberfläche für V+V bietet zum einen die Möglichkeit der Beobachtung des zu testenden Systems im Black-Box-Test und weitergehend eine spezifische Schnittstelle für die systeminterne Logik des OFFIS Systems. Hierdurch ist die tiefgehende Analyse im Rahmen eines Grey-Box-Tests möglich. Neben dieser Beobachtung des zu testenden Systems beobachtet die Benutzeroberfläche des Monitorings MATEControl die ganzheitliche Testdurchführung durch die stationären NaviBoxen der Referenzwasserstraße und den Zustand der Testfeld-Infrastruktur von LABSKAUS.

**Testdurchführung.** Der Systemintegrationstest des hochautomatisierten Schiffsführungssystems des OFFIS verwendet das Forschungsboot ZUSE als Testträger in dem dieses das Umweltlagebild im Hafengebiet und die Eigenschiffdaten an das OFFIS System übermittelt. Ähnlich wie in Kapitel 6.3.2 beschrieben, lässt sich mit der Hilfe von MATEControl als Benutzeroberfläche für V+V die funktionsfähige Kommunikationsverbindung sowie syntaktische und strukturelle Korrektheit der eingehenden Daten verifizieren und auftretende Fehler protokollieren. Für den Systemintegrationstest dienen die Datensätze der mobilen NaviBox, die LABSKAUS fehlerfrei dem OFFIS System über die polymorphe Schnittstelle zur Verfügung stellt. Ein Experte des OFFIS Systems validiert anschließend die erfolgreiche Integration durch die Überprüfung der semantischen Korrektheit der Daten durch einen Vergleich mit dem vorliegenden Realweltszenario in der Referenzwasserstraße.

Im Anschluss findet der Systemtest in dem durch die Referenzwasserstraße erfassten Bereich der Innenjade statt. Die mit dem Szenario-Editor von HAGGIS spezifizierten Testszenarien beschreiben die durchzuführenden Schiffsbegegnungen sowie die notwendigen Routen der Schiffe, wie für das Testszenario eines von rechts kreuzenden Fremdschiffes in Abbildung 67 illustriert. Die Zielroute des eigenen Schiffes ZUSE ist südlich Richtung Norden zwar in HAGGIS definiert, die Daten werden jedoch nicht an die zentrale nachrichtenorientierte Middleware von LABSKAUS gesendet. Der Simulations-Adapter überträgt lediglich die Daten des von rechts Richtung Süden kreuzenden Fremdschiffes, so dass die Eigenschiffdaten der ZUSE ausschließlich in der Simulation von HAGGIS existieren. Die Definition der Route des Forschungsbootes ZUSE dient der Reaktion durch die Simulation.

Das Testszenario beginnt an dem aktuellen Startpunkt des Forschungsbootes ZUSE, in dem die Simulation gestartet und die Testfälle zur Überprüfung der korrekten Funktionalität OFFIS Systems in einem Test im Closed-Loop-Verfahren durchgeführt werden. Das OFFIS System wird in den fünf Testszenarien eingesetzt und folgt der gesetzten Route. Während-

dessen führt das OFFIS System verschiedene Manöver der Kollisionsverhütung unter Beobachtung durch den küstenseitigen Leitstand durch, nachdem die Objektidentifikation das Umweltlagebild erzeugt.

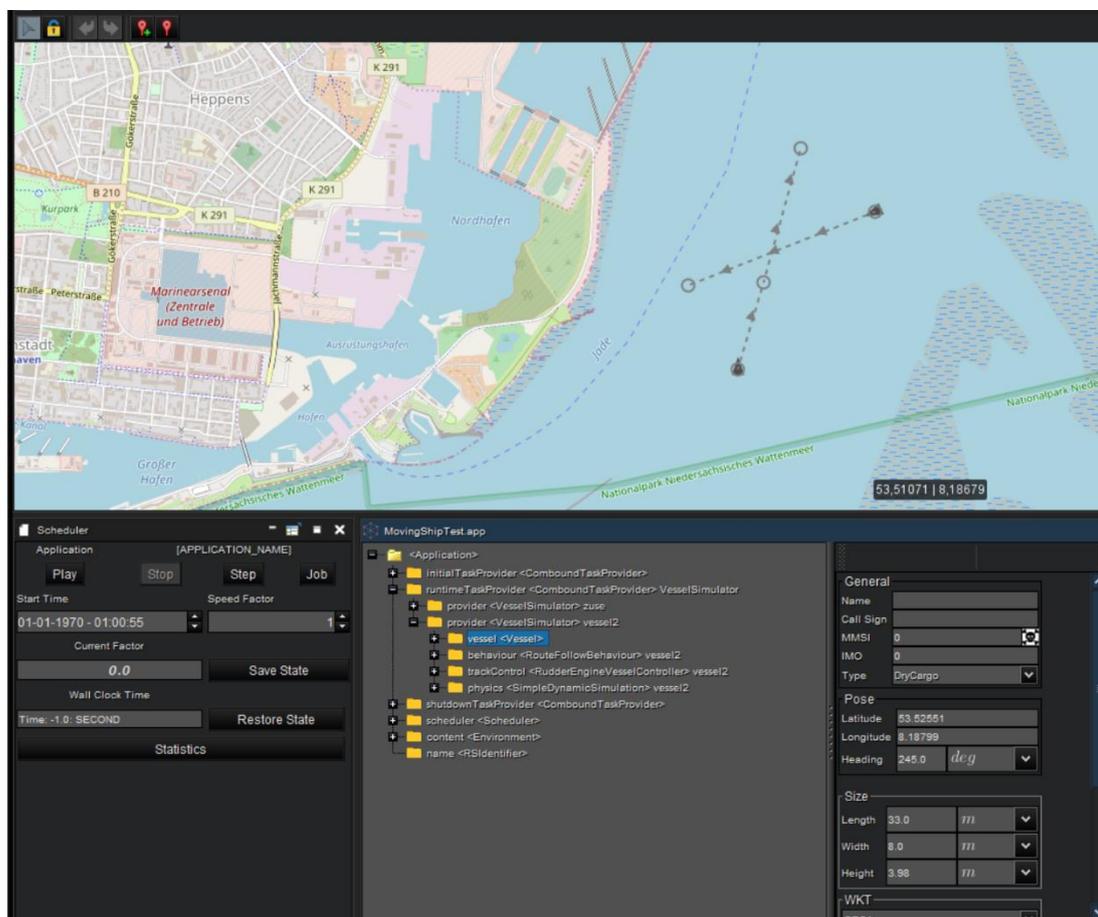


Abbildung 67: Definition eines Testszenarios in HAGGIS zur Erprobung des hochautomatisierten Schiffsführungssystems des OFFIS während einer Schiffsbegegnung mit kreuzenden Kursen

Zum einen beurteilen die Systementwickler und Testbeauftragten die Testausführung hinsichtlich der korrekten Funktionalität durch das zu testenden System, zum anderen dient für die Beobachtung der Testdurchführung die Benutzeroberfläche von MATEControl als Werkzeug für die V+V. Diese in Abbildung 68 dargestellte Benutzeroberfläche abonniert die zentrale nachrichtenorientierte Middleware und beobachtet somit zum einen die durch den Testträger produzierten Daten sowie weitergehend die ganzheitlichen Beobachtungsdaten des Testszenarios durch die stationären NaviBoxen. Die Benutzeroberfläche nimmt durch das Abonnieren des zentralen RabbitMQ Broker keinen Einfluss auf die Testdurchführung, da eine lose Kopplung der nachrichtenorientierten Middleware mit Publish-Subscribe-Kommunikationsform vorliegt. Nach dem Empfang der Daten erfolgt die Aufbereitung dieser für die Live-Darstellung. Zum Zwecke der Dokumentation und Reproduzierbarkeit sowie der nachgelagerten Analyse eventuellen Fehlverhaltens, findet die Speicherung sämtlicher Beobachtungsdaten durch MATEControl im Byte64 Format statt. Somit ist das durchgeführte Testszenario des OFFIS Systems im Nachgang reproduzierbar und ermöglicht die weitergehende Analyse.

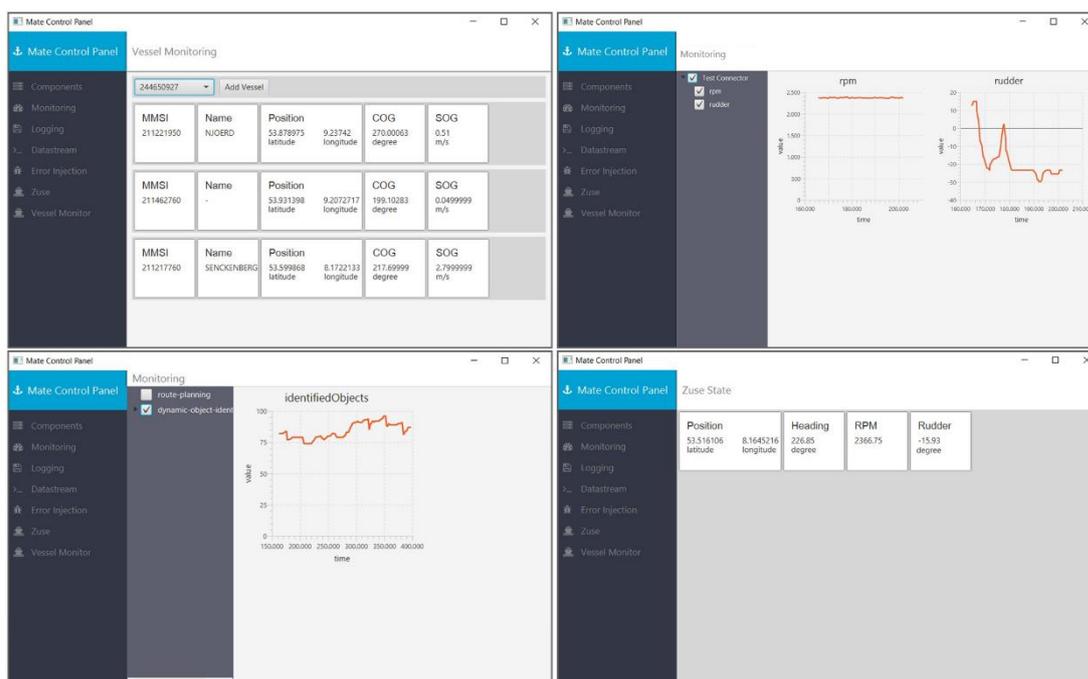


Abbildung 68: Ausschnitte aus der Benutzeroberfläche für V+V MATEControl für die Beobachtung und Steuerung des physikalischen Testfeldes LABSKAUS. Oben links: Beobachtung der Parameter der erfassten Schiffe in der Referenzwasserstraße. Unten links: Beobachtung des internen Systemverhaltens der dynamischen Objekterkennung des OFFIS Systems für einen Grey-Box-Test. Oben rechts: Beobachtung des Verhaltens der zu steuernden Aktuatoren des Forschungsbootes ZUSE über Zeit. Unten rechts: Beobachtung ausgewählter Testparameter des Forschungsbootes ZUSE in Echtzeit.

Abbildung 69 zeigt exemplarisch die Durchführung eines Testszenarios, in dem das hochautomatisierte Schiffsführungssystem das Forschungsboot ZUSE gemäß einer vorgegebenen Route führt und statischen sowie dynamischen Hindernissen ausweicht. Unter anderem dienen während der Testdurchführung, die durch HAGGIS erzeugten virtuellen Fremdschiffe in Kombination mit dem erfassten realen Schiffsverkehr als dynamische Hindernisse, denen gemäß den Kollisionsverhütungsregeln ausgewichen wird. Das Forschungsboot ZUSE erfasst über das simulierte AIS und Radarsystem die auszuweichenden Fremdschiffe und komplementiert das Umweltszenario durch das auf der ZUSE verbaute AIS, Radarsystem, den Ruderwinkel-Indikator, Triducer, die IMU, Motorsteuereinheit und das DGPS.

Auf der Basis dieser Sensordaten als Teststimulation erfolgt die Berechnung der Trajektorien und automatisiert erzeugten Befehle an die Aktorik des Forschungsbootes ZUSE, das im Regelkreis automatisiert durch das OFFIS System gesteuert wird. Mit der Hilfe des küstenseitigen Leitstandes ist das Erstellen und Senden der Routen an das schiffseitige System möglich und die anschließende Führung des Schiffes sowie die Umgebung visuell beobachtbar. Der küstenseitige Leitstand ist nicht direkt mit LABSKAUS verbunden, sondern erhält die jeweiligen Daten von dem schiffseitig verorteten Führungssystem als ein Bestandteil des Systemverbundes.



Abbildung 69: Hochautomatisiertes Schiffsführungssystem des OFFIS während des szenariobasierten Systemtests im Closed-Loop-Verfahren. Ein Experte des OFFIS Systems überprüft die korrekte Funktionserfüllung, während der die notwendigen Teststimuli und das Videokamerabild in Fahrtrichtung bereitstellende Forschungsboot ZUSE durch das zu testende System automatisch gemäß einer von einem küstenseitigen Leitstand bereitgestellten Route unter Berücksichtigung der Kollisionsvermeidung geführt wird. Während des Abfahrens der Route passierte das Passagierschiff HARLE KURIER die Route (links). Ausschnitt des küstenseitigen Leitstands für die Überwachung des automatisierten Abfahrens der vorgegebenen Route (rechts). Die gestrichelte graue Linie mit Pfeilen ist die Soll-Route und die durchgezogene graue Linie die tatsächlich abgefahrte Route.

Für die Manipulation der Datenströme im Rahmen des Fehlerinjektionstests wird die Benutzeroberfläche von MATEControl als Testwerkzeug verwendet. Diese abonniert die nachrichtenorientierte Middleware von LABSKAUS um die originalen Datenströme zu erhalten. Anschließend manipuliert und versendet MATEControl die im einheitlichen Datenmodell gemäß S-100 modellierten Daten über die nachrichtenorientierte Middleware von LABSKAUS an das zu testende System. Ein Systemexperte gibt die Fehlerparameter gemäß dem Testfall in MATEControl ein und stimuliert hierdurch das OFFIS System durch die gesteuerte Testfeld-Infrastruktur.

### 6.4.3 Ergebnisse des Einsatzes für Closed-Loop-Tests

Der Einsatz von LABSKAUS ermöglicht die Durchführung eines szenariobasierten Systemtests im Closed-Loop-Verfahren. Für die Bereitstellung der entsprechenden Testumgebung in dem Evaluationsszenario dient ein ähnlicher Testaufbau, wie in dem Kapitel 6.3 beschriebenen Systemtest des Teilautomatisierungssystems MTCAS. Das Forschungsboot ZUSE und eine mobile NaviBox stellen die Umgebungsdaten und Eigenschiffdaten als Teststimulation an das OFFIS System bereit. Dieses Evaluationsexperiment bestätigt die in Kapitel 6.2.3 und Kapitel 6.3.3 beschriebenen Ergebnisse der Anforderungserfüllung durch die Testfeld-Architektur.

Die Integration der Aktorik zur Ausführung der vom zu testenden System gesandten Steuerungsbefehle gemäß der Testfeld-Architektur ergänzt die Untersuchung des Lösungsansatzes in diesem Evaluationsexperiment. Hierbei erweist sich die Testfeld-Architektur als flexibel erweiterbar, indem in die mobile NaviBox als Testfeld-Komponente die entspre-

chende Aktorik-Einheit integriert und Datenbefehle über die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform und dem einheitlichen Datenmodell gemäß S-100 übertragen werden. Hierbei lässt sich insbesondere das einheitliche Datenmodell gemäß S-100 zweckdienlich erweitern, während die weiteren Testfeld-Komponenten von LABSKAUS nach wie vor einsatzfähig sind.

LABSKAUS stellt drahtlose und drahtgebundene Kommunikationsinfrastruktur zur Verfügung, um dem zu testenden System die operative Umgebung zu bieten. Hierbei nutzt der küstenseitige Leitstand die LTE Kommunikationsinfrastruktur von LABSKAUS, um dem schiffseitigen Führungssystem die Routen zu senden und die anschließende automatisierte Schiffsführung zu überwachen. Weitergehend bietet LABSKAUS die notwendige LTE Kommunikationsinfrastruktur, um die verteilten seeseitigen und landseitigen Sensorsystemknoten zu verbinden und eine ganzheitliche Testbeobachtung und Steuerung des Tests zu ermöglichen. Die stationären NaviBoxen erzeugen je Sensorsystemknoten eine kontinuierliche Datenübertragungsrate von durchschnittlich 5 Kilobyte/s, während die mobile NaviBox eine durchschnittliche Datenübertragungsrate von 33 Kilobyte/s erzeugt. Die von den NaviBoxen generierten Daten sind im einheitlichen Datenmodell gemäß S-100 modelliert.

Die für die drahtlose Übertragung eingesetzte LTE Kommunikationstechnologie besitzt eine minimale Datenübertragungsrate für den Upload von 2.226 Megabit/s der stationären NaviBox in Brunsbüttel und 9.773 Megabit/s der stationären NaviBox in Cuxhaven, wie in Kapitel 6.1.2 beschrieben. Die mobile NaviBox weist über die LTE Kommunikationstechnologie eine minimale Datenübertragungsrate von 0,15 Megabit/s auf. Dies unterschreitet in wenigen Randfällen der Messung die notwendige Datenübertragungsrate der mobilen NaviBox. Ähnlich wie in Kapitel 6.3.3 beschrieben gleicht der Mechanismus der Lebensdauer von RabbitMQ dies kurzzeitig aus, so dass sämtliche Daten an die zentrale nachrichtenorientierte Middleware übertragen werden können. LABSKAUS ist somit in der Lage, das anfallende Datenaufkommen der stationären und mobilen NaviBoxen an das Zielsystem und die zentrale nachrichtenorientierte Middleware zu übertragen. Die Teststimulation ist hiervon nicht betroffen, da diese über eine kabelgebundene Verbindung stattfindet.

Das automatisierte Schiffsführungssystem ist ein sicherheitskritisches System und besitzt die Anforderung an eine Latenz für die Datenbereitstellung von maximal einer Sekunde. Die Latenz der Teststimulation während der Testdurchführung durch physikalische Testfeld-Komponenten beträgt im Mittelwert 14 Millisekunden und maximal 532 Millisekunden, sowie für den Empfang der Steuerbefehle 12 Millisekunden und maximal 26 Millisekunden. Die virtuell von der Sensorsimulation von HAGGIS erzeugten Daten des AIS und Radars der Fremdschiffe weisen eine Latenz von durchschnittlich 22,8 Millisekunden auf. Somit erfüllt LABSKAUS die Anforderungen der Latenz des hochautomatisierten Schiffsführungssystems des OFFIS von maximal einer Sekunde.

Das zu testende System des OFFIS wurde außerhalb des Rahmens dieser Arbeit simulativ mit der Hilfe des virtuellen Testfeldes HAGGIS erprobt. Hierbei erfolgte zunächst ein MIL Test der Komponente für die in MATLAB implementierten Regelungsalgorithmen und die anschließende Überführung dieser in eine C++ Implementierung sowie die simulative Erprobung in einem SIL Test. Abschließend fand die Installation auf der Zielhardware und ein

simulativer HIL Test statt. Im Anschluss wurde das zu testende System nahtlos anhand der inhärenten Schnittstellen in das physikalische Testfeld überführt und in der Realwelt getestet. Hierbei wurde insbesondere das in HAGGIS definierte und durchgeführte simulative Testszenario verwendet, um dieses ebenfalls in der Realwelt im Feld durchzuführen und hinreichend auf eine korrekte Funktionalität des zu testenden Systems zu prüfen. Durch HAGGIS ist die Komplementierung des Umweltszenarios der Realwelt möglich, in dem AIS und Radar Daten simulativ erzeugt und an die nachrichtenorientierte Middleware von LABSKAUS gesandt werden. Durch die Verwendung des einheitlichen Datenmodells gemäß S-100 in dem virtuellen und physikalischen Testfeld ist die Integration dieser durch den Simulations-Adapter möglich. Das physikalische Testfeld zeigt sich durch diese Integration mit einem virtuellen Testfeld als steuerbar gemäß dem zu Grunde liegenden Testszenario, um insbesondere nicht in der Realwelt auftretende Phänomene dennoch unter operativen Bedingungen zu erproben. Durch die Rückkopplung der Daten von LABSKAUS in HAGGIS ist die Überwachung der simulativen Datenerzeugung und die Reaktion auf die in der Realwelt vorherrschenden Bedingungen möglich. Der Simulations-Adapter zeigt die Fähigkeit den bilateralen Datenaustausch für die Integration des virtuellen und physikalischen Testfeldes durchzuführen. Diese Integration bietet vielfältige Möglichkeiten für anknüpfenden Forschungsbedarf, in dem die Anreicherung des physikalischen Testfeldes durch simulative Daten für Tests im Closed-Loop Verfahren erweitert wird.

Die Testfeld-Architektur zeigt die Fähigkeit vielfältige Daten über die Kommunikationsinfrastruktur von LABSKAUS zu übertragen. Georäumliche und hydrodynamische Daten, maschinenbezogene Daten für die Steuerung des Schiffes und visuelle Daten für die dynamische Objekterkennung mit der Hilfe einer Kamera überträgt LABSKAUS gemäß der Publish-Subscribe Kommunikationsform an das zu testende System und für die Beobachtung. In dem Evaluationsexperiment wurde nicht die polymorphe Schnittstelle für die Übertragung der Kameradaten verwendet. Dies wäre durch den erweiterbaren und modularen Aufbau möglich, so dass im Rahmen zukünftiger Bemühungen die Integration von audiovisuellen Daten in die polymorphe Schnittstelle denkbar ist.

Für die Fehlertoleranz und Notfallmechanismen im Falle einer unvorhergesehenen und womöglich sicherheitskritischen Situation besitzt der Testträger einen Notastaster als ergänzendes Sicherheitselement. Nach dem Betätigen deaktiviert dieser die digitale Ansteuerung der Aktorik durch das zu testende System und ermöglicht die Steuerung des Testträgers über den werkseitigen Signalweg des Steuerrades und Gashebels durch einen an Bord anwesenden Kapitän als Steuerungersatz. Während der Testdurchführung ist im Falle eines unvorhergesehenen Verhaltens oder einer kritischen Verkehrssituation die Übersteuerung der Signale möglich.

Die Testfeld-Architektur ermöglicht das Manipulieren der Datenströme, um die in den Testfällen definierten Fehlerzustände zu provozieren und das Systemverhalten zu erproben. Hierfür nimmt das Werkzeug für V+V implementiert durch MATEControl durch die Publish-Subscribe Kommunikationsform lose gekoppelt an der Kommunikation über die nachrichtenorientierte Middleware von LABSKAUS teil und beeinflusst nicht die Testausführung. Zum einen übermittelt die zentrale Kommunikationsinfrastruktur die Beobachtungsdaten

für Werkzeuge der V+V, zum anderen ermöglichen diese Werkzeuge entweder das manuelle oder automatisierte Erzeugen beziehungsweise Manipulieren der Daten. Hierbei dient MATEControl in diesem Evaluationsexperiment als ein Werkzeug für die V+V, über welche das physikalische Testfeld gemäß dem Testszenario konfiguriert und gesteuert wird. MATEControl ist lose gekoppelt in die Testfeld-Infrastruktur integriert und beeinflusst nicht die Testausführung. Sämtliche erfassten Daten des Testszenarios speichert die zentrale nachrichtenorientierte Middleware und dokumentiert die ganzheitliche Testdurchführung in nachvollziehbarer Weise. Die Reproduzierbarkeit ist somit gegeben, denn neben den Teststimuli und Ausgabereaktionen des zu testenden Systems erfasst LABSKAUS die Umweltdaten und den Zustand der Testfeld-Infrastruktur.

## 6.5 Anforderungserfüllung und Zielabdeckung

Durch die Evaluation ist die Widerlegung der in Kapitel 6 aufgestellten Hypothesen nicht möglich. Die Experimentaufbauten, Methoden sowie Resultate sind ausführlich beschrieben. Die erste Hypothese zur Validierung der Anwendbarkeit und Darstellung der Eignung der Testfeld-Architektur für die V+V von heterogenen automatisierten Schiffsführungssystemen demonstriert der Einsatz von LABSKAUS erfolgreich. Mit der Hilfe der durch das physikalische Testfeld LABSKAUS bereitgestellten Testumgebung setzt die Testfeld-Architektur verschiedene repräsentative Einsatzszenarien für die Erprobung von automatisierten Schiffsführungssystemen entlang des Entwicklungsprozesses um.

Für die Nutzung von LABSKAUS in den Einsatzszenarien stellen Messungen der Kommunikationsinfrastruktur für die vernetzten Sensorsystemknoten von LABSKAUS die Leistung des physikalischen Testfeldes im Hinblick auf die Latenz und den Datendurchsatz der vernetzten Komponenten der verteilten physikalischen Testfeld-Infrastruktur dar. Hierdurch ist das Treffen einer quantitativen Aussage hinsichtlich der Anforderungserfüllung gemäß den Normen für digitale Schnittstellen eines automatisierten Schiffsführungssystems durch das physikalische Testfeld LABSKAUS möglich.

Die Einsatzszenarien der Testfeld-Architektur variieren nicht nur hinsichtlich der Systementwicklungsprozesse, sondern ebenfalls bezüglich der getesteten Systemkonzepte des Zielsystems und der hieraus resultierenden Anforderungen an die Testumgebung. Durch die Testfeld-Architektur findet die Erprobung sämtlicher Systemebenen eines automatisierten Schiffsführungssystems statt, die das MTCAS beziehungsweise hochautomatisierte Schiffsführungssystem des OFFIS implementieren. Das Potenzial einer offen und flexibel gestalteten physikalischen Testfeld-Architektur als generische Testumgebung, die durch die Systementwicklungsmethodik und Konzepte des Zielsystems fundiert ist, stellt diese Forschungsarbeit hinreichend dar. Die Untersuchung der zweiten Hypothese ist durch die Evaluationsexperimente hinsichtlich der Anforderungserfüllung möglich, wie in Anhang C dargestellt. Dies verifiziert die Erfüllung der Anforderungen durch die Testfeld-Architektur, die bisherige maritime Testaufbauten unzureichend erfüllen. Die Kapitel 6.2.3, 6.3.3 und 6.4.3 beschreiben natürlichsprachlich die Anforderungserfüllung unter der Bezugnahme auf die

Evaluationsexperimente. Insbesondere erfüllt die Testfeld-Architektur die von den gegenwärtigen Testaufbauten nicht erfüllten Anforderungen der Steuerbarkeit (KA4), Polymorphie (KA6), losen Kopplung (KA7) und Konfigurierbarkeit (KA8).

Neben der Leistungsdatenerhebung in Kapitel 6.1 und den in den Kapiteln 6.2.3, 6.3.3 und 6.4.3 berichteten detaillierten Ergebnissen der Anwendung von LABSKAUS, fokussiert sich diese Zusammenfassung der Evaluation auf die erreichte Zielerfüllung dieser Forschungsarbeit. Hierfür betrachtet dieses Kapitel die in Kapitel 1.3 definierten Teilfragen und hierfür notwendigen Ziele an einen Lösungsansatz und nimmt Bezug auf die Ergebnisse der Evaluation.

### **Teilfrage 1: Welche Technologiekonzepte und Systementwicklungsmethoden müssen in der Gestaltung des physikalischen Testfeldes berücksichtigt werden?**

Während der Evaluation erfolgen mehrere repräsentative Anwendungen von LABSKAUS für zu testende Systeme, denn sämtliche notwendige Funktionen eines automatisierten Schiffsführungssystems erfüllt die Testfeld-Architektur während des Testens. Die Testfeld-Architektur bewährt sich daher unter Berücksichtigung der notwendigen Technologien und Konzepte des automatisierten Schiffsführungssystems als das Zielsystem (Ziel 1). Diese Technologien und Konzepte basieren auf dem Drei-Schichten-Modell eines automatisierten Schiffsführungssystems und den hiermit einhergehenden Standards und Technologien für digitale Schnittstellen für die Schichten der Route, des Verkehrs und der Kontrolle. Für die Heterogenität und Komplexität des Systems und Interoperabilitäts Herausforderungen dient der auf der IEC 61850 basierende Lösungsansatz für konzeptionelle Interoperabilität des MAF unter anderem für automatisierte Schiffsführungssysteme. Mit dessen Hilfe ist die Harmonisierung der heterogenen Systeme des physikalischen Testfeldes und die Integration neuer Komponenten in die existierenden Strukturen der IT-Infrastruktur möglich.

Die Testfeld-Architektur dient in der Evaluation zur Unterstützung der etablierten Systementwicklungsmethodik des V-Modells durch die Anwendung für verschiedene Methoden der V+V. Hierbei unterstützt LABSKAUS durch die Beobachtungsfähigkeiten für die explorative Datenerhebung zur Bildung von Annahmen, Bedingungen und Modelle für die Definition und Gestaltung der Systeme. Weiterführend ermöglicht LABSKAUS durch eine beobachtbare und steuerbare IT-Infrastruktur Tests im Open-Loop und Closed-Loop Verfahren für die Konformitätsprüfung und Systemtests. Die durch die Methoden der Entwicklung, Validierung und Verifikation fundierte Testfeld-Architektur ermöglicht die Unterstützung während des gesamten Systementwicklungsprozesses, insbesondere für die Zuverlässigkeitsüberprüfung sicherheitskritischer Systeme (Ziel 2).

### **Teilfrage 2: Wie stellt sich eine wiederverwendbare Gestaltung durch die Systemarchitektur des physikalischen Testfeldes dar?**

Die nahtlose V+V ist möglich durch die Integration virtueller und physikalischer Komponenten entsprechend der Testfeld-Architektur durch ein einheitliches Datenmodell gemäß S-100. Insbesondere ermöglicht die Testfeld-Architektur für das szenariobasierte Testen einen nahtlosen Übergang zwischen der simulationsbasierten Erprobung bis zu Realwelttests

im Feld, in denen im Closed-Loop Verfahren eine steuerbare Testumgebung die komplexen Testszenerarien für automatisierte Schiffsführungssysteme realisiert (Ziel 3).

Die flexible Integration des zu testenden Systems durch die polymorphe Schnittstelle in die Infrastruktur des physikalischen Testfeldes ist möglich. Die polymorphe Schnittstelle von LABSKAUS ermöglicht die Anpassung der starren Testfeld-Infrastruktur an wechselnde zu testende Systeme, deren Schnittstellen auf etablierten Standards oder proprietären Spezifikationen basieren. Mit der Hilfe des Pipes und Filter Softwarearchitekturmusters als Bestandteil der polymorphen Schnittstelle ist die flexible Kombination von erweiterbaren Transformationskomponenten möglich. Diese dynamische Integration zur Laufzeit der Testausführung ermöglicht die Nutzung von LABSKAUS als nachhaltiges und generisches Testfeld für gegenwärtige und zukünftige Tests (Ziel 4).

LABSKAUS bietet die für die Einsatzszenarien notwendigen Komponenten und Systeme zur Bereitstellung der Testumgebung. Wegen der wechselnden funktionalen Anforderungen der zu testenden Systeme während der unterschiedlichen Anwendungen der Evaluation ermöglicht die Testfeld-Architektur durch eine lose Kopplung die Anpassung und Erweiterung der Testfeld-Infrastruktur für weitere die Testumgebung komplementierende Komponenten. Diese werden anhand des einheitlichen Datenmodells gemäß S-100 als transitiven Referenzpunkt und eine nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform in die Infrastruktur des physikalischen Testfeldes in interoperabler Weise integriert. Hierbei beeinflusst die Integration der erweiternden Testfeld-Komponenten nicht die existierenden Komponenten und Systeme des physikalischen Testfeldes (Ziel 5).

Diese Forschungsarbeit erreicht die Ziele und bietet Antworten auf die in Kapitel 1.3 aufgestellten wissenschaftlichen Fragestellungen durch die entwickelte Testfeld-Architektur.

## 7 Zusammenfassung und Ausblick

Das folgende Kapitel fasst diese ingenieurwissenschaftliche Arbeit in Kapitel 7.1 zusammen und bietet einen Ausblick auf potenzielle Zukunftsaufgaben in Kapitel 7.2.

### 7.1 Ergebnisse und Forschungsbeitrag

Diese anwendungsorientierte Forschungsarbeit bietet eine modellierte Systemarchitektur für ein nachhaltiges und wiederverwendbares physikalisches Testfeld, um die integrierte Entwicklung von automatisierten Schiffsführungssystemen zu unterstützen.

Das Forschungsprogramm der Bundesregierung „Maritime Technologien der nächsten Generation“ identifiziert die Förderungsnotwendigkeit der Entwicklung maritimer Technologien als strategisch bedeutsam für die gesamte maritime Wirtschaft. Die kooperative Aufgabenerfüllung eines automatisierten Schiffsführungssystems stellt wegen der Systemkomplexität beachtliche gestalterische und methodische Herausforderungen an die Verifikation und Validierung für die Sicherstellung der korrekten Funktionalität und insbesondere der funktionalen Sicherheit. Die Gestaltung dieser Technologien als offene Systeme vergrößert diese Herausforderungen, denn deren Komponenten sind als wechselnde Systemverbünde ausgeprägt und die Integration von intelligenten Systemen resultiert in einem nichtdeterministischen Verhalten. Das aktuell angewandte isolierte Testen einzelner Komponenten ist den zukünftigen Herausforderungen dieser komplexen Systeme nicht gewachsen, wie beispielsweise durch den Beschluss MSC.252(83)/IEC 61924-2 für das Testen der Komponenten eines integrierten Navigationssystems vorgeschlagen.

Ein physikalisches Testfeld trägt dazu bei die Verifikation und Validierung (V+V) während des gesamten Systementwicklungsprozesses zu fördern und ermöglicht die Erprobung des zu testenden Systems in einer simulativen oder physikalischen Testumgebung. Die Heterogenität der zu testenden Systeme und umfangreiche Anforderungen durch die Systementwicklungsmethodik für diese neuartigen Systeme stellen die Gestaltung der aktuellen maritimen Testaufbauten vor Herausforderungen. Diese sind individuell für spezifizierte Testszenarien ausgeprägt und nicht nachhaltig konstruiert, denn die Gestaltung folgt keiner strukturierten Methode. Um den Systementwicklern und Testingenieuren das effiziente Testen der automatisierten Schiffsführungssysteme zu ermöglichen, bietet diese Arbeit eine Antwort auf die Frage „Wie muss ein physikalisches Testfeld gestaltet sein, um die Verifikation und Validierung in der Systementwicklung von automatisierten Schiffsführungssystemen effizient zu unterstützen?“.

Für die Zielerreichung erfolgt die Darstellung des aktuellen Stands der Wissenschaft und Technik hinsichtlich der Verifikation und Validierung von automatisierten Schiffsführungssystemen in Kapitel 2. Dieses Kapitel führt die wesentlichen Konzepte des Zielsystems auf der Basis des Drei-Schichten-Modells eines automatisierten Schiffsführungssystems ein. Weitergehend erörtert diese Arbeit die Methoden der Systementwicklung auf der Basis des V-Modells, mit deren Hilfe die korrekte Funktionalität und insbesondere die Sicherheit und

Gefahrenabwehr sichergestellt wird. Das Einsatzpotenzial von Testfeldern zur Unterstützung der V+V durch empirische Datenerhebung und Tests im Open-Loop und Closed-Loop Verfahren wird in diesem Kontext dargestellt. Abschließend leitet diese Arbeit Anforderungen auf der Basis des automatisierten Schiffsführungssystems, der Systementwicklungsmethodik des Systems Engineering und der Gestaltung einer Systemarchitektur eines physikalischen Testfelds ab.

Auf der Basis der hergeleiteten Anforderungen und Problemstellung untersucht das Kapitel 3 die verwandten Arbeiten, um Handlungsbedarfe und Gestaltungspotenziale aufzudecken. Hierfür werden nicht nur unterschiedliche physikalische Testaufbauten der maritimen Domäne untersucht, sondern weiterführend die artverwandten physikalischen Testfelder der Kraftfahrt- und Luftfahrt domäne als naheliegende Einsatzgebiete von intelligenten Transportsystemen. Im Rahmen dieser Untersuchung identifiziert diese Arbeit den Handlungsbedarf der fehlenden Generizität in der Gestaltung des überwiegenden Teils der maritimen Testaufbauten. Die Systemarchitekturen weisen wenig bis keinen Fokus auf eine Erweiterbarkeit und Wiederverwendung des Testaufbaus auf. Weitergehend erkennt diese Arbeit die ungenügende Orientierung an den etablierten Systementwicklungsmethoden, beispielsweise der nahtlosen Überführung eines sich in der Entwicklung befindlichen Systems im Sinne der modellbasierten Systementwicklung. Diese Aspekte erfüllt der wesentliche Teil der physikalischen Testfelder aus dem Kraftfahrt- und Luftfahrtbereich.

Das Kapitel 4 leitet im Kontext des aufgezeigten Handlungsbedarfes den ausführlichen Entwurf einer offen und flexibel gestalteten Testfeld-Architektur als Lösungsansatz dieser Arbeit her. Diese Testfeld-Architektur ist fundiert durch die Methoden der Entwicklung, Validierung und Verifikation für ein nahtloses Testen und die Integration von virtuellen und physikalischen Komponenten. Hierfür wird die Referenzarchitektur eines generischen Testfeldes als Rahmen für die V+V des automatisierten Schiffsführungssystems durch die Konzepte des Zielsystems auf der Basis des Drei-Schichten-Modells eines automatisierten Schiffsführungssystems ausgestaltet.

Die vorgestellte Testfeld-Architektur nutzt eine zentrale nachrichtenorientierte Middleware mit einer Publish-Subscribe-Kommunikationsform, um die Komponenten des physikalischen Testfeldes und das zu testende System zu integrieren. Als Lösung der Interoperabilitäts herausforderungen nutzt die Testfeld-Architektur ein einheitliches Datenmodell gemäß S-100, das als transitiver Referenzpunkt für die Integration sämtlicher an der Kommunikation teilnehmenden Komponenten dient. Diese Gestaltungsentscheidung ergänzt sich mit der nachrichtenorientierten Middleware, da diese keine Kenntnis über den semantischen oder syntaktischen Inhalt der über die Kommunikationsinfrastruktur übertragenen Daten besitzt. Insbesondere erfüllt die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform die geforderten Beobachtungsfähigkeiten hinsichtlich des zu testenden Systems, Testszenarios und der Testfeld-Infrastruktur. Neben der Beobachtung ist die Integration eines intermediären Manipulators durch die nachrichtenorientierte Middleware möglich. Dieser Manipulator der Teststeuerung ist in der Lage die über die Testfeld-Infrastruktur übertragenen Daten zu modifizieren, um Fehlerinjektionstests gemäß der Testspezifikation durchführen zu können. Durch die Integration eines virtuellen Testfeldes auf

---

der Basis des einheitlichen Datenmodells gemäß S-100 ist die Anreicherung des physikalischen Testfeldes um simulativ erzeugte Daten möglich. Dies fördert das szenariobasierte Testen in einem Closed-Loop-Verfahren durch eine steuerbare physikalische Infrastruktur.

Da die dynamische Integration von unterschiedlichen zu testenden Systemen entgegen der statischen Integration der Testfeld-Komponenten andere Herausforderungen aufweist, verortet die Testfeld-Architektur die polymorphe Schnittstelle in Kombination mit der Kommunikationsinfrastruktur. Mit der Hilfe des Pipes und Filter Softwarearchitekturmusters als Bestandteil der polymorphen Schnittstelle ist die flexible Kombination von erweiterbaren Transformationskomponenten möglich. Diese Transformationskomponenten übernehmen dedizierte Interoperabilitäts- und Datenstromtransformationaufgaben. Als Methode zur Lösung der sich als besonders herausfordernd darstellenden semantischen Interoperabilität dient die Verwendung des einheitlichen Datenmodells gemäß S-100 als Referenz und die Überführung sämtlicher lokaler Datenmodelle. Das S-100 Datenframework bietet einheitliche Datenschemata sowie eine gemeinsame Terminologie, die in einer Datenbank gespeichert werden.

Das Kapitel 5 beschreibt die Realisierung der Testfeld-Architektur durch das physikalische Testfeld LABSKAUS (Labor für sicherheitskritische Analysen auf See) im Kontext von eMIR, das für die Einsatzszenarien der Evaluation genutzt wird.

Die Evaluation in Kapitel 6 wendet die Testfeld-Architektur in drei repräsentativen Einsatzszenarien eines physikalischen Testfeldes an. In diesem Rahmen beschreibt diese Arbeit eine umfangreiche Leistungsdatenerhebung der im Kontext von LABSKAUS implementierten Testfeld-Infrastruktur. Hierdurch ist das Treffen einer quantitativen Aussage hinsichtlich der Anforderungserfüllung durch die Kommunikationsinfrastruktur möglich. Die anschließenden Einsatzszenarien sind die explorative Beobachtung, der Test im Open-Loop-Verfahren und der Test im Closed-Loop-Verfahren. Diese Evaluation validiert die Testfeld-Architektur durch die Überprüfung der Anwendbarkeit von LABSKAUS und verifiziert den Lösungsbeitrag hinsichtlich der Anforderungserfüllung. Es erfolgt die explorative Beobachtung durch die Erfassung des Umweltszenarios mit der Hilfe von mobilen und stationären Sensorsystemknoten und der Bereitstellung der Daten über eine lose gekoppelte nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform. Dieses Einsatzszenario zeigt die Anwendbarkeit der Testfeld-Architektur für die Beobachtung von Annahmen und Bedingungen, die für statische Methoden der V+V genutzt werden. Neben dem ist es durch die beobachteten Szenarien möglich, die Modellbildung für eine simulative Testumgebung zu unterstützen.

Weitergehend setzt die Evaluation LABSKAUS als Testumgebung in der Realwelt zur Erprobung verschiedener zu testender Systeme ein. Ein Test im Open-Loop-Verfahren des Teilautomatisierungssystems MTCAS führt zu der Erkenntnis, dass durch die polymorphe Schnittstelle die Abstraktion der Testfeld-Infrastruktur durch eine dynamische Integration zur Laufzeit möglich ist. Hierbei zeigt die Kommunikationsinfrastruktur von LABSKAUS die Fähigkeit, die notwendigen Anforderungen des MTCAS hinsichtlich der Latenz und Daten-

übertragungsrate zu erfüllen. Eine Bereitstellung der notwendigen Teststimuli und die ganzheitliche Beobachtung der Testausführung sind durch die interoperable Modellierung sämtlicher Daten im einheitlichen Datenmodell gemäß S-100 möglich.

Das Einsatzszenario eines Tests des hochautomatisierten Schiffsführungssystems des OFFIS im Closed-Loop-Verfahren bestätigt das Ergebnis der Eignung der Testfeld-Architektur, da das Experiment die Testfeld-Infrastruktur wiederholt ohne infrastrukturell bedingte Änderungen wiederverwendet. Die polymorphe Schnittstelle übernimmt durch die Anpassung der Transformationsregelsätze die dynamische Integration des hochautomatisierten Schiffsführungssystems des OFFIS in die Testfeld-Infrastruktur. Hierdurch ermöglicht LABSKAUS sowohl die Bereitstellung der Teststimuli wie ebenfalls die Entgegennahme und Ausführung der kontinuierlichen Reaktionen des zu testenden Systems durch die Aktorik des Testträgers. Insbesondere für das szenariobasierte Testen ermöglicht die Kombination aus dem virtuellen Testfeld HAGGIS und LABSKAUS gemäß der Testfeld-Architektur einen nahtlosen Übergang zwischen der simulationsbasierten Erprobung bis zu Realwelttests. Während des Tests in einem Closed-Loop-Verfahren wird das hochautomatisierte Schiffsführungssystem des OFFIS in einer steuerbaren Testumgebung erfolgreich durch die Integration des virtuellen und physikalischen Testfeldes erprobt.

Die Evaluation bestätigt, dass die dargestellte Testfeld-Architektur die Ziele dieser anwendungsorientierten Forschungsarbeit erfüllt und Antworten für die in Kapitel 1.3 aufgestellten wissenschaftlichen Fragestellungen bietet.

## 7.2 Anknüpfender Forschungsbedarf

Die durchgeführte Evaluation zeigt die Zielerreichung und Anforderungserfüllung der in dieser Arbeit entwickelten Testfeld-Architektur. Basierend auf diesen Evaluationsergebnissen identifiziert diese Arbeit die folgenden anknüpfenden Erweiterungen und Anpassungen der Testfeld-Architektur.

**Echtzeitfähigkeit.** Automatisierte Systeme müssen in der Praxis quantitative Echtzeitanforderungen erfüllen, insbesondere für die regelungstechnische Kontrolle zur Führung eines Schiffes [FoPN17, S.192]. LABSKAUS verwendet als Implementierung für die nachrichtenorientierte Middleware mit Publish-Subscribe-Kommunikationsform die nicht echtzeitfähige Software RabbitMQ. Zwar sind die Tests der funktionalen Korrektheit weitestgehend während der Systementwicklung möglich, ohne dass die Testfeld-Infrastruktur die Bereitstellung eines Verarbeitungsergebnisses innerhalb einer fest definierten Zeitspanne garantiert. Für die vollständige Testabdeckung von Regelungstechnik ist jedoch die Echtzeitfähigkeit des physikalischen Testfeldes nötig, insbesondere im Rahmen einer Qualifikation des zu testenden Systems. Die vorgestellte Testfeld-Architektur bietet durch die modulare Gestaltung einen Ansatzpunkt in Form der mobilen NaviBox an Bord des Testträgers, in der ein echtzeitfähiges Betriebssystem und entsprechende Softwarekomponenten einsetzbar sind. Die Untersuchung der Auswirkungen einer Anforderung für Echtzeitfähigkeit und die Korrelation

---

hinsichtlich der Funktionalitäten für V+V bietet Potenzial für anknüpfende Forschungsarbeiten.

**Automatisierte Transformationsregelgenerierung.** Die für die dynamische Integration des zu testenden Systems vorgestellte polymorphe Schnittstelle verwendet durch einen Systemexperten manuell definierte Transformationsregeln. Durch Analyseverfahren wie beispielsweise einem Text Mining ist es denkbar, die Bedeutung der Schnittstellenspezifikationen semantisch zu analysieren und mögliche Transformationsregeln und auftretende Kollisionen automatisiert zu identifizieren. Die Gestaltung der polymorphen Schnittstelle als eine modulare Software-Komponente ermöglicht die Integration entsprechender Algorithmen. Denkbar ist eine technologische Weiterentwicklung in die Richtung der automatisierten Erzeugung von formalen XSLT Regelsätzen auf Basis der identifizierten Transformationen und die Ausführung dieser durch die existierende Laufzeitumgebung der polymorphen Schnittstelle. Zu beachten ist die notwendige Qualitätsprüfung der automatisiert generierten Transformationsregeln, um durch fehlerhafte Identifikationen das Testergebnis nicht zu beeinflussen.

**Erprobung von Kommunikationstechnologie.** Das physikalische Testfeld ist dahingehend erweiterbar, dass dieses neben dem Test von automatisierten Schiffsführungssystemen ebenso Kommunikationstechnologie erproben kann. Ein möglicher Ansatz ist die Installation einer parallelen Kommunikationsinfrastruktur, um die zu testenden Kommunikationstechnologien nicht durch die notwendigen datenaustauschenden Werkzeuge für die V+V zu beeinflussen.

**Erweiterung der Sensorik.** Der vorliegende Entwurf der Testfeld-Architektur und die Umsetzung in LABSKAUS orientieren sich an den Konzepten des Zielsystems und bieten eine dies adressierende Testumgebung. Für eine höhere Genauigkeit und Wahrscheinlichkeit des Testergebnisses sowie die Erweiterung der potenziell zu testenden Systeme ist die Integration von höherwertigeren und ergänzenden Sensoren möglich. Durch einen Multi-Sensor-Aufbau wäre die Verifikation der von dem zu testenden System benötigten Sensordaten realisierbar, um bei der Beurteilung der korrekten Funktionalität eine geringere Unsicherheit hinsichtlich möglicher, durch die installierte Sensorik resultierender Fehler zu besitzen. Die Verwendung für die Verifikation des Testszenarios neben der Bereitstellung der Teststimulation wäre beispielsweise durch die Integration von Fehlermodellen für das Simulieren der durch die Technologie auftretenden Ungenauigkeiten denkbar, wie im Falle eines Radarsystems oder GNSS.

**Qualifizierung des physikalischen Testfeldes.** Um im Rahmen der Zertifizierung und Systemerprobung von sicherheitsrelevanten Systemen genutzt zu werden bedarf es der Qualifizierung des physikalischen Testfeldes. Standards wie die ISO 26262 betrachten neben der Überprüfung der zu testenden Systeme ebenso die Eignung der Softwarewerkzeuge für die Systementwicklungs- und Testaktivitäten. Die Softwarewerkzeuge für die V+V von sicherheitsrelevanten Systemen sind unabdingbar, wie in dieser Arbeit vorgestellt. Dennoch ber-

gen diese das Potenzial für das Einschleusen von Fehlern. Die Qualifizierung des physikalischen Testfeldes würde zeigen, dass dieses dazu geeignet ist im Systementwicklungsprozess eines sicherheitskritischen Produkts ein oder mehrere von den Regularien geforderte Systementwicklungs- und Testaktivitäten zu erfüllen [SWPJ00].

**Erweiterung der Testfeld-Architektur für Multimedia Daten.** Moderne Automatisierungssysteme verwenden für die Objekterkennung zunehmend optische und akustische Sensoren, um auf die gewonnenen Daten umfassende Musteranalysen und Mustererkennungen anzuwenden. Insbesondere für sicherheitskritische Funktionen benötigen derartige Systeme für die Lagebilderzeugung bewegliche und zoomfähige Kameras oder nachtsichtfähige Systeme für die automatisierte Steuerung eines Schiffes [DaKa17], [HuBH00]. Der Lösungsbeitrag dieser Arbeit fokussiert strukturierte Daten beispielsweise eines AIS, die Maschinendaten und georäumliche Daten. Daher bieten die Ergebnisse dieser Arbeit anknüpfend das Potenzial, die Daten aus Multimedia Sensoren in die Testfeld-Architektur zu integrieren. Diese führen zu anderen Herausforderungen wegen der Leitungs- und Systemkapazitäten zur Kodierung, Übertragung und Darstellung verbunden mit den sicherheitskritischen Anforderungen des automatisierten Schiffsführungssystems. Diese Erweiterung würde die Bandbreite der unterstützten Anwendungen vergrößern.

Die Testfeld-Architektur und Realisierung durch LABSKAUS im Rahmen von eMIR ermöglicht die Erprobung von automatisierten Schiffsführungssystemen zur Unterstützung der maritimen Forschung in einer realen Umgebung. Dieser Beitrag unterstützt die Nachhaltigkeit der Forschungsergebnisse und maritime Forschungsprojekte, in dem das physikalische Testfeld für alle Interessierten der Wissenschaft, Behörden und Industrie zugänglich gemacht wird. Durch die offene Gestaltung des Lösungsbeitrages dieser Arbeit in der Form der Testfeld-Architektur ist die sukzessive Verbesserung und Erweiterung an zukünftige Anforderungen und die dargestellten anknüpfenden Forschungsbedarfe möglich, so dass potenziell eine lange Nutzungsdauer gewährleistet ist.

---

## Literatur

- [AbFH15] ABDELAAL, MOHAMED ; FRANZLE, MARTIN ; HAHN, AXEL: Nonlinear Model Predictive Control for Tracking of Underactuated Vessels under Input Constraints. In: *Modelling Symposium (EMS), 2015 IEEE European*. Madrid : IEEE, 2015 – ISBN 978-1-5090-0206-1, S. 313–318
- [AbFH16] ABDELAAL, MOHAMED ; FRÄNZLE, MARTIN ; HAHN, AXEL: NMPC-based Trajectory Tracking and Collision Avoidance of Underactuated Vessels with Elliptical Ship Domain. In: *IFAC-PapersOnLine* Bd. 49 (2016), Nr. 23, S. 22–27
- [AkMC08] AKYILDIZ, IAN F. ; MELODIA, TOMMASO ; CHOWDHURY, KAUSHIK R.: Wireless Multimedia Sensor Networks: Applications and Testbeds. In: *Proceedings of the IEEE* Bd. 96 (2008), Nr. 10, S. 1588–1605
- [Ando14] ANDO, HIDEYUKI: *Smart Ship Application Platform Project (SSAP Project)*. URL [https://www.monohakobi.com/ja/wp-content/uploads/2015/03/pdf-smartship\\_application\\_platform02.pdf](https://www.monohakobi.com/ja/wp-content/uploads/2015/03/pdf-smartship_application_platform02.pdf). - abgerufen am 2017-11-07
- [Ando17] ANDO, HIDEYUKI: *Activities of Smart Ship Application Platform 2 Project (SSAP2)*. URL [http://www.jsmea.or.jp/ssap/assets/pdf/9c\\_SSAP\\_in\\_IMPA\\_LONDON\(20170912\).pdf](http://www.jsmea.or.jp/ssap/assets/pdf/9c_SSAP_in_IMPA_LONDON(20170912).pdf). - abgerufen am 2017-11-07
- [APEW13] AWAIS, M. U. ; PALENSKY, P. ; ELSHEIKH, A. ; WIDL, E. ; MATTHIAS, S.: The high level architecture RTI as a master to the functional mock-up interface components. In: *2013 International Conference on Computing, Networking and Communications (ICNC)*. San Diego : IEEE, 2013 – ISBN 978-1-4673-5288-8, S. 315–320
- [ÅsKä76] ÅSTRÖM, KARL J. ; KÄLLSTRÖM, CLAES G.: Identification of ship steering dynamics. In: *Automatica* Bd. 12 (1976), Nr. 1, S. 9–22
- [ASRU13] ANDREN, FILIP ; STRASSER, THOMAS ; ROHJANS, SEBASTIAN ; USLAR, MATHIAS: Analyzing the need for a common modeling language for Smart Grid applications. In: *11th IEEE International Conference on Industrial Informatics (INDIN)*. Bochum : IEEE, 2013 – ISBN 978-1-4799-0752-6, S. 440–446
- [BaDG06] BARESI, LUCIANO ; DI NITTO, ELISABETTA ; GHEZZI, CARLO: Toward open-world software: Issues and challenges. In: *Computer* Bd. 39 (2006), Nr. 10, S. 36–43
- [BaGi11] BAHETI, RADHAKISAN ; GILL, HELEN: Cyber-physical systems. In: *The impact of control technology* Bd. 12 (2011), S. 161–166
- [BaMu12] BALAJI, S. ; MURUGAIYAN, M. SUNDARARAJAN: Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. In: *International Journal of Information Technology and Business Management* Bd. 2 (2012), Nr. 1, S. 26–30
- [Baum06] BAUMANN, GERD: *Was verstehen wir unter Test? Abstraktionsebenen, Begriffe und Definitionen*. URL [http://www.fkfs.de/uploads/publikationen/Baumann\\_Auto-test\\_2006\\_Test\\_Einfuehrung\\_Paper.pdf](http://www.fkfs.de/uploads/publikationen/Baumann_Auto-test_2006_Test_Einfuehrung_Paper.pdf)
- [BBBL14] BRODARIC, BOYAN ; BOOTH, NATE ; BOISVERT, ERIC ; LUCIDO, JESSICA: Levels of data interoperability in the emerging North American groundwater data network. In: *11th International Conference on Hydroinformatics*. New York City, 2014

- [Bell09] BELL, RON: Introduction & Revision of IEC 61508. In: *Measurement and Control* Bd. 42 (2009), Nr. 6, S. 174–179
- [Bend17] BENDEL, OLIVER: *Definition Cyber-physische Systeme*. URL <http://wirtschaftslexikon.gabler.de/Definition/cyber-physische-systeme.html>. - abgerufen am 2017-06-07
- [Benj16] BENJAMIN WEINERT, AXEL HAHN, OLIVER NORKUS: A domain-specific architecture framework for the maritime domain. In: *Lecture Notes in Informatics (LNI)*, 2016 — ISBN 978-3-88579-653-4, S. 773–784
- [BePr04] BENSO, ALFREDO ; PRINETTO, PAOLO: *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation*. Boston : Springer Science + Business Media, Inc., 2004 — ISBN 978-0-306-48711-8
- [Bill15] BILLESØ, MADSS BENTZEN: *ACCSEAS Final Report*. URL <http://www.accseas.eu/content/download/8190/74102/ACCSEAS%2520Final%2520Report%2520v1.pdf>
- [BlFa13] BLANCHARD, BENJAMIN S ; FABRYCKY, W. J: *Systems engineering and analysis*. München : Pearson, 2013 — ISBN 978-1-292-03839-1
- [Bmvi16] BMVI: *BMVI - Teststrecke auf der A9 erhält neue Schilder für das automatisierte Fahren*. URL <https://www.bmvi.de/SharedDocs/DE/Pressemitteilungen/2016/199-dobrindt-neue-schilder-dta.html>. - abgerufen am 2017-07-28
- [Bmvi17] BMVI: *Vorhaben auf digitalen Testfeldern (2017)*
- [Bmvi18] BMVI: *Digitales Testfeld Autobahn*. URL <https://www.bmvi.de/DE/Themen/Digitales/Digitales-Testfeld-Autobahn/Digitales-Testfeld-Autobahn.html>. - abgerufen am 2018-01-22
- [Bmwi17] BMWI: *Maritime Agenda 2025 - Für die Zukunft des maritimen Wirtschaftsstandortes Deutschland*. URL [https://www.bmwi.de/Redaktion/DE/Publikationen/Wirtschaft/maritime-agenda-2025.pdf?\\_\\_blob=publicationFile&v=16](https://www.bmwi.de/Redaktion/DE/Publikationen/Wirtschaft/maritime-agenda-2025.pdf?__blob=publicationFile&v=16). - abgerufen am 2017-09-20
- [Boru12] BORUP, OLE: *e-Navigation Test Bed with a suitable Infrastructure including necessary Communication Links*. URL [http://efficiensea.org/files/mainoutputs/wp4/efficiensea\\_wp4\\_9.pdf](http://efficiensea.org/files/mainoutputs/wp4/efficiensea_wp4_9.pdf). - abgerufen am 2017-11-03
- [BrFo05] BREIVIK, M. ; FOSSEN, T.I.: Principles of Guidance-Based Path Following in 2D and 3D. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. Seville : IEEE, 2005 — ISBN 978-0-7803-9567-1, S. 627–634
- [BrHa17] BRINKMANN, MARIUS ; HAHN, AXEL: Testbed architecture for maritime cyber physical systems. In: *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. Emden : IEEE, 2017 — ISBN 978-1-5386-0837-1, S. 923–928
- [BrKr08] BRINGMANN, ECKARD ; KRÄMER, ANDREAS: Model-Based Testing of Automotive Systems. In: *1st International Conference on Software Testing, Verification, and Validation*. Lillehammer : IEEE, 2008 — ISBN 978-0-7695-3127-4, S. 485–493

- 
- [BRRN05] BROWN, DANIEL G. ; RIOLO, RICK ; ROBINSON, DEREK T. ; NORTH, MICHAEL ; RAND, WILLIAM: Spatial process and data models: Toward integration of agent-based models and GIS. In: *Journal of Geographical Systems* Bd. 7 (2005), Nr. 1, S. 25–47
- [BuLö00] BUSCHMANN, F. ; LÖCKENHOFF, C. (Hrsg.): *Pattern-orientierte Softwarearchitektur: ein Pattern-System*. 1. München : Addison-Wesley, 2000 — ISBN 978-3-8273-1282-2
- [Burg12] BURGHARDT, MANFRED: *Projektmanagement: Leitfaden für die Planung, Überwachung und Steuerung von Projekten*. 9. Erlangen : Publicis, 2012 — ISBN 978-3-89578-399-9
- [Burm16] BURMEISTER, HANS-CHRISTOPH: *Autonomous Navigation Results from the MUNIN Testbed*. URL <https://www.cml.fraunhofer.de/content/dam/cml/de/documents/Sonstiges/MUNIN%20-%20160621%20-%20ASS%20-%20MUNIN%20-%20final.pdf>. - abgerufen am 2017-07-16
- [Buse12] BUSEMANN, CLAAS: *Ein effizienter Ansatz zur Übersetzung zwischen Service-Protokollen durch die Modellierung von Verhaltensunterschieden*. Oldenburg, Universität Oldenburg, 2012
- [CaWe85] CARDELLI, LUCA ; WEGNER, PETER: On understanding types, data abstraction, and polymorphism. In: *ACM Computing Surveys* Bd. 17 (1985), Nr. 4, S. 471–523
- [CoFS05] CONRAD, MIRKO ; FEY, INES ; SADEGHIPOUR, SADEGH: Systematic Model-Based Testing of Embedded Automotive Software. In: *Electronic Notes in Theoretical Computer Science* Bd. 111 (2005), S. 13–26
- [Curr05] CURRY, EDWARD: Message-Oriented Middleware. In: MAHMOUD, Q. H. (Hrsg.): *Middleware for Communications*. Chichester, UK : John Wiley & Sons, Ltd, 2005 — ISBN 978-0-470-86208-7, S. 1–28
- [DaHe18] DAMM, WERNER ; HEIDL, PETER: *SafeTRANS Working Group - Highly Automated Systems: Test, Safety, and Development Processes*. URL [https://www.of-fis.de/fileadmin/content/files/download\\_tools/roadmaps\\_und\\_studien/SafeTRANS\\_ManagementSummary\\_HighlyAutomatedSystems.pdf](https://www.of-fis.de/fileadmin/content/files/download_tools/roadmaps_und_studien/SafeTRANS_ManagementSummary_HighlyAutomatedSystems.pdf). - abgerufen am 2018-03-09
- [DaJa95] DAWSON, S. ; JAHANIAN, F.: Probing and fault injection of protocol implementations. In: *Proceedings of 15th International Conference on Distributed Computing Systems*. Vancouver : IEEE Comput. Soc. Press, 1995 — ISBN 978-0-8186-7025-1, S. 351–359
- [DaKa17] DAMM, WERNER ; KALMAR, RALF: Autonome Systeme: Fähigkeiten und Anforderungen. In: *Informatik-Spektrum* Bd. 40 (2017), Nr. 5, S. 400–408
- [DBFH16] DENKER, CHRISTIAN ; BALDAUF, MICHAEL ; FISCHER, SANDRO ; HAHN, AXEL ; ZIEBOLD, RALF ; GEHRMANN, ELKE ; SEMANN, MIKA: e-Navigation based Cooperative Collision Avoidance at Sea: The MTCAS Project. In: *2016 European Navigation Conference (ENC)*. Helsinki, 2016
- [DEGN12] DAVIS, L. A. ; ENGE, P. ; GAO, G. X. ; NATIONAL ACADEMY OF ENGINEERING ; ZHONGGUO GONG CHENG YUAN (Hrsg.): *Global navigation satellite systems: report of a joint workshop of the National Academy of Engineering and the Chinese*

*Academy of Engineering*. Washington, D.C : National Academies Press, 2012  
— ISBN 978-0-309-22275-4

- [DiHa14] DIBBERN, CHRISTOPH ; HAHN, AXEL: Maritime Traffic Co-Simulation For Analyses of Maritime Systems. In: *28th European Conference on Modelling and Simulation ECMS 2014, At Brescia, Italy*, 2014. — 00000
- [Din05] DIN: *ISO 17894:2005 - Ships and marine technology - Computer applications - General principles for the development and use of programmable electronic systems in marine applications* : Beuth, 2005
- [Din17] DIN: *Leittechnik - Prozeßautomatisierung - Automatisierung mit Prozeßrechen-systemen, Begriffe*. URL <https://www.beuth.de/de/vornorm/din-v-19233/3361842>. - abgerufen am 2017-07-21
- [DJMT96] DAWSON, S. ; JAHANIAN, F. ; MITTON, T. ; TECK-LEE TUNG: Testing of fault-tolerant and real-time distributed systems via protocol fault injection. In: *Proceedings of Annual Symposium on Fault Tolerant Computing*. Sendai : IEEE Comput. Soc. Press, 1996 — ISBN 978-0-8186-7261-3, S. 404–414
- [Dlr18a] DLR: *Advanced Technology Research Aircraft*. URL <http://www.dlr.de/dlr/desktopdefault.aspx/tabid-10913/>. - abgerufen am 2018-03-19
- [Dlr18b] DLR: *ATTAS - Advanced Technologies Testing Aircraft System*. URL <http://www.dlr.de/Portaldata/18/Resources/dokumente/attas.pdf>. - abgerufen am 2018-01-29
- [Dnvg11] DNVGL: *DNV Ship rules Pt.6 Ch.8 - Nautical Safety - Rules and standards*. URL <https://rules.dnvgl.com/docs/pdf/DNV/rulesship/2011-01/ts608.pdf>. - abgerufen am 2018-05-21
- [Dong12] DONGES, EDMUND: Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort. In: WINNER, H. ; HAKULI, S. ; WOLF, G. (Hrsg.): . Wiesbaden : Vieweg+Teubner Verlag, 2012 — ISBN 978-3-8348-1457-9, S. 15–23
- [DWPM11] DROLIA, UTSAV ; WANG, ZHENYAN ; PANT, YASH ; MANGHARAM, RAHUL: AutoPlug: An automotive test-bed for electronic controller unit testing and verification. In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Washington, D.C : IEEE, 2011 — ISBN 978-1-4577-2197-7, S. 1187–1192
- [EFGK03] EUGSTER, PATRICK TH. ; FELBER, PASCAL A. ; GUERRAOU, RACHID ; KERMARREC, ANNE-MARIE: The many faces of publish/subscribe. In: *ACM Computing Surveys* Bd. 35 (2003), Nr. 2, S. 114–131
- [Enge10] ENGEL, AVNER: *Verification, validation, and testing of engineered systems*. Hoboken : Wiley, 2010 — ISBN 978-0-470-61884-4
- [Enge16] ENGEL, ALEXANDER: *EUROCONTROL Specification for Surveillance Data Exchange - Part 1*. URL <http://www.eurocontrol.int/sites/default/files/content/documents/nm/asterix/Part%201%20-%20EUROCONTROL%20Specification%20ASTERIX%20%28SPEC-149%29%20Ed%202.4.pdf>. - abgerufen am 2018-05-12

- 
- [Euro16] EUROPEAN COMMISSION: *Paris Agreement - European Commission*. URL [http://ec.europa.eu/clima/policies/international/negotiations/paris/index\\_en.htm](http://ec.europa.eu/clima/policies/international/negotiations/paris/index_en.htm). - abgerufen am 2016-08-22
- [FeLo04] FETTKE, PETER ; LOOS, PETER: *Entwicklung eines Bezugsrahmens zur Evaluierung von Referenzmodellen*. Mainz : ISYM, 2004
- [Fire14] FIRESMITH, DONALD G.: *Common system and software testing pitfalls: how to prevent and mitigate them: descriptions, symptoms, consequences, causes, and recommendations*. Upper Saddle River : Addison-Wesley, 2014 — ISBN 978-0-13-374855-0
- [Fmid14] FMI DEVELOPMENT GROUP: *Functional Mock-up Interface for Model Exchange and Co-Simulation*. URL [https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI\\_for\\_ModelExchange\\_and\\_CoSimulation\\_v2.0.pdf](https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI_for_ModelExchange_and_CoSimulation_v2.0.pdf). - abgerufen am 2018-03-27
- [FoPN17] FOSSEN, THOR I ; PETTERSEN, K. Y ; NIJMEIJER, H: *Sensing and control for autonomous vehicles: applications to land, water and air vehicles*. Basel : Springer International Publishing, 2017 — ISBN 978-3-319-55372-6
- [Foss11] FOSSEN, THOR I.: *Handbook of Marine Craft Hydrodynamics and Motion Control: Fossen/Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester : John Wiley & Sons, Ltd, 2011 — ISBN 978-1-119-99413-8
- [FPGD16] FISCHER, MARTIN ; PLÄTTNER, JENS ; GROLMS, GAVIN ; DIRK, ASSMANN ; LAPOEHN, STEPHAN ; FRANK, KÖSTER: *Virtual Prototyping for Highly Automated Vehicle Function Validation utilizing a Vehicle-in-the-Loop Driving Simulator Facility*. In: *Proceedings of the DSC 2016 Europe*. Paris, 2016
- [FrSK12] FRANKIEWICZ, TOBIAS ; SCHNIEDER, LARS ; KÖSTER, FRANK: *Application Platform Intelligent Mobility-Test Site Architecture and Vehicle2X Communication Setup*. In: *ITS World Congress*. Wien. Bd. 22, 2012
- [GaEK01] GAUSEMEIER, JÜRGEN ; EBBESMEYER, PETER ; KALLMEYER, FERDINAND: *Produktinnovation: strategische Planung und Entwicklung der Produkte von morgen*. München : Hanser, 2001 — ISBN 978-3-446-21631-0
- [GaRi03] GAMMA, E. ; RIEHLE, D. (Hrsg.): *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software, Professionelle Softwareentwicklung*. 5. München : Addison-Wesley, 2003 — ISBN 978-3-8273-1862-6
- [Govu00] GOVUK: *SOLAS V on Safety of Navigation*. URL [https://www.gov.uk/government/uploads/system/uploads/attachment\\_data/file/343175/solas\\_v\\_on\\_safety\\_of\\_navigation.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/343175/solas_v_on_safety_of_navigation.pdf). - abgerufen am 2017-05-12
- [GroV13] GROVES, PAUL D.: *Principles of GNSS, inertial, and multisensor integrated navigation systems, GNSS technology and application series*. 2nd ed. Boston : Artech House, 2013 — ISBN 978-1-60807-005-3
- [GrVE09] GRAHAM, DOROTHY ; VEENENDAAL, ERIK VAN ; EVANS, ISABEL: *Foundations of software testing ISTQB certification*. Andover : Course technology Cengage Learning, 2009 — ISBN 978-1-84480-989-9

- [Gupt13] GUPTA, SATYANDRA K.: *Developing Autonomy for Unmanned Surface Vehicles*. URL <https://pdfs.semanticscholar.org/4478/8d62804f6511d249cb9873a6037d0c6214bb.pdf>
- [HaHG13] HAHN, AXEL ; HÄUSLER, STEFAN ; GROBE AUSTING, STEPHAN: *Quantitatives Entwicklungsmanagement: modellbasierte Analyse von Produktentwicklungsprozessen*. Berlin : Springer Vieweg, 2013 — ISBN 978-3-642-34509-8
- [Hahn14] HAHN, AXEL: Test Bed for Safety Assessment of New e-Navigation Systems. In: *International Journal of e-Navigation and Maritime Economy* Bd. 1 (2014), S. 14–28
- [Hahn15] HAHN, AXEL: Simulation Environment for Risk Assessment of E-Navigation Systems. In: *2015 Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering (OMAE2015): Volume 6* : ASME, 2015 — ISBN 978-0-7918-5649-9, S. V003T02A072
- [Halb94] HALBACH, WULF R.: *Interfaces: Medien- und kommunikationstheoretische Elemente einer Interface-Theorie*. München : Fink, 1994 — ISBN 978-3-7705-2934-6
- [HaSS18] HAHN, AXEL ; STEIDEL, MATTHIAS ; SIEGEL, LEON: Künstliche Intelligenz und hochautomatisierte Navigation. In: *Schiff & Hafen* Bd. 02/2018 (2018), S. 52–55
- [HBFF16] HAHN, AXEL ; BOLLES, ANDRE ; FRÄNZLE, MARTIN ; FRÖSCHLE, SIBYLLE ; HYOUNG PARK, JIN: Requirements for e-Navigation Architectures. In: *International Journal of eNavigation and Maritime Economy* Bd. 5 (2016), S. 1–20
- [HGBS15] HAHN, AXEL ; GOLLÜCKE, VOLKER ; BUSCHMANN, CARSTEN ; SCHWEIGERT, SÖREN: Virtual test bed for maritime safety assessment. In: *44 Scientific Journals of the Maritime University of Szczecin* (2015), Nr. 44, S. 116–122
- [HöHR08] HÖHN, R. ; HÖPPNER, S. ; RAUSCH, A. (Hrsg.): *Das V-Modell XT: Anwendungen, Werkzeuge, Standards, eXamen.press*. Berlin : Springer, 2008 — ISBN 978-3-540-30250-6
- [Holz11] HOLZAPFEL, F. (Hrsg.): *Advances in aerospace guidance, navigation and control: selected papers of the 1st CEAS Specialist Conference on Guidance, Navigation and Control*. Berlin : Springer, 2011 — ISBN 978-3-642-19817-5
- [HoWo04] HOHPE, GREGOR ; WOOLF, BOBBY: *Enterprise integration patterns: designing, building, and deploying messaging solutions*. Boston : Addison-Wesley, 2004 — ISBN 978-0-321-20068-6
- [HuBH00] HUME, DAVID ; BEAUCHAMP, TOM L. ; HUME, DAVID: *An enquiry concerning human understanding: a critical edition, The Clarendon edition of the works of David Hume*. Oxford: New York : Clarendon Press; Oxford University Press, 2000 — ISBN 978-0-19-825060-9
- [Iala00] IALA: *IALA Recommendation V-145 on the Inter-VTS Exchange Format (IVEF) Service - Edition 1*. URL <http://www.iala-aism.org/product/inter-vts-exchange-format-service-145/?download=true>. - abgerufen am 2018-05-12
- [Iala17] IALA: *Planning and Reporting of E-Navigation Testbeds*. URL <http://www.iala-aism.org/content/uploads/2017/03/1107-Ed2-on-Planning-Testbeds-and-Reporting-of-Testbed-Results-Ed-2-June-2016-1.pdf>. - abgerufen am 2017-05-19

- 
- [Iala18] IALA: *Test Beds/Projects*. URL <http://www.iala-aism.org/products-projects/e-navigation/test-bedsprojects/>. - abgerufen am 2018-03-19
- [Iec06] IEC: *Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA)*. URL <https://webstore.iec.ch/publication/3571>. - abgerufen am 2017-09-20
- [Iec08] IEC: *IEC 61162-3:2008 - Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 3: Serial data instrument network*. URL <https://webstore.iec.ch/publication/4714>. - abgerufen am 2018-05-11
- [Iec15] IEC: *IEC 61174:2015 - Maritime navigation and radiocommunication equipment and systems - Electronic chart display and information system (ECDIS) - Operational and performance requirements, methods of testing and required test results*. URL <https://webstore.iec.ch/publication/23128>. - abgerufen am 2018-05-11
- [Iec16] IEC: *IEC 61162-1:2016 - Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 1: Single talker and multiple listeners*. URL <https://webstore.iec.ch/publication/25754>. - abgerufen am 2018-05-11
- [Iec18a] IEC: *IEC 61162-460:2018 - Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 460: Multiple talkers and multiple listeners - Ethernet interconnection - Safety and security*. URL <https://webstore.iec.ch/publication/63097>. - abgerufen am 2018-05-11
- [Iec18b] IEC: *IEC 61850:2018 - Communication networks and systems for power utility automation*. URL <https://webstore.iec.ch/publication/6028>
- [Iec98] IEC: *IEC 61162-2:1998 - Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 2: Single talker and multiple listeners, high-speed transmission*. URL <https://webstore.iec.ch/publication/4711>. - abgerufen am 2018-05-11
- [Iho17] IHO: *S-100 - Universal Hydrographic Data Model*. URL [https://www.iho.int/iho\\_pubs/standard/S-100/S-100\\_Ed\\_3/S-100\\_Edition\\_3.0.0.pdf](https://www.iho.int/iho_pubs/standard/S-100/S-100_Ed_3/S-100_Edition_3.0.0.pdf). - abgerufen am 2018-01-18
- [Iic17] IIC: *Testbed Working Group | Industrial Internet Consortium*. URL <https://www.iiconsortium.org/wc-testbeds.htm>. - abgerufen am 2017-05-13
- [Imo11] IMO: *Report from the EfficienSea Project*. URL [http://www.iala-aism.org/content/uploads/2016/06/COMSAR\\_16-INF.2\\_-\\_Report\\_from\\_the\\_EfficienSea\\_Project\\_Denmark\\_Estonia\\_Finland...1.pdf](http://www.iala-aism.org/content/uploads/2016/06/COMSAR_16-INF.2_-_Report_from_the_EfficienSea_Project_Denmark_Estonia_Finland...1.pdf). - abgerufen am 2017-07-16
- [Imo14] IMO: *Development of an E-Navigation Strategy Implementation Plan*. URL <http://s3-eu-west-1.amazonaws.com/stm-stmvalidation/uploads/20160420153610/NCSR1-INF16-Denmark-the-Republic-of-Korea-and-Sweden.pdf>. - abgerufen am 2018-01-28
- [Imo16] IMO: *e-Navigation Strategy Implementation Plan*. URL <http://www.imo.org/en/OurWork/Safety/Navigation/Documents/enavigation/SIP.pdf#search=sip>. - abgerufen am 2017-04-13

- [Imo17a] IMO: *International Convention for the Safety of Life at Sea (SOLAS), 1974*. URL [http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-\(SOLAS\),-1974.aspx](http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS),-1974.aspx). - abgerufen am 2017-04-07
- [Imo17b] IMO: *IMO NCSR 4/6: Guidelines Associated with Multi-system Shipborne Radio-navigation Receivers Dealing with the Harmonized Provision of PNT Data and Integrity Information. Report of the Correspondence Group on the Development of the Guidelines for Shipborne Position, Navigation and Timing Data Processing*. London : IMO, 2017
- [IsII15] ISO ; IEC ; IEEE: *ISO/IEC/IEEE 15288:2015 - Systems and software engineering -- System life cycle processes*. URL <https://www.iso.org/standard/63711.html>. - abgerufen am 2017-07-25
- [Isma17] ISMAGILOV, MARAT: *Hermitage - First E-Navigation Test-Bed in Russia*. URL <http://www.iala-aism.org/content/uploads/2016/09/1030-Marat-Ismagilov-Hermitage-First-e-Nav-Testbed-in-Russia-%E2%80%94-v2.pdf>. - abgerufen am 2017-11-06
- [Iso16] ISO: *ISO 26262-5:2011(en), Road vehicles — Functional safety — Part 5: Product development at the hardware level*. URL <https://www.iso.org/obp/ui/#iso:std:iso:26262:-5:ed-1:v1:en>. - abgerufen am 2016-11-11
- [JBBB14] JOHN, OLE ; BURMEISTER, HANS-CHRISTOPH ; BRÖDJE, ANDERS ; BORNHORST, CLAUS ; GRUBE, CHRISTIAN: *Assessing the MONALISA 2.0 Concept: Establishment of the European Maritime Simulation Network*. In: *International Symposium Information on Ships (ISIS)*. Hamburg, 2014, S. 11
- [JHHK10] JOST, HENNING ; HAHN, AXEL ; HAUSLER, STEFAN ; KOHLER, SILKE ; GACNIK, JAN ; KOSTER, FRANK ; LEMMER, KARSTEN: *Supporting qualification: Safety standard compliant process planning and monitoring*. In: *IEEE Symposium on Product Compliance Engineering Proceedings*. Boston : IEEE, 2010 — ISBN 978-1-4244-7498-1, S. 1–6
- [JoPC16] JOHANSEN, TOR ARNE ; PEREZ, TRISTAN ; CRISTOFARO, ANDREA: *Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection With Predictive Hazard Assessment*. In: *IEEE Transactions on Intelligent Transportation Systems* Bd. 17 (2016), Nr. 12, S. 3407–3422
- [KKDH05] VAN KATWIJK, R. ; VAN KONINGSBRUGGEN, P. ; DE SCHUTTER, B. ; HELLENDORRN, J.: *Test Bed for Multiagent Control Systems in Road Traffic Management*. In: *Transportation Research Record: Journal of the Transportation Research Board* Bd. 1910 (2005), S. 108–115
- [KlAl11] KLEE, HAROLD ; ALLEN, RANDAL: *Simulation of dynamic systems with MATLAB and Simulink /: Harold Klee, Randal Allen*. 2. Boca Raton : CRC Press, 2011 — ISBN 978-1-4398-3673-6
- [KnDH09] KNAGAARD, ANNEMETTE ; DZIEWICKI, MAREK ; HEIKONEN, KAISU: *A plan for testing and verification of important components of e-Navigation and associated technical solutions*. URL [http://efficiensea.org/files/mainoutputs/wp4/efficiensea\\_wp4\\_2.pdf](http://efficiensea.org/files/mainoutputs/wp4/efficiensea_wp4_2.pdf). - abgerufen am 2017-11-03

- 
- [Kong17] KONGSBERG: *KNC to revolutionise ship traffic management with launch of SESAME Straits e-Navigation project*. URL <https://www.kongsberg.com/en/kds/knc/news/featurestories/knc-to-revolutionise-ship-traffic-management-with-launch-of-sesame-straits-e-navigation-project/>. - abgerufen am 2017-11-07
- [KoWa16] KOOPMAN, PHILIP ; WAGNER, MICHAEL: Challenges in Autonomous Vehicle Testing and Validation. In: *SAE International Journal of Transportation Safety* Bd. 4 (2016), Nr. 1, S. 15–24
- [Kris05] KRISTIANSEN, SVEIN: *Maritime transportation: safety management and risk analysis*. Amsterdam : Elsevier/Butterworth-Heinemann, 2005 — ISBN 978-0-7506-5999-4
- [KuDW00] KUHL, FREDERICK ; DAHMANN, JUDITH ; WEATHERLY, RICHARD: *Creating computer simulation systems: an introduction to the high level architecture*. Upper Saddle River : Prentice Hall PTR, 2000 — ISBN 978-0-13-022511-5
- [LaHa17] LAMM, ARNE ; HAHN, AXEL: Detecting Maneuvers in Maritime Observation Data with CUSUM. In: *IEEE International Symposium on Signal Processing and Information*. Bilbao, 2017
- [Lane17] LANE, DEIRDRE: *Dublin Bay Digital Diamond e-Navigation Demonstrator*. URL <http://www.iala-aism.org/content/uploads/2016/09/1400-Deirdre-Lane-Dubin-Bay-digital-diamond.pdf>. - abgerufen am 2017-11-03
- [LåVK15] LÅG, STEINAR ; VARTDAL, BJØRN-JOHAN ; KNUTSEN, KNUT ERIK: *DNV GL Strategic Research & Innovation Position Paper 04-2015 - Ship Connectivity*. URL <https://www.dnvgl.com/publications/ship-connectivity-28107>. - abgerufen am 2017-03-29
- [LeDz17] LEDOCHOWSKI, MAREK ; DZIEWICKI, MAREK: *D4.8 Results for tests with sensor data and analysis*. URL <https://efficiensea2.org/wp-content/uploads/2018/04/Deliverable-4.8-Results-for-tests-with-sensor-data-and-analysis.pdf>. - abgerufen am 2018-05-16
- [Lemm18] LEMMER, KRASTEN: *Application Platform for Intelligent Mobility*. URL [http://www.dlr.de/fs/en/Portaldata/16/Resources/aim/DLR-TS\\_Application-PlatformIntelligentMobility.pdf](http://www.dlr.de/fs/en/Portaldata/16/Resources/aim/DLR-TS_Application-PlatformIntelligentMobility.pdf). - abgerufen am 2018-01-22
- [LeSe17] LEE, EDWARD A. ; SESHIA, SANJIT A.: *Introduction to embedded systems: a cyber-physical systems approach*. 2. Cambridge : MIT Press, 2017 — ISBN 978-0-262-53381-2
- [LeWi17] LETTNIN, D. ; WINTERHOLER, M. (Hrsg.): *Embedded Software Verification and Debugging, Embedded Systems*. New York : Springer New York, 2017 — ISBN 978-1-4614-2265-5
- [LiHL09] LIGGINS, M. E. ; HALL, D. L. ; LLINAS, J. (Hrsg.): *Handbook of multisensor data fusion: theory and practice, The electrical engineering and applied signal processing series*. 2. Boca Raton : CRC Press, 2009 — ISBN 978-1-4200-5308-1
- [LiHu06] LIN, BIN ; HUANG, CHIH-HAO: Comparison between ARPA radar and AIS characteristics for vessel traffic services. In: *Journal of marine science and technology* Bd. 14 (2006), Nr. 3, S. 182–189

- [MaMo18] MALYANKAR, RAPHAEL ; MONG, EIVIND: *S-100 Product Specification Interoperability Design*. URL [https://www.iho.int/mtg\\_docs/com\\_wg/S-100WG/S-100WG2/S-100WG02-10.3A\\_InteroperabilityDesign.pdf](https://www.iho.int/mtg_docs/com_wg/S-100WG/S-100WG2/S-100WG02-10.3A_InteroperabilityDesign.pdf). - abgerufen am 2018-02-10
- [MBMC07] MOTAHARI NEZHAD, HAMID REZA ; BENATALLAH, BOUALEM ; MARTENS, AXEL ; CURBERA, FRANCISCO ; CASATI, FABIO: Semi-automated adaptation of service interactions. In: *Proceedings of the 16th international conference on World Wide Web* : ACM Press, 2007 — ISBN 978-1-59593-654-7, S. 993–1002
- [MiTa04] MISSIKOFF, MICHELE ; TAGLINO, FRANCESCO: An Ontology-based Platform for Semantic Interoperability. In: STAAB, S. ; STUDER, R. (Hrsg.): *Handbook on Ontologies*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2004 — ISBN 978-3-662-11957-0, S. 617–633
- [MSRH10] MICHAILIDIS, ALEXANDER ; SPIETH, UWE ; RINGLER, THOMAS ; HEDENETZ, BERND ; KOWALEWSKI, STEFAN: Test front loading in early stages of automotive software development based on AUTOSAR. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Piscataway : IEEE, 2010 — ISBN 978-3-9810801-6-2
- [NHMG17] NICKOVIC, DEJAN ; HERZNER, WOLFGANG ; MÖHLMANN, EIKE ; GERWINN, SEBASTIAN ; GARCIA PADILLA, GUSTAVO ; DIEGMANN, FABIAN ; HAKKI, SADRI ; SNOOK, COLIN: *D3.2.2 v1 VV Methodology.pdf*. URL <https://drive.google.com/file/d/1F5UnxSzSM4n52fPbrvFCo3iT-CDW4kfzU/view>. — ENABLE-S3 - D3.2.2 v1 V&V Methodology
- [NiKi09] NIARAKI, ABOLGHASEM SADEGHI ; KIM, KYEHYUN: Ontology based personalized route planning system using a multi-criteria decision making approach. In: *Expert Systems with Applications* Bd. 36 (2009), Nr. 2, S. 2250–2259
- [NPWM13] NETTEN, BART ; PASSCHIER, IGOR ; WEDEMEIJER, HARRY ; MAAS, SANDER ; VAN LEEUWEN, COEN: Technical Evaluation of Cooperative Systems Experience from the DITCM Test Site. In: *9 th ITS European Congress*, 2013
- [PaLP16] PARK, DAEWON ; LEE, KWANGIL ; PARK, SUHYUN: Common Vocabulary Set for Support of Interoperability of Data Contents of Maritime Equipment and Systems in E-Navigation Services. In: *6th International Conference on IT Convergence and Security (ICITCS)*. Prague : IEEE, 2016 — ISBN 978-1-5090-3765-0, S. 1–4
- [PaPa16] PARK, DAEWON ; PARK, SUHYUN: Multiple-domain marine data utilization structure for e-navigation. In: *Cluster Computing* Bd. 19 (2016), Nr. 1, S. 301–308
- [PaVC00] PALECZNY, MICHAEL ; VICK, CHRISTOPHER ; CLICK, CLIFF: The Java HotSpot™ Server Compiler. In: *JVM'01 Proceedings of the 2001 Symposium on Java™ Virtual Machine Research and Technology Symposium - Volume 1*. Monterey : ACM, S. 13
- [PBJP14] PORATHE, THOMAS ; BORUP, OLE ; JEONG, JUNG SIK ; PARK, JIN HYOUNG ; ANDERSEN CAMRE, DAVID ; BRÖDJE, ANDERS: Ship traffic management route exchange: acceptance in Korea and Sweden, a cross cultural study. In: *Proceedings of the International Symposium Information on Ships, ISIS 2014*. Hamburg, 2014, S. 64–79

- 
- [PCVB06] PEIKENKAMP, THOMAS ; CAVALLO, ANTONELLA ; VALACCA, LAURA ; BÖDE, ECKARD ; PRETZER, MATTHIAS ; HAHN, E. MORITZ: Towards a Unified Model-Based Safety Assessment. In: *Computer Safety, Reliability, and Security*. Bd. 4166. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006 — ISBN 978-3-540-45762-6, S. 275–288
- [PhNa07] PHILLIPS, CHARLES L ; NAGLE, H. TROY: *Digital control systems analysis and design*. Harlow : Prentice Hall, 2007 — ISBN 978-0-13-081222-3
- [PNWM13] PASSCHIER, I. ; NETTEN, B.D. ; WEDEMEIJER, H. ; MAAS, S.M.P. ; VAN LEEUWEN, C.J. ; SCHACKMANN, P.P.M.: DITCM roadside facilities for cooperative systems testing and evaluation. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. The Hague : IEEE, 2013 — ISBN 978-1-4799-2914-6, S. 936–942
- [PoOl15] PORATHE, THOMAS ; OLTSMANN, JAN-HENDRIK: *ACCSEAS Baseline and Priorities Report*, 2015
- [Pora13] PORATHE, THOMAS: *ACCSEAS Baseline and Priorities Report*. URL <http://www.accseas.eu/content/download/2743/25720/ACCSEAS%2520Baseline%2520and%2520Priorities%2520Report%2520v3.0.pdf>. - abgerufen am 2016-08-22
- [PuRP06] PUDER, ARNO ; RÖMER, KAY ; PILHOFER, FRANK: *Distributed systems architecture: a middleware approach*. Amsterdam ; Boston : Elsevier : Morgan Kaufmann, 2006 — ISBN 978-1-55860-648-7
- [QPPS08] QUARTEL, DICK ; POKRAEV, STANISLAV ; PESSOA, RODRIGO MANTOVANELI ; SINDEREN, MARTEN VAN: Model-Driven Development of a Mediation Service. In: *12th International IEEE Enterprise Distributed Object Computing Conference*. München : IEEE, 2008 — ISBN 978-0-7695-3373-5, S. 117–126
- [RLSS10] RAJKUMAR, RAGUNATHAN (RAJ) ; LEE, INSUP ; SHA, LUI ; STANKOVIC, JOHN: Cyber-physical systems: the next computing revolution. In: *Proceedings of the 47th Design Automation Conference*. Anaheim : ACM Press, 2010 — ISBN 978-1-4503-0002-5, S. 731
- [RoGS14] ROSTAŃSKI, MACIEJ ; GROCHLA, KRZYSZTOF ; SEMAN, ALEKSANDER: Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ. In: *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*. Warsaw : ACSIS, 2014, S. 879–884
- [RoRo06] ROBERTSON, SUZANNE ; ROBERTSON, JAMES: *Mastering the requirements process. 2*. Upper Saddle River : Addison-Wesley, 2006 — ISBN 978-0-321-41949-1
- [RoVe08] ROSEMANN, MICHAEL ; VESSEY, IRIS: Toward Improving the Relevance of Information Systems Research to Practice: The Role of Applicability Checks. In: *MIS Quarterly* Bd. 32 (2008), Nr. 1, S. 1
- [RTTZ04] ROYCHOUDHURY, A. ; THIAGARAJAN, P.S. ; TUAN-ANH TRAN ; ZVEREVA, V.A.: Automatic Generation of Protocol Converters from Scenario-Based Specifications. In: *25th IEEE International Real-Time Systems Symposium*. Lisbon : IEEE, 2004 — ISBN 978-0-7695-2247-0, S. 447–458

- [SaDP12] SANGIOVANNI-VINCENTELLI, ALBERTO ; DAMM, WERNER ; PASSERONE, ROBERTO: Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems\*. In: *European Journal of Control* Bd. 18 (2012), Nr. 3, S. 217–238
- [Schä09] SCHÄTZ, BERNHARD: Formalization and Rule-Based Transformation of EMF E-core-Based Models. In: GAŠEVIĆ, D. ; LÄMMEL, R. ; VAN WYK, E. (Hrsg.): *Software Language Engineering*. Bd. 5452. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009 — ISBN 978-3-642-00433-9, S. 227–244
- [Schu17] SCHUETT, TODD: *SESAME Straits e-Navigation Test Bed*. URL <http://straitstms.com/>. - abgerufen am 2017-11-07
- [ScKr15] SCHNIEDER, LARS ; KRENKEL, RALF: Betreibermodell einer Forschungsinfrastruktur für die Entwicklung intelligenter Mobilitätsdienste im realen Verkehrsumfeld. In: *16. Symposium Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*. Bd. 16. Braunschweig, 2015 — ISBN 978-3-937655-34-5, S. 108–116
- [ScSp12] SCHILL, ALEXANDER ; SPRINGER, THOMAS: *Verteilte Systeme: Grundlagen und Basistechnologien, eXamen.press*. 2. Berlin : Springer Vieweg, 2012 — ISBN 978-3-642-25796-4
- [ScZu10] SCHÄUFFELE, JÖRG ; ZURAWKA, THOMAS: *Automotive Software Engineering Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. Wiesbaden : Vieweg+Teubner Verlag / GWV Fachverlage, Wiesbaden, 2010 — ISBN 978-3-8348-9368-0
- [Sgcg12] SGCG: *Smart Grid Coordination Group Smart Grid Reference Architecture*. URL [https://ec.europa.eu/energy/sites/ener/files/documents/xpert\\_group1\\_reference\\_architecture.pdf](https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf). - abgerufen am 2018-01-21
- [Sher10] SHERIF, M. H. (Hrsg.): *Handbook of enterprise integration*. Boca Raton : Auerbach Pub./CRC Press, 2010 — ISBN 978-1-4200-7821-3
- [Shet99] SHETH, AMIT P.: Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In: *Interoperating Geographic Information Systems*. Boston : Springer US, 1999 — ISBN 978-1-4613-7363-6, S. 5–29
- [Sing15] SINGH, DURGESH: *IBS: Integrated Bridge System*. URL <http://maritimeknowledge.blogspot.com/2015/12/ibsintegrated-bridge-system.html>. - abgerufen am 2017-07-15
- [SKKR11] SANDURAWAN, DAMITHA ; KODIKARA, NIHAL ; KEPPITIYAGAMA, CHAMATH ; ROSA, REXY: A Six Degrees of Freedom Ship Simulation System for Maritime Education. In: *International Journal on Advances in ICT for Emerging Regions (ICTer)* Bd. 3 (2011), Nr. 2, S. 34
- [SmSi11] SMITH, DAVID JOHN ; SIMPSON, KENNETH G. L.: *Safety critical systems handbook: a straightforward guide to functional safety: IEC 61508 (2010 edition) and related standards; including: process IEC 61511, machinery IEC 62061 and ISO 13849*. 3. Amsterdam : Elsevier/Butterworth-Heinemann, 2011 — ISBN 978-0-08-096781-3

- 
- [Somm12] SOMMERVILLE, IAN: *Software engineering, Always learning*. 9. München Harlow Amsterdam : Pearson, 2012 — ISBN 978-3-86894-099-2
- [StHM08] STATHEROS, THOMAS ; HOWELLS, GARETH ; MAIER, KLAUS McDONALD: Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. In: *Journal of Navigation* Bd. 61 (2008), Nr. 01
- [SVEH12] STAHL, THOMAS ; VÖLTER, MARKUS ; EFFTINGE, SVEN ; HAASE, ARNO: *Modellgetriebene Softwareentwicklung Techniken, Engineering, Management*. Heidelberg : Dpunkt.verlag, 2012 — ISBN 978-3-89864-881-3
- [SWPJ00] SLOTSCH, OSCAR ; WILDMOSER, MARTIN ; PHILIPPS, JAN ; JESCHULL, REINHARD ; ZALMAN, RAFAEL: ISO 26262 - Tool Chain Analysis Reduces Tool Qualification Costs. In: *Automotive - Safety & Security 2012*. Karlsruhe, S. 12
- [Szła12] SZŁAPCZYŃSKI, RAFAŁ: Evolutionary approach to ship's trajectory planning within Traffic Separation Schemes. In: *Polish Maritime Research* Bd. 19 (2012), Nr. 1
- [Tass14] TASS: *Cooperative Mobility - Test facilities for cooperative mobility*. URL <https://tass.plm.automation.siemens.com/cooperative-mobility>. - abgerufen am 2018-01-29. — TASS International
- [TBSS06] TALLANT, GREGORY S. ; BUFFINGTON, JAMES M. ; STORM, WALTER A. ; STANFILL, PETER O. ; KROGH, BRUCE H.: *Validation & verification for emerging avionic systems*. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.538.688&rep=rep1&type=pdf>. - abgerufen am 2016-08-22. — Validation & Verification for Emerging Avionic Systems
- [ThFu00] THOMKE, STEFAN ; FUJIMOTO, TAKAHIRO: The Effect of „Front-Loading“ Problem-Solving on Product Development Performance. In: *Journal of Product Innovation Management* Bd. 17 (2000), Nr. 2, S. 128–142
- [Tolk13] TOLK, ANDREAS: Interoperability, Composability, and Their Implications for Distributed Simulation: Towards Mathematical Foundations of Simulation Interoperability. In: *IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*. Delft : IEEE, 2013 — ISBN 978-0-7695-5138-8, S. 3–9
- [Tsou10] TSOU, MING-CHENG: Integration of a Geographic Information System and Evolutionary Computation for Automatic Routing in Coastal Navigation. In: *Journal of Navigation* Bd. 63 (2010), Nr. 02, S. 323
- [ULRC17] URIAGEREKA, GARAZI JUEZ ; LATTARULO, RAY ; RASTELLI, JOSHUE PEREZ ; CALONGE, ESTIBALIZ AMPARAN ; RUIZ LOPEZ, ALEJANDRA ; ESPINOZA ORTIZ, HUASCAR: Fault injection method for safety and controllability evaluation of automated driving. In: *IEEE Intelligent Vehicles Symposium (IV) 2017*. Los Angeles : IEEE, 2017 — ISBN 978-1-5090-4804-5, S. 1867–1872
- [USDT13] USLAR, MATHIAS ; SPECHT, MICHAEL ; DANEKAS, CHRISTIAN ; TREFKE, JORN ; ROHJANS, SEBASTIAN ; GONZALEZ VAZQUEZ, JOSÉ M. ; ROSINGER, CHRISTINE ; BLEIKER, ROBERT: *Standardization in smart grids: introduction to IT-related methodologies, architectures and standards, Power systems*. Berlin ; New York : Springer, 2013 — ISBN 978-3-642-34915-7

- [Vanb08] VANBRABANT, ROBBIE: *Google Guice: agile lightweight dependency injection framework, firstPress*. Berkeley : Apress, 2008 — ISBN 978-1-4302-0863-1
- [VeBK10] VERMASSEN, ERWIN ; BECKMANN, DIRK ; KÖSTER, FRANK: The ITS Test Beds Project. In: *17th ITS World Congress*. Busan : The National Academies of Sciences, Engineering, Medicine, 2010
- [ViWi12] VIDELA, ALVARO ; WILLIAMS, JASON J. W.: *RabbitMQ in Action: Distributed Messaging for Everyone*. 1. Shelter Island : Manning Publications, 2012 — ISBN 978-1-935182-97-9
- [VWFG13] VALENTINI, UWE ; WEIßBACH, RÜDIGER ; FAHNEY, RALF ; GARTUNG, THOMAS ; GLUNDE, JÖRG ; HERRMANN, ANDREA ; HOFFMANN, ANNE ; KNAUSS, ERIC ; HERRMANN, A. ; KNAUSS, E. ; WEIßBACH, R. (Hrsg.): *Requirements Engineering und Projektmanagement, Xpert.press*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013 — ISBN 978-3-642-29431-0
- [WAGP08] WARD, ROBERT ; ALEXANDER, LEE ; GREENSLADE, BARRIE ; PHARAOH, ANTHONY: *IHO S-100: The New Hydrographic Geospatial Standard for Marine Data and Information*. URL [http://scholars.unh.edu/ccom/425/?utm\\_source=scholars.unh.edu%2Fccom%2F425&utm\\_medium=PDF&utm\\_campaign=PDFCoverPages](http://scholars.unh.edu/ccom/425/?utm_source=scholars.unh.edu%2Fccom%2F425&utm_medium=PDF&utm_campaign=PDFCoverPages). - abgerufen am 2017-10-04
- [Wass05] WASSON, CHARLES S.: *System analysis, design, and development: concepts, principles, and practices*. Hoboken : Wiley-Interscience, 2005 — ISBN 978-0-471-39333-7
- [WaTW09] WANG, WENGUANG ; TOLK, ANDREAS ; WANG, WEIPING: The levels of conceptual interoperability model: applying systems engineering principles to M&S. In: *Proceedings of the 2009 Spring Simulation Multiconference* : Society for Computer Simulation International, 2009, S. 168
- [WeHN16] WEINERT, BENJAMIN ; HAHN, AXEL ; NORKUS, OLIVER: A domain-specific architecture framework for the maritime domain. In: *Lecture Notes in Informatics (LNI)*. Bonn, 2016, S. 12
- [Weil07] WEILKIENS, TIM: *Systems engineering with SysML/UML: modeling, analysis, design*. Amsterdam; Boston : Morgan Kaufmann OMG Press/Elsevier, 2007 — ISBN 978-0-12-374274-2
- [WHAD07] WIDERGREN, STEVEN E. ; HARDIN, DAVE ; AMBROSIO, RON ; DRUMMOND, R. ; GUNTHER, E. ; GILCHRIST, GRANT ; COHEN, DAVID: *Interoperability Context-Setting Framework* : Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2007
- [WiWJ16] WINNER, HERMANN ; WACHENFELD, WALTHER ; JUNIETZ, PHILIPP: *Safety Assurance for Highly Automated Driving - The PEGASUS Approach*. URL <http://www.pegasus-projekt.info/files/tmpl/pdf/Automation%20Vehicle%20Symposium%202016%20Folien.pdf>. - abgerufen am 2018-03-16
- [WRFH15] WALDEN, DAVID D. ; ROEDLER, GARRY J. ; FORSBERG, KEVIN ; HAMELIN, R. DOUGLAS ; SHORTELL, THOMAS M.: *INCOSE: Systems engineering handbook: a guide for system life cycle processes and activities*. 4. Hoboken : Wiley, 2015 — ISBN 978-1-118-99940-0

- 
- [YIKG13] YLIÈS, FALCONE ; KLAUS, HAVELUND ; GILES, REGER: A Tutorial on Runtime Verification. In: *NATO Science for Peace and Security Series, D: Information and Communication Security* (2013), S. 141–175
- [Zell09] ZELLER, ANDREAS: *Why programs fail: a guide to systematic debugging*. 2. Amsterdam ; Boston : Elsevier/Morgan Kaufmann, 2009 — ISBN 978-0-12-374515-6
- [Zhan15] ZHANG, DIAN: *A multi-modal smart sensing network for marine environmental monitoring*. Dublin, Dublin City University. School of Electronic Engineering, 2015
- [ZiAV04] ZIADE, HAISSAM ; AYOUBI, RAFIC A. ; VELAZCO, RAOUL: A survey on fault injection techniques. In: *Int. Arab J. Inf. Technol.* Bd. 1 (2004), Nr. 2, S. 171–186



---

# Abbildungen

Abbildung 1: Kosten der V+V in der Entwicklung von Luftfahrtführungssystemen nach.....	2
Abbildung 2: Integriertes Brückensystem.....	5
Abbildung 3: Ingenieurwissenschaftliche Methodik der Arbeit.....	11
Abbildung 4: Drei-Schichten-Modell zur Schiffsführung nach [Dong12] und [Foss11, S.232] .....	16
Abbildung 5: V-Modell mit Front-Loading der V+V nach.....	26
Abbildung 6: Dynamische V+V in der Gestaltungsphase und vertragsbasierte Gestaltung.....	29
Abbildung 7: Vorgeschlagene Methoden der V+V für die Entwicklung von sicherheitskritischen Systemen der Vehikel-Führung nach.....	30
Abbildung 8: Abgrenzung zwischen Open-Loop-Test und Closed-Loop-Test nach.....	31
Abbildung 9: Dynamische V+V der modellbasierten Systementwicklung im Regelkreis.....	32
Abbildung 10: Nahtlose Unterstützung der V+V während der Systementwicklung durch Testfelder.....	33
Abbildung 11: Einsatz des physikalischen Testfeldes in der Systementwicklung.....	34
Abbildung 12: Referenzarchitektur eines generischen Testfeldes für automatisierte cyber- physische Systeme.....	36
Abbildung 13: Systemarchitektur des physikalischen Testaufbaus für die Erprobung des in ACCSEAS entwickelten VTS und IVEF Informationsaustauschformats.....	46
Abbildung 14: Systemarchitektur des physikalischen Testaufbaus für die Erprobung des in EfficienSea2 entwickelten Smart Buoy Systems, das als aktiv kommunizierende Komponente der Navigationshilfe und Lagebilderstellung dient.....	47
Abbildung 15: Systemarchitektur des physikalischen Testaufbaus Hermitage für die Erprobung von Anwendungen der e-Navigation.....	48
Abbildung 16: Systemarchitektur des physikalischen Testaufbaus für die Erprobung einer Anwendung für den Routenaustausch im Rahmen des Projekts MONALISA.....	49
Abbildung 17: Systemarchitektur des physikalischen Testaufbaus für die Erprobung der in SSAP entwickelten Datenplattform.....	50
Abbildung 18: Systemarchitektur des physikalischen Testfeldes der Anwendungsplattform für intelligente Mobilität für den Test automatisierter Kraftfahrzeugführungsfunktionen.....	51
Abbildung 19: Systemarchitektur des physikalischen Testfeldes DITCM für die stationäre Erfassung des Verkehrslagebilds zur Erprobung von V2X Anwendungen.....	52
Abbildung 20: Systemarchitektur des physikalischen Testfeldes ATTAS zur Erprobung von unter anderem Flugzeugführungssystemen.....	53
Abbildung 21: Genereller Überblick des Testfeldes für automatisierte Schiffsführungssysteme.....	65
Abbildung 22: Elemente eines Testfeldes für automatisierte Schiffsführungssysteme.....	66
Abbildung 23: Ganzheitliche Systemarchitektur des physikalischen Testfeldes.....	69
Abbildung 24: Interoperabilitätsmechanismen des S-100 Frameworks für den Datenaustausch und die Einordnung des Datenmodells des physikalischen Testfeldes.....	71
Abbildung 25: Schiff mit einer Position und einer MMSI modelliert als ein mit dem S-100 Framework konformes Datenobjekt.....	72
Abbildung 26: Elemente einer Middleware nach.....	73
Abbildung 27: Datenverteilung der Systemknoten des physikalischen Testfeldes über zentrale nachrichtenorientierte Middleware nach.....	75
Abbildung 28: Veröffentlichen und Abonnieren von Daten in dem physikalischen Testfeld .....	76
Abbildung 29: Dynamische und statische Integration der Komponenten in das einheitliche Datenmodell gemäß S-100.....	78

Abbildung 30: Physikalische Verteilung der Komponenten des physikalischen Testfeldes	82
Abbildung 31: Schematische Darstellung der Fehlerinjektion und Steuerung des physikalischen Testfeldes durch eine die originalen Dateneingaben manipulierende Komponente	84
Abbildung 32: Architektur des virtuellen Testfeldes HAGGIS	87
Abbildung 33: Integrierte Systemarchitektur des virtuellen und physikalischen Testfeldes	88
Abbildung 34: Simulations-Adapter für die Kopplung der HLA des virtuellen Testfeldes und nachrichtenorientierten Middleware des physikalischen Testfeldes	89
Abbildung 35: Softwarearchitektur der polymorphen Schnittstelle	92
Abbildung 36: Komponenten für die transitive Abbildung einer semantischen Annotation	94
Abbildung 37: Methode der semantischen Transformation durch die polymorphe Schnittstelle	95
Abbildung 38: Serialisierung und Deserialisierung von S-100 Datenobjekten mit der Hilfe des EMF	105
Abbildung 39: Technologien zur Umsetzung der polymorphen Schnittstelle	106
Abbildung 40: Paketdiagramm des Aufbaus der polymorphen Schnittstelle in LABSKAUS	107
Abbildung 41: Stark vereinfachte Klassen für das Laden von Filtern in der polymorphen Schnittstelle	108
Abbildung 42: Implementierung von Filterkomponenten gemäß der API der polymorphen Schnittstelle	108
Abbildung 43: Klassendiagramm der semantischen Transformation durch die polymorphe Schnittstelle	109
Abbildung 44: Aufbau der Konfiguration der polymorphen Schnittstelle	110
Abbildung 45: Vorgehen für die Evaluation der Testfeld-Architektur	114
Abbildung 46: Datenströme für die Messung der Latenz und des Datendurchsatzes der Infrastruktur von LABSKAUS	115
Abbildung 47: Latenzen der polymorphen Schnittstelle mit einer beziehungsweise drei schiffseitigen NaviBoxen als Dateneingabe. Die reduzierten Eingaben stellen die polymorphe Schnittstelle während des laufenden Betriebs exklusive der anfänglichen Latenz dar, welche durch die Laufzeitumgebung <i>Hotspot</i> von Java verursacht wird	117
Abbildung 48: Latenzen der Sensordatenextraktion und des Befehlsempfangs der mobilen NaviBox auf dem Forschungsboot ZUSE über die Zeit	119
Abbildung 49: Datenübertragungsrate der mobilen NaviBox auf dem Forschungsboot ZUSE	120
Abbildung 50: Datenübertragungsrate einer stationären NaviBox der Referenzwasserstraße in Brunsbüttel in Kilobyte/s	121
Abbildung 51: Übertragungsgeschwindigkeit der LTE Kommunikationsverbindung der stationären NaviBoxen der Referenzwasserstraße von LABSKAUS	122
Abbildung 52: Übertragungsgeschwindigkeit der LTE Kommunikationsverbindung der mobilen NaviBoxen auf dem Forschungsboot ZUSE	122
Abbildung 53: Aufbau des physikalischen Testfeldes für die explorative Datensammlung zur Unterstützung der Beobachtung von Annahmen, Bedingungen und Modellbildung für frühe Systementwicklungsphasen	125
Abbildung 54: Stationäre NaviBoxen der Referenzwasserstraße von LABSKAUS in Cuxhaven (links) und Brunsbüttel (rechts)	126
Abbildung 55: Testfeld-Aufbau von LABSKAUS für die explorative Datenerhebung für die Szenarien-Datenplattform	127
Abbildung 56: Forschungsboot ZUSE (links) und mobile NaviBox an Bord der ZUSE (rechts) für die Verifikation der durch die Szenarien-Datenplattform identifizierten Manöver	127

---

Abbildung 57: Ausschnitt der explorierten Daten des AIS und Radarsystems der LABSKAUS Infrastruktur in der deutschen Bucht für die Szenario-Datenplattform.....	128
Abbildung 58: Aufbau des physikalischen Testfeldes für das dynamische Testen im Open-Loop Verfahren .....	132
Abbildung 59: Testfeld-Aufbau von LABSKAUS für den Test im Open-Loop-Verfahren des MTCAS.....	133
Abbildung 60: Integrationskonfiguration der polymorphen Schnittstelle für MTCAS.....	134
Abbildung 61: Diagnose-Software des MTCAS für den Systemintegrationstest in LABSKAUS.....	136
Abbildung 62: MTCAS Anwendung während des szenariobasierten Systemtests im Open-Loop-Verfahren auf dem Testträger ZUSE. Experte des MTCAS überprüft die korrekte Funktionserfüllung während der Testträger die notwendigen Teststimuli durch eine mobile NaviBox erfasst und an das MTCAS sendet. ....	136
Abbildung 63: Aufbau des physikalischen Testfeldes für das dynamische Testen im Closed-Loop-Verfahren .....	142
Abbildung 64: Testfeld-Aufbau von LABSKAUS für die Erprobung des hochautomatisierten Schiffsführungssystems des OFFIS im Closed-Loop-Verfahren .....	143
Abbildung 65: Physikalische Verteilung des szenariobasierten Systemtests des hochautomatisierten Schiffsführungssystems im Closed-Loop-Verfahren .....	145
Abbildung 66: Konfiguration der polymorphen Schnittstelle für die dynamische Integration des OFFIS Systems.....	146
Abbildung 67: Definition eines Testszenarios in HAGGIS zur Erprobung des hochautomatisierten Schiffsführungssystems des OFFIS während einer Schiffsbegegnung mit kreuzenden Kursen .....	149
Abbildung 68: Ausschnitte aus der Benutzeroberfläche für V+V MATEControl für die Beobachtung und Steuerung des physikalischen Testfeldes LABSKAUS.....	150
Abbildung 69: Hochautomatisiertes Schiffsführungssystem des OFFIS während des szenariobasierten Systemtests im Closed-Loop-Verfahren. ....	151

# Tabellen

Tabelle 1: Stufen für automatisierte Schiffsführungsfunktionen.....	14
Tabelle 2: Technologieübersicht der Kommunikationsmuster.....	75
Tabelle 3: Interoperabilitätskollisionen und mögliche Verluste.....	96
Tabelle 4: Leistungsdatenerhebung hinsichtlich der auftretenden Latenz von LABSKAUS .....	116

# Anhang

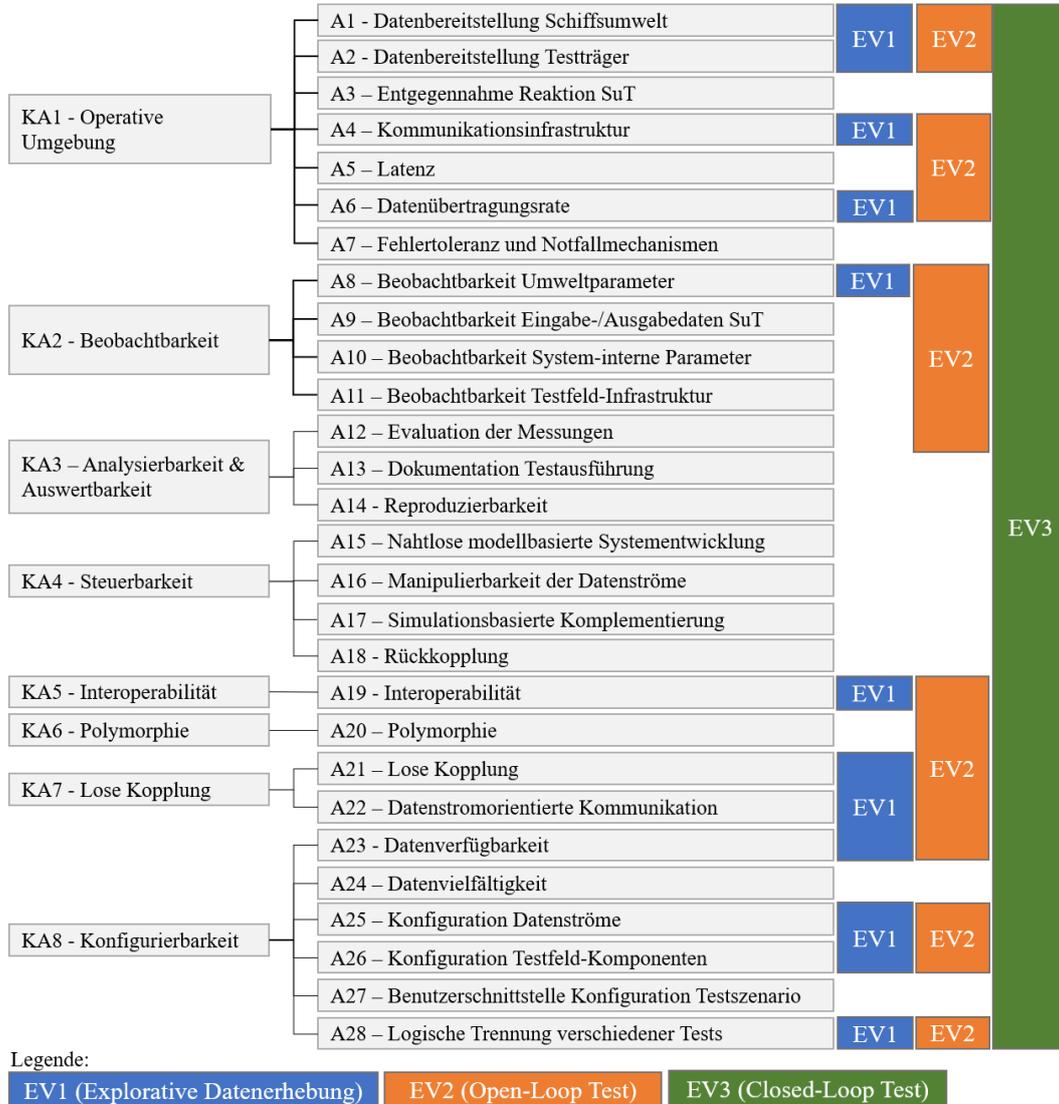
## A Identifikation und Auswahl der maritimen Testaufbauten

Testaufbau/Projekt	SA	Fokus	Doku	Beschreibung	Zeitraum
ACCSEAS	x	x	x	Nordsee: Vertikaler Testaufbau zur Erprobung von Resilient PNT und E-Navigationsdiensten	2012 - 2015
AMSA VDES	x			Australien: Testaufbau für ein neues UKW Datenaustauschsystems (VDES) zwischen einem landseitigen und zwei schiffsseitigen Systemen	2015 - 2015
ARCTIC WEB			x	Operative Web Service zur Überwachung der arktischen und antarktischen Region	2015 - 2017
ARIADNA	x			Daten der Navigation, Position und Schiffscharakteristiken erfassen und kombinieren, um ein kollaboratives Navigationsnetzwerk zu erstellen.	2010 - 2013
AVANTI/PRONTO				Webanwendung im Hafenbereich von Rotterdam für Port Call Optimierung	2015 - 2016
BALTCOAST	x			Ostsee: Studien für Integration menschlicher Aktivitäten in Küstenanwendungen	2015 - 2018
CASCADE	x			Methoden für den Informationsaustausch und Darstellung auf einer Schiffsbrücke	2012 - 2015
DATACRON	x			Testaufbau für die Zusammenführung von Beobachtungsdaten der Luftfahrt und maritime Domäne für Lagebildstellung	2013 - 2019
DUBIN BAY DIGITAL DIAMOND	x	x		Sensoren und Kommunikationstechnologie in den Gewässern in der Nähe von Dublin für E-Navigation Technologieerprobung	2013 - 2015
E-ATON JTCD	x			Küstenseitige AIS Auswertung zur Verbesserung der Schifffahrt	2013 - 2015
EFARWAY	x			Entwicklung technischer Lösungen für verteilten Informationsaustausch von Fahren	2013 - 2015
EFFICIENSEA	x	x		Ostsee: Kommunikationsframework küsten- und seeseitig zur Bereitstellung von maritimen Daten	2009 - 2018
EFFICIENSEA 2	x	x		Arktis + Ostsee: Kommunikationsframework küsten- und seeseitig zur Bereitstellung von maritimen Daten	2015 - 2018
E-FREIGHT	x			Framework für Prozesse zur Echtheit Warenverfolgung	2010 - 2013
EMAR	x			Standardisierung der Kommunikation in der EU für Schiffsdienste (SOA)	2012 - 2014
ENSI	x			Erprobung von AIS in einer Referenzstrecke in den USA	2015 - 2017
ENSI	x			Finnischer Golf: Routen-Prüfungsdienst + VTS zur Verbesserung der Schiffsverkehrssicherheit	2015 - 2018
ESABALT	x			Ostsee: Implementierung einer Softwareplattform für einen länderübergreifenden maritimen Informationsaustausch	2010 - 2017
E-SENS (Testaufbau: "Minder")	x			"Minder" ist Testaufbau zur Verifikation der Konformität von Systemen gemäß Spezifikationen/Standards digitaler Behördendienste	2013 - 2017
EUCISE2020	x			Informationsaustausch zur Überwachung zwischen den Behörden der europäischen Staaten (CISE)	2015 - 2017
E-YANGSHAN PORT	x			Ostchinesische See: Datenbereitstellung zur Verbesserung der Navigation anhand des Anwendungsfalls schlechter Sicht im Hafenbereich	2014 - 2016
FAROS	x			Menschliche Faktoren in das Risiko-Basierte Design von Schiffen zu integrieren	2012 - 2015
FLAGSHIP	x			Sicherheit, Umweltfreundlichkeit und die Wettbewerbsfähigkeit des europäischen maritimen Transports verbessern.	2010 - 2011
GREAT LAKES*	x			USA/Kanada: Bereitstellung eines Breitband-Kommunikationsnetzes zur Datenübertragung maritimer Daten	2010 - 2011
HERMITAGE	x	x		Ostlicher Finnischer Golf/Russland: AIS Infrastruktur in Referenzstrecke für E-Navigationssysteme	2016 - 2020
INTELLIGENT FAIRWAY				Bathymetrische Modelle des Seebodens werden mit Umweltinformationen verknüpft	2017 - 2018
IONIO	x			Adria/Ionische See: Marine Observatorium für Umweltinformationen und darauf basierende Dienste wie Routenoptimierung	2012 - 2013
MARNIS	x			Testaufbau zur Überprüfung der Ergebnisse von MARNIS, AIS, GALILEO, ECDIS Informationsmanagement für sichere Schifffahrt	2004 - 2008
MICE (MONA LISA)	x			Arktis: Datenaustausch zwischen Küste und Schiff für Routenoptimierung z.B. durch Weiterdienste	2013 - 2014
MONALISA 1.0	x			Ostsee: Tests von dynamischen und proaktiven Routenplanungsdiensten	2010 - 2013
MONALISA 2.0	x			Ostsee & Mittelmeer: Sea-Traffic-Management für Routen- und Navigationsdienste	2013 - 2015
MEH	x			Straße von Malakka: Netzwerk aus maritimen Informationstechnologien für verbesserte Sicherheit und Umweltschutz auf See.	2010 - 2012
MUNIN	x			Simulative Überprüfung von Konzepten für autonome Schiffe zu entwickeln und zu überprüfen	2013 - 2015
NORSAT 2	x			Realisierbarkeit von UKW-Datenaustausch über Satellit in einer realen Betriebsumgebung demonstrieren	2015 - 2016
NORWEGIAN E-NAVIGATION TRIAL	x			Navor Systemtest des digitalen Routenaustausch eines Schiffes mit einer Routenplanungssoftware (Navstation)	2014 - 2016
POLAR ICE	x			Arktis & Antarktis: Entwickelt Meer-Eis-Informationssysteme um bestehende Lücken in der Arktis und Antarktis zu schließen.	2017 - 2018
SBAS <sup>3</sup>	x			Australien & Neuseeland: Satellitensystem für Positionsbestimmung	2014 - 2017
SESAME STRAITS	x	x		Straße von Malakka: Test eines Systems für das kooperative Entscheidungsfinden zwischen Schiff-Brücken-Team und dem Küstenpersonal	2015 - 2018
SHEBA/STORMWINDS	x			Aktuelle und zukünftige Verkehrsaufkommen analysieren und daraus Emissionen und Lärm berechnen	2010 - 2012
SKEMA	x			Wissensmanagementplattform für Entscheidungsträger der Politik und anderer Stakeholdergruppen der See- und Logistikindustrie	2013 - 2020
Smart Navigation Project	x			Korea: E-Navigationsdienste entwickeln und testen	2012 - 2015
SSAP	x			Testaufbau stellt diverse maritime Sensordaten bereit, um E-Navigation Anwendungen zu realisieren und testen	2012 - 2015
STM VALIDATION	x			Europa: Infrastrukturaufbau in einem Testaufbau für Anwendungsvalidierung von Mice, Monalisa I und Monalisa II	2015 - 2018
Tianjin Port e-Navigation	x			China: Informationsbereitstellung für E-Navigationsdienste und Anwendungsvalidierung	2015 - 2017
TORRES STRAITS	x			Australien: Datensammlung für Bewältigung des erhöhten Schiffsverkehrs für den Umweltschutz	2012 - 2015
WINMOS	x			Entwicklung eines Eisbrecher-Management-Systems	
*GREAT LAKES - ST LAWRENCE SEAWAY SYSTEM					

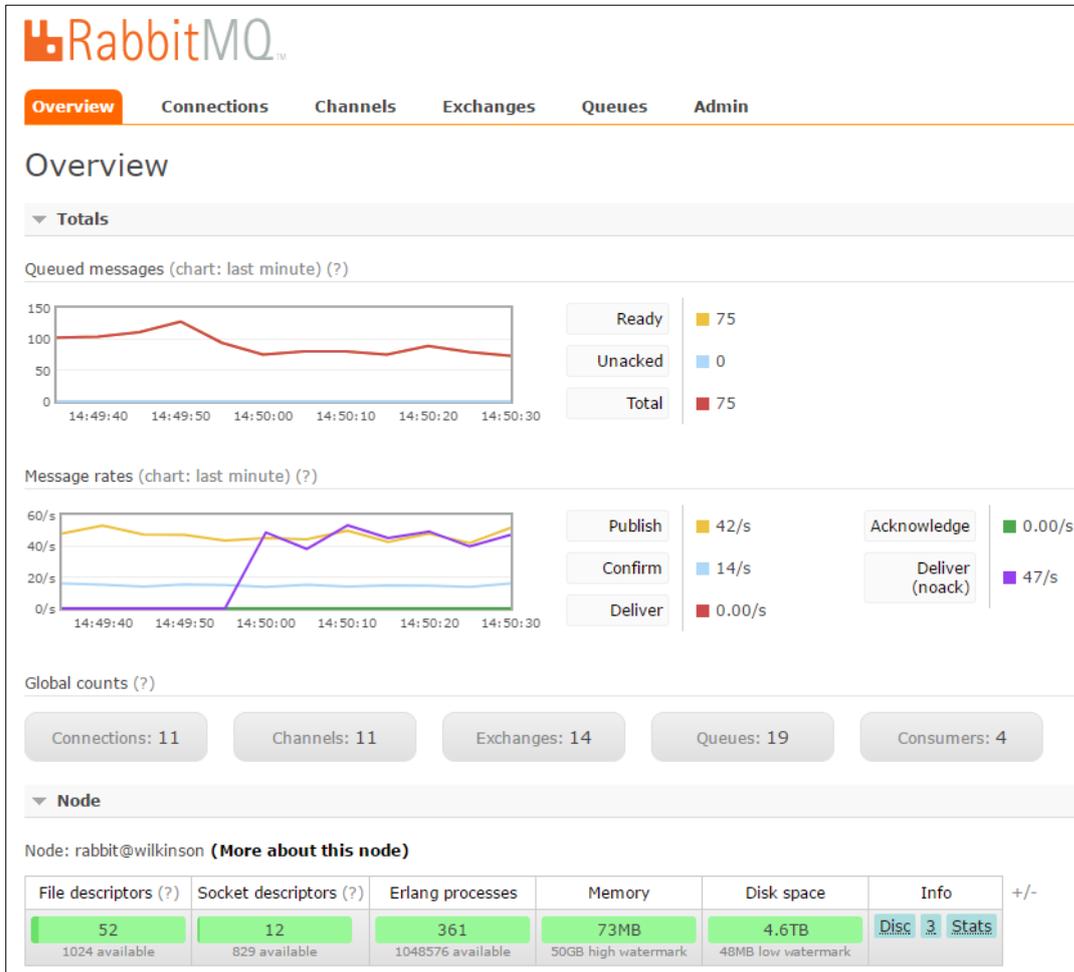
## B Ableitung von Kernanforderungen für die Untersuchung der verwandten Arbeiten



## C Anforderungsverifikation des Lösungsansatzes durch die LABSKAUS Einsatzszenarien



## D Weboberfläche der zentralen nachrichtenorientierten Middleware



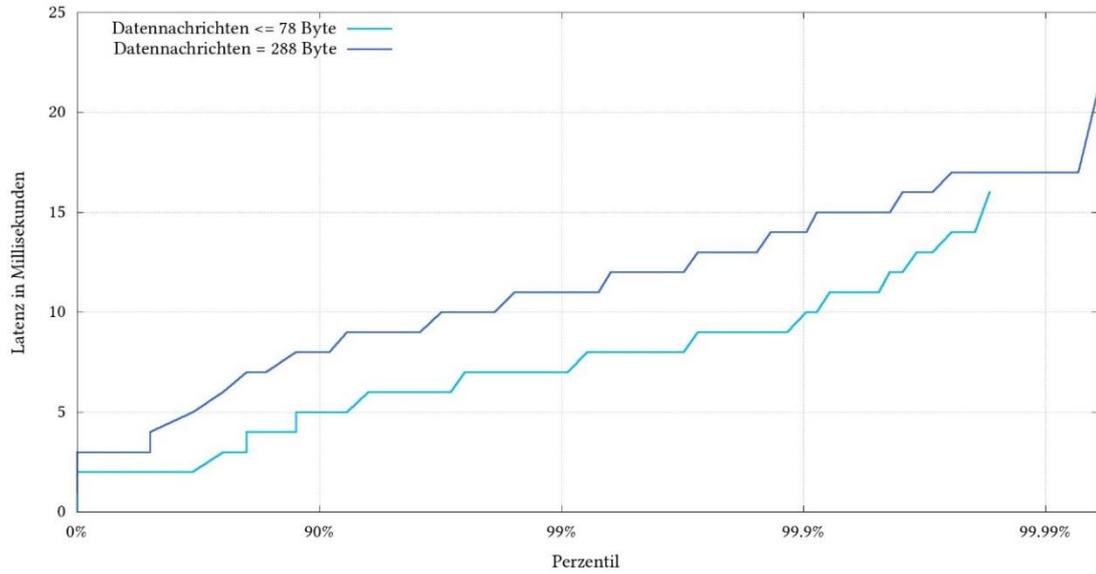
---

## E Implementierte Filter in Java und Transformationsregelsätze in XSLT

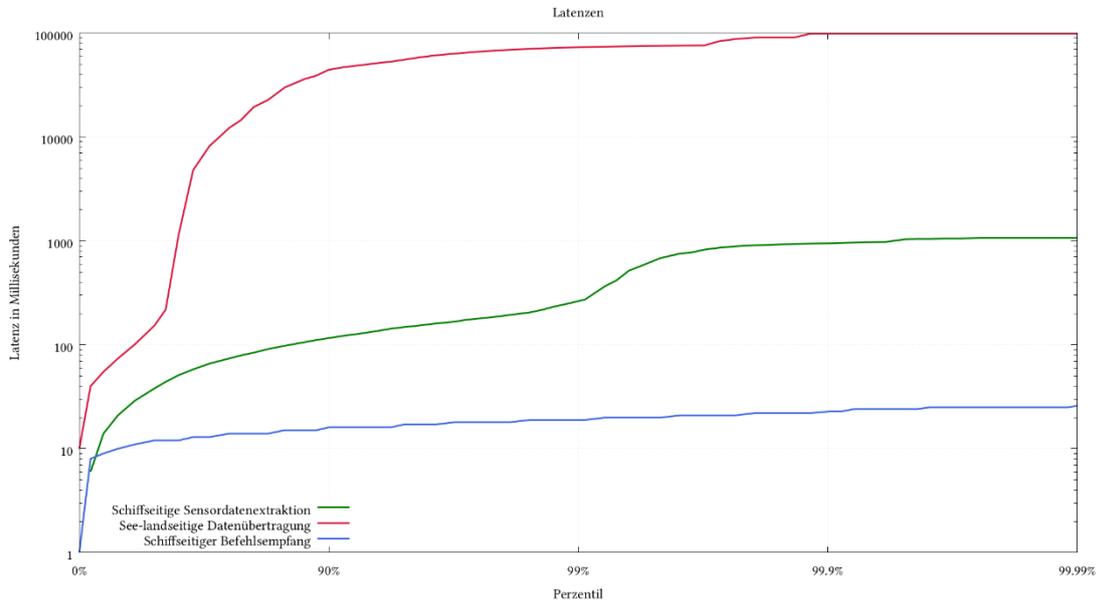
<i>Kommunikation</i>	<i>Syntax</i>	<i>Semantik</i>
RabbitMQInputFilter RabbitMQOutputFilter	S100XMLInputFilter S100XMLOutputFilter	NMEA2000-to- NMEA0183.xsl
HttpInputFilter HttpOutputFilter	NMEA2000XMLInputFilter NMEA2000XMLOutputFilter NMEA2000ZUSECommandXMLOutput- Filter	NMEA2000-to-S100.xsl S100-to-NMEA2000.xsl
UdpInputFilter UdpOutputFilter	NMEA0183XMLOutputFilter	RTZ-to-NMEA2000.xsl
FileInputFilter FileOutputFilter	XMLJsonOutputFilter	S100-to-MTCAS.xsl
WebSocketOutputAdapter		



## G Latenz der polymorphen Schnittstelle bezüglich der Randfälle der Nachrichtengröße

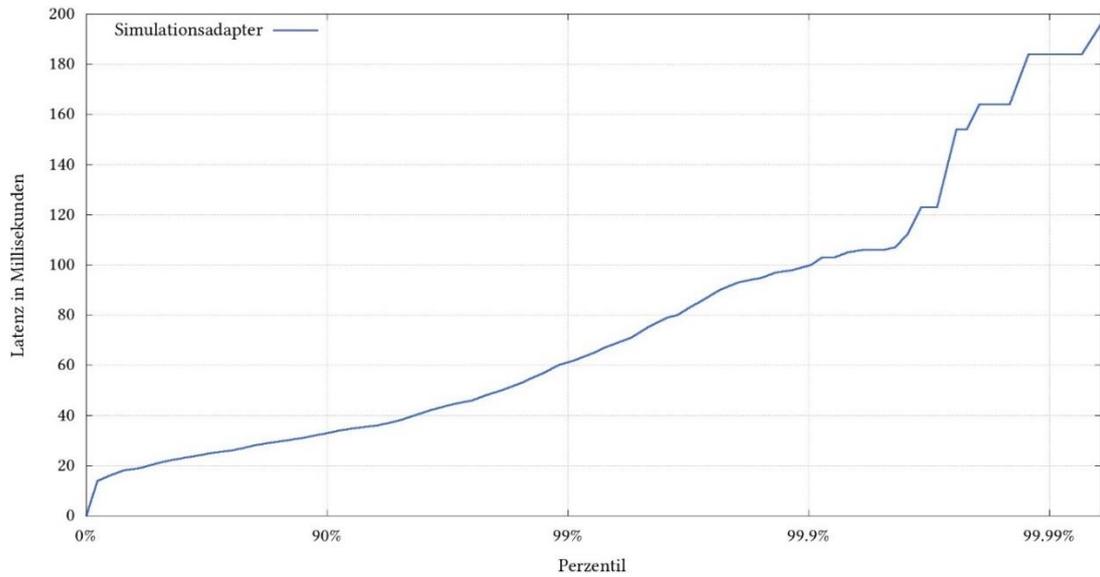


## H Histogramm der Latenzen der Sensordatenextraktion und des Befehlsempfangs



---

# I Latenz des Simulations-Adapters im Falle der Anreicherung von LABSKAUS durch simulativ erzeugte AIS und Radardaten durch das virtuelle Testfeld HAGGIS



## Eidesstattliche Erklärung

Hiermit versichere ich, Marius BRINKMANN, dass ich die von mir vorgelegte Dissertation mit dem Titel „Physikalische Testfeld-Architektur für die Unterstützung der Entwicklung von automatisierten Schiffsführungssystemen“ selbständig verfasst habe, dass ich alle von anderen Autoren wörtlich übernommenen Stellen, wie die sich an die Gedankengänge anderer Autoren eng anliegenden Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

Oldenburg, den 29. Oktober 2018



---

Marius Brinkmann