

2015, 44 (116), 116–122  
ISSN 1733-8670 (Printed)  
ISSN 2392-0378 (Online)  
DOI: 10.17402/065

Received: 31.08.2015

Accepted: 06.11.2015

Published: 07.12.2015

## Virtual test bed for maritime safety assessment

Axel Hahn<sup>1</sup>✉, Volker Gollücke<sup>1</sup>, Carsten Buschmann<sup>1</sup>, Sören Schweigert<sup>2</sup>

<sup>1</sup> Universität Oldenburg

26111 Oldenburg, Germany, e-mail: {hahn; golluecke; buschmann}@wi-ol.de

<sup>2</sup> OFFIS – Institute for Information Technology

2 Escherweg, 26121 Oldenburg, Germany, e-mail: soeren.schweigert@offis.de

✉ corresponding author

**Key words:** virtual test bed, co-simulation, e-Navigation, maritime traffic simulation, safety assessment, sensor simulation, hardware in the loop, software in the loop, model in the loop

### Abstract

“Safe voyage from berth to berth”: This is the goal of all e-Navigation strains driven by new technologies, new infrastructures, and new organizational structures on bridge, on shore, as well as in the cloud. To facilitate these efforts suitable engineering and safety/risk assessment methods are required. Understanding maritime transportation as a sociotechnical system allows the usage of system-engineering methods. Simulation-based test beds for verification and validation (V +V) of e-Navigation technologies are important methods to obtain functional safety and reliability. The modeling and simulation toolset HAGGIS is a co-simulation system for the evaluation of e-Navigation concepts and systems. It provides a maritime traffic simulator and a physical world (n-body) simulator and services for finding rare events of failures. HAGGIS is accompanied by the physical test bed LABSKAUS, which implements a reference port and waterways. This paper describes an integrated and seamless approach for developing new e-Navigation technologies starting with virtual simulation-based safety assessment and ending in physical real-world demonstrations. It gives an overview of the actual test bed and introduces requirements, concepts and elements of HAGGIS and LABSKAUS, which are joined in the e-Maritime Integrated Reference Platform (eMIR) test bed.

### Introduction

Information and communication technologies are revolutionizing the world of navigation as in every other production and transportation system. These new systems require completely new ways of safety assessment and engineering methods including verification and validation technology to ensure functional safety as, for example, required by ISO 26262.

In this paper it is assumed that, to ensure safety of e-Navigation technologies, a holistic engineering approach is required, taking the whole sociotechnical system (man and machine) and its environment into account.

Based on this assumption, this paper introduces a simulation environment for the assessment of upcoming e-Navigation technologies focusing especially on safety and risks. These new technologies require new methods for safety assessment.

Model-driven engineering technologies support the safety analysis during the design phase by using formal analysis methods (introduced in (Läsche et al., 2014)) and simulation-based V+V and risk assessment. Therefore, this paper introduces a simulation framework named Hybrid Architecture for Granularly, Generic and Interoperable Simulations (HAGGIS). For scientific grounding and in situ experiments, the physical test bed LABSKAUS extends the simulation environment by providing experimental Vessel Traffic Services (VTS) and Bridge Systems, reference waterways and port areas.

In this paper we present the simulation environment HAGGIS to verify and validate the effects of a new e-Navigation assistance system on the outcome of the simulated example scenario. In the following sections we present an overview of the HAGGIS simulation architecture and its compo-

nents, with descriptions of the available tools for simulation-based V+V. The paper concludes with a short summary.

## HAGGIS

HAGGIS is a platform for co-simulated verification of safety and efficiency of maritime transport systems. The platform is used for the optimization/decision support and safety evaluations. It is an infrastructure for the integration of individual simulators, simulation control components, observers and analysis components. The concept of HAGGIS ensures a high degree of reusability of simulation systems and leads to an overall simulation environment used for research in the maritime transportation sector as well as a basis for further projects and subsequent development.

As part of the e-Maritime Integrated Reference Platform HAGGIS takes over the virtual simulation part of the development of new e-Maritime and e-Navigation technologies. HAGGIS is well integrated with the physical test bed LABSKAUS (further described in Stasch, Hahn & Bolles, 2014) to allow a seamless development process from virtual models to fully fledged physical demonstrations.

## Architecture

The Architecture of HAGGIS is shown in Figure 1. It can be subdivided into three parts: modeling, simulation, and analysis. An additional interoperability layer, consisting of a formally described data model and a communication infrastructure, complete the architecture.

Further, the HAGGIS System can be split into a design time or modeling part, to describe the

simulated scenario, as well as a runtime part where the simulation and (in most cases) online analysis of the research objective take place.

The **World Editor** is an Eclipse-based editor that provides a system model to allow the setting up of a static scene according to a predefined scenario. This system model contains the fundamental components/entities of all used resources, actors and environmental factors. The user is able to load 3D geometric models of ships, for example. The properties of these objects can be set according to the user's needs.

The MTS is a flexible **maritime traffic simulator** for implementing, executing and observing the behavior of multiple vessels in a realistic context. Routes for vessel traffic can be modeled or imported via National Marine Electronics Association (NMEA). Simulation of vessel movements can then be coupled with sea charts and environmental simulations.

The **Environment Simulation** provides a simulation of maritime environmental factors such as wind, current, and tide. Originally developed as part of the MTS, the environment simulation will be further developed, due to the great interest shown by industry and research, into an independent simulation within the HAGGIS framework. Thereby the Environment Simulation will be split into a modeling part, to describe the environmental scenario, and a simulation part.

A **sensor simulation** is used to generate realistic sensor measurements from a simulated context, e.g., the context of the maritime traffic simulation. To provide realistic sensor data the generated measurements can be extended by statistical, systematic or context-sensitive fault and accuracy models. Using this capability the sensor simulation

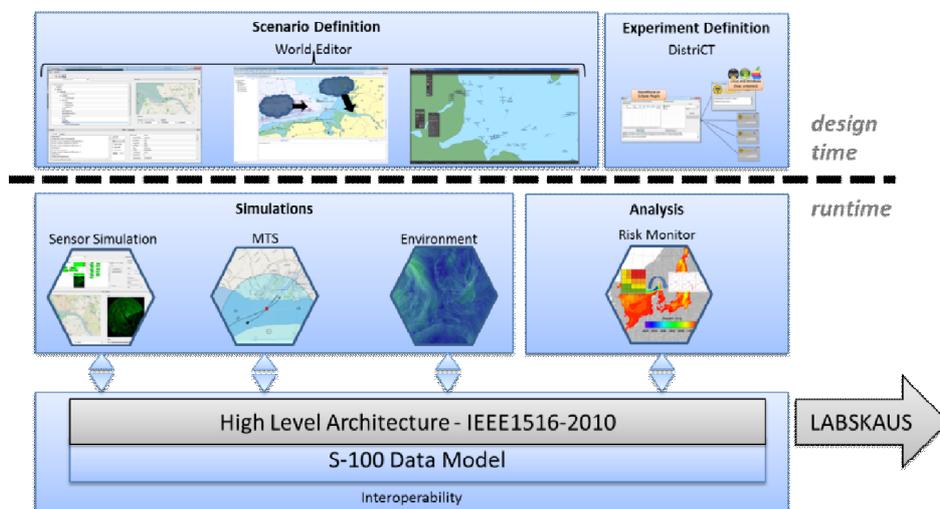


Figure 1. Overall architecture of the HAGGIS framework

plays an important role as the bridge between the virtual environment of HAGGIS and the physical test bed LABSKAUS.

To perform safety analysis on the simulated scenario, a **risk monitor** is used to determine the distance to predefined risk situations during simulation runs. The outcome of this distance estimation can then be further used to guide the simulation into a specific direction. This is done using the **Distributed Controlling Toolkit (DistriCT)** that allows the automation of simulation experiments by performing a systematic or guided parameter space exploration.

To ensure the technical interoperability of the several subsystems, we use the **High Level Architecture (HLA)** as a communication middleware. Originally developed by the US Army, to combine their combat simulations, the High Level Architecture has become an international standard for co-simulation systems approved by the IEEE under the name IEEE 1516(-2010). In the context of HAGGIS the usage of an IEEE standard allows us to easily extend our simulations with external simulations, supporting the same standard. At the same time, the HLA takes over important tasks for the co-simulations such as the synchronization of different simulations as well as the management of the simulation time.

While this section gives just a short overview, the relevant components for the introduced scenario will be further described in the following sections.

## HAGGIS Data Model

To ensure interoperability within HAGGIS a common semantic model is used for the data interchanged by the simulation components to ensure the consistence of the simulation models. The semantic model is structured in five layers (Table 1) and based on the S100 standard.

**Table 1. Five semantic model layers**

Content	Name
Ecore, OWL, ...	0: Languages
Graph, Math, Geometry, ...	1: Universal
Physics, Geography, ...	2: Engineering
Traffic System, Vehicles, Sense, Environment, HMI, ...	3: Application
Maritime, ...	4: Domain

The basic layer covers all languages, which are necessary to describe the semantic model. The concepts and data types are described using the meta model Ecore which is based on the Meta Object Facility standard (Steinberg et al., 2008).

The layers one to four are comprised of model packages. Each model package encapsulates topic

specific concepts and data types and follows the design rationale high coherence. This simplifies the exchangeability and maintainability of model packages if, for example, a new standard has to be considered. The *Universal* layer comprises concepts and data types which have a general meaning for all layers below. The *Engineering* layer includes model packages of physical and geographic concepts. The model package *Physics*, for example, contains concepts to describe a two- or three-dimensional world. The package *Geography* comprises concepts and data types derived from the S100 spatial schema. The *Application* layer includes all concepts and data types to integrate different applications like sensor simulations and cognitive simulations. It is made up of the model packages *Traffic System*, *Vehicles*, *Sense*, *Human Machine Interaction* (HMI) and *Environment*. The model package *Sense* for example, is used to integrate sensor systems into the simulation to enable Hardware/Software/Model in the Loop experiments. For this purpose it includes models to describe observations of sensor devices. The *Human Machine Interaction* model package includes concepts and data types like Display and Information Object to be able to integrate *Cognitive Simulations* (Lüdtke et al., 2009).

The *Domain* layer consists of model packages of different domains such as maritime, aviation and automotive. The model package *Maritime* provides additional concepts and data types such as specific vessel types and sea marks.

## Maritime Traffic Simulation

The Maritime Traffic Simulation (MTS) handles the analyses of maritime systems for efficiency and safety of the worldwide shipping traffic. The simulation environment is used for implementing, executing and observing the behavior of multiple ships. Therefore, developers can be supported when testing new technologies. One example is the COSINUS assistance system, which works in a virtual but still realistic and controllable context. The MTS is built as a multi-agent system to provide an extendable and flexible test bed. In addition the MTS is integrated into the HAGGIS framework by utilizing the HLA communication infrastructure and taking advantage of additional components such as the environment simulation.

An architectural overview of the MTS is given in Figure 2. The internal architecture of the MTS (upper part of Figure 2) follows the same approach as the HAGGIS framework; to be composed of different components. Within this context, the main components are *World* and *Ships*. The *Visualization*

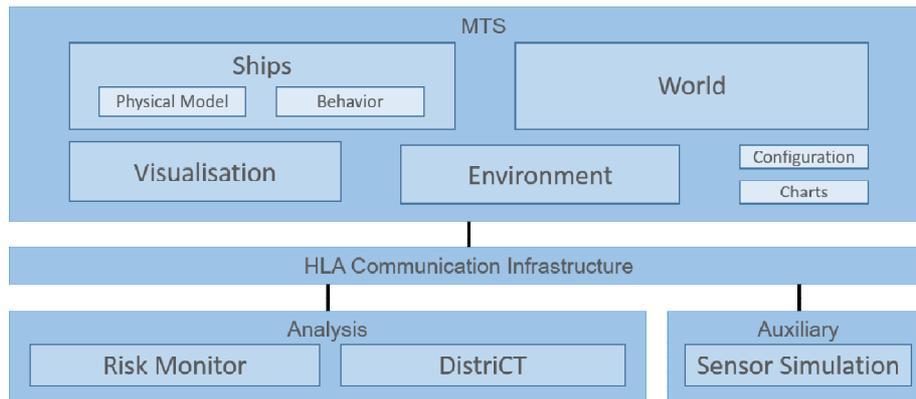


Figure 2. Architecture overview of the maritime traffic simulation (MTS)

as well as the *Environment* can be appended to a maritime traffic simulation, if necessary. The *Configuration* as well as the *Charts* component serve as data input generators. Thereby the *Configuration* component is used to configure a vessel as well as its behavior within a certain scenario. The *Charts* component is used to provide valid nautical charts to either the world component or to specific vessels. This information may be used to do a grounding analysis based on the shape of a vessel and the nautical charts inside the world component. They can also be used for the route and trajectory planning from the vessel agents.

The *World* component serves as a representation of the actual surroundings. It can be seen as the ground truth of the applied scenario. The *World* takes the vessels from the *Ships* component as well as the environmental factors of the *Environment* as input. In addition the *World* component can be configured to contain additional objects, such as buoys and other navigation marks, or irregular objects such as, life rafts or diving containers.

The *Ships* component contains a representation of one or more vessels, which in turn is divided into a physical and a behavioral model. The *Physical Model* is used to apply the physical effects on the vessel, with respect to its surroundings. The required information is provided to the *Physical Model* through the *World* component. Examples for provided information are the local current or local wind conditions which have an impact on the vessel's movement.

The behavioral model determines the behavior of the vessel by sending appropriate steering commands to the *Physical Model*. Thereby it can take various nautical aspects into account, such as the obligation to drive on the right side of the shipping lane or evasion behavior according the COLREGS. The modular structure of the MTS architecture, in particular the separation between *World* and *Ships*, allows the examination of aspects of incomplete

knowledge about the vessel's surroundings. In addition this can be controlled for different types of behavioral models, since the *World* component is capable of controlling the amount of information, provided to the behavior.

The *Environment* simulator is used to generate environmental factors such as wind, current or tide. These factors can be either artificially generated or replay a historical situation. In the context of the MTS, the artificially generated factors can be used to investigate the impact of environmental factors on the physical or behavioral model of a vessel. On the other hand, historical data, e.g., real measurements, as provided by the BSH (*Bundesamt für Seeschifffahrt und Hydrographie* – Federal Maritime and Hydrographic Agency of Germany) and other authorities, can be used to replay historical events. This capability can be used to evaluate the impact of newly developed e-Navigation technologies on critical situations as described in the scenario.

### Sensor Simulation

The sensor simulation has been developed to serve as a bridge between the virtual test platform HAGGIS and the physical test platform LABSKAUS (see Figure 3). This is used to provide the capability of performing Hardware or Software in the Loop (HIL / SIL) experiments. In this context, the sensor simulation generates sensor readings from simulated situations as they would be measured by physical sensors in a real-world environment. However, it is also capable of generating symbolic measurements as they would be required to perform Model in the Loop (MIL) tests, to support the analyzed scenario. For this purpose, the sensor simulation can be coupled with other simulations from within the HAGGIS context, which provide the observed scenario, for example, the maritime traffic simulation of the previous section. Within the sensor simulation each sensor can be

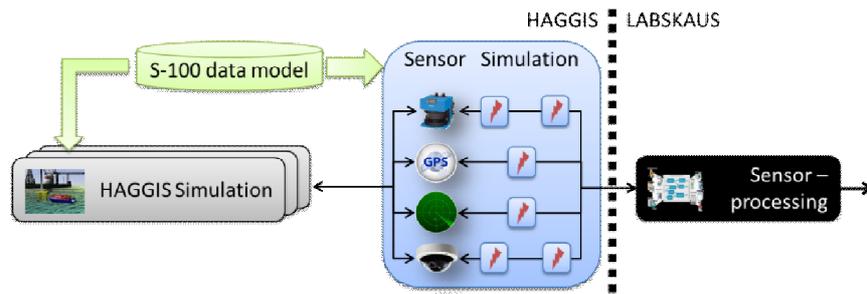


Figure 3. The sensor simulation as bridge between the virtual HAGGIS simulation platform and the physical LABSKAUS testbed

configured to generate different sensor fault and/or accuracy models. This allows considering different qualities of sensor readings during the development of new sensor processing software in the context of LABSKAUS, or to represent different knowledge about the behaviors' environment, in the context of the introduced scenario.

The communication of sensor readings can be achieved using two different approaches. In the context of HIL test cases, the sensor simulation rebuilds the original interfaces of the real sensor hardware, as far as possible. This approach has some advantages over other solutions, used in the robotic domain, which try to homogenize the communication by using a middleware. Using the original communication interfaces we are able to create sensor readings for hardware in the loop scenarios as well as software in the loop scenarios. The usage of well-known standards of the maritime domain, like NMEA0183 (NMEA, 2002) or IVEF (IALA, 2011) enables us to provide sensor readings for existing solutions as a VTS or an integrated navigation system as they are described in Stasch et al. (Stasch, Hahn & Bolles, 2014). On the other hand, symbolic measurements can be communicated using the HLA infrastructure of HAGGIS.

In contrast to similar systems often used in the robotic domain (such as Collett, MacDonald & Gerkey, 2005; Brooks et al., 2007; Carpin et al., 2007), the sensor simulation focuses on a flexible way to integrate the handling of sensor accuracy and failures. This is done by decoupling the error handling from the process of generating the actual sensor readings. The utilization of a data-stream oriented process allows a post-processing of the generated sensor measurements, depending on either a manual input, some probabilistic (error) distributions, or on the current context of the sensor (e.g. the geographic position or environmental influences).

**DistriCT**

The DistriCT tool is used for automation of simulation experiments and analysis of them. For

this purpose, simulations can be set up, distributed to different physical platforms and controlled on them. This involves the starting and stopping of simulation components as well as receiving and sending objects. In addition, the simulation components can then be automatically configured to perform a systematic parameter space exploration (Schweigert et al., 2014).

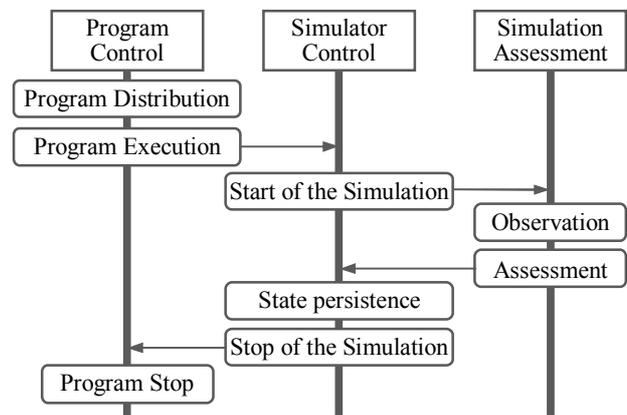


Figure 4 The three supported tasks by DistriCT and their connection with each other

Figure 4 shows the three tasks to support a co-simulation framework. The first part (*Program Control*) is used to setup the simulation components on software level, the second part (*Simulator Control*) is used to control the simulations behavior, like saving the current state of the simulation and the last part (*Simulation Assessment*) is used to access and analyze the simulation states.

The *Program Control* is divided into a client and a server part; the clients can be started on different, physically separated computer systems. The server is used to deploy the components to the clients and offers the ability to start, stop, and configure them using a central user interface.

The controlling of the simulations behavior, e.g., a parameter space exploration, is done by reacting on previously defined events which could be thrown by all participants of the co-simulation. The Simulator Control is notified about the occurrence of such an event, by following the communication

over the HLA infrastructure. A mediator, is used to ease the communication (Läsche, Gollücke & Hahn, 2013). The controlling itself follows a *Simulation Plan*, which is a graph-based representation of the control flow for each simulation participant. Such a *Simulation Plan* is constructed out of predefined nodes, while the connection between the nodes defines the order of execution. An example for such a node would be a setup node to initialize the connected simulators, or an analyzer node to evaluate the current co-simulation state.

To perform a parameter space exploration, a description is used to define the course of a simulation. It consists of parameter bounds, parameter exploration techniques, and different trigger types (i.e., simulation-time or a certain risk level) to cause parameter changes. The parameter bounds of properties are used to describe, for example, a minimum and maximum visibility range for a vessel. A parameter exploration technique can be, for example, a linear or random exploration of the parameter values. And the triggers can be active at a given simulation time or on a certain risk level. While many of the current simulation frameworks like EMSO (EMSO, 2004) or Plant Simulation (Plant Simulation, 2015) offer a parameter exploration, these solutions are tightly integrated into the respective simulation framework, while our approach is not bound to any specific simulation, but only to the communication infrastructure.

The *Simulation Assessment* part is composed of a risk monitor component and can be used in a simulation plan to determine a risk level during simulation runs. To determine the risk proximity, distance functions are used. In order to create them a methodology has been developed, which supports the selection of influent attribute values of the system (Gollücke et al., 2014). The risk monitor

allows the calculation of distances for different risky situations at the current time of a simulation run. The risk proximity can then be used, for example, to support the important splitting technique (Lagnoux, 2006), which is used to guide the simulation into potentially risky situations.

In the context of a simulation experiment the DistriCT tool is used to set up the involved simulators, the parameters, to be systematically explored as well as the distance functions. The distance functions are used to calculate the different risks, based on shared parameters of the involved simulators, for example the impact of a newly developed assistance system. Using the simulation control as well as the program control functionality of the Distributed Controlling Toolkit, a simulation run that would inevitably lead to a risky situation could be aborted immediately to shorten the time of the overall analysis.

## Application Scenario

Systems engineering design methodologies rely on a seamless model-driven approach. Starting with requirements first, concepts are described by using models as a basis for further implementation of software and hardware components. The HAGGIS Simulation framework supports the verification and validation for models (MIL), software (SIL) and hardware (HIL).

In Figure 5 the configuration and simulation cycle of the virtual test bed HAGGIS can be seen.

First applications of the HAGGIS framework are done by testing the radar data exchange between no board Electronic Chart Display and Information System (ECDIS) and on shore VTS systems to gain a better operational picture on board and on shore by using additional data.

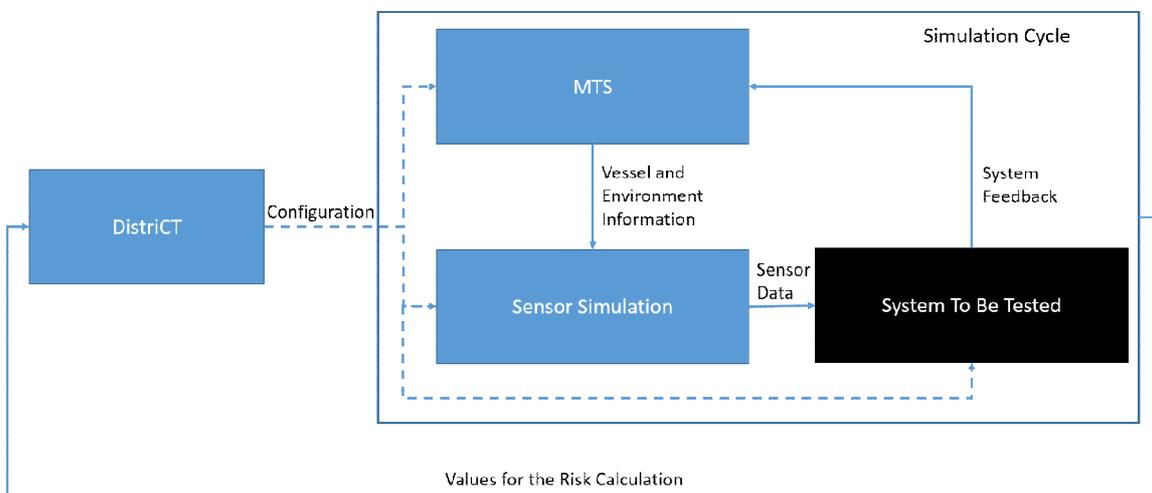


Figure 5. Configuration and simulation cycle of the virtual test bed HAGGIS

With the analysis of the requirements, test cases are defined, and are modeled by using the World Modeling tools of HAGGIS. For testing dynamic systems, executable models can be run within applications like Matlab Simulink and then tested within HAGGIS.

The HAGGIS framework, in this case, runs different maritime traffic scenarios with MTS and the sensor simulation generated radar and AIS data from these scenarios to test the software (SIL-Test). After successful testing, the software will be integrated in the ECDIS and VTS (as part of the LABSKAUS installation) and then the sensor simulation provides data from the MTS as it would be received at sea (HIL test). The tests are controlled by DistriCT. By observing the simulation runs, simulation scenarios are changed to find systematically errors within the system.

Finally, the LABSKAUS physical test environment is used for the test runs and demonstration.

## Conclusions

In this paper we present an approach to verify and validate new e-Navigation technologies based on the e-Maritime Integrated Reference Platform (eMIR).

The approach consists of a simulation-based test bed called HAGGIS which is a co-simulation system based on an HLA infrastructure as well as on a common semantic data model. It consists of a maritime traffic simulator, a sensor simulation, and services for automation of simulation experiments. We describe the usage of the simulation platform: This includes how the vessels, their behavior and dynamic models, and the surroundings can be configured and how an e-Navigation system could be automatically evaluated regarding some risk assessment functions. This is done by defining mathematical functions to measure the distance to the assessed risks. In an additional step, an exploration of parameters is set up regarding value ranges of the system under test.

The next step would be to integrate the accompanying physical test bed LABSKAUS in a software in the loop evaluation. This can easily be done by using the involved sensor simulation as a bridge between the virtual simulation part, HAGGIS and the physical test bed, as well as an implementation of the e-Navigation system.

The eMIR test bed which contains HAGGIS and LABSKAUS therefore enables an integrated and seamless approach for developing and evaluating new e-Navigation technologies starting with virtual simulation-based safety assessments and ending in physical real-world demonstrations.

## Acknowledgments

The project is funded by the German Federal State of Lower Saxony within the Research Center Critical System Engineering for Sociotechnical Systems.

## References

1. BROOKS, A., KAUPP, T., MAKARENKO, A., WILLIAMS, S. & OREBÄCK, A. (2007) Orca: a component model and repository. *Software Engineering for Experimental Robotics*. Heidelberg: Springer, pp. 231–251.
2. CARPIN, S., LEWIS, M., WANG, J., BALAKIRSKY, S. & SCRAPPER, C. (2007) *USARSim: a robot simulator for research and education*. Robotics and Automation. 2007 IEEE International Conference on. IEEE, pp. 1400–1405.
3. COLLETT, T.H., MACDONALD, B.A. & GERKEY, B.P. (2005) Player 2.0: *Toward a practical robot programming framework*. Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005). p. 145.
4. EMSO (2004) EMSO Environment for Modelling, Simulation and Optimization. EMSO Environ. Model. Simul. Optim. [Online] Available from: <http://www.vrtech.com.br/rps/emso.html> [Accessed: 14 June 2015].
5. GOLLÜCKE, V., PINKOWSKI, J., LÄSCHE, C., GERWINN, S. & HAHN, A. (2014) *Simulation-based Completeness Analysis and Adaption of Fault Trees*. SIMUL 2014, The Sixth International Conference on Advances in System Simulation. Nice, France. pp. 228–235.
6. IALA, I.A. of M.A. to Navigation and L.A. (2011) IALA Recommendation V-145 On the Inter-VTS Exchange Format (IVEF) Service Edition 1.
7. LAGNOUX, A. (2006) Rare event simulation. *Probab. Eng. Information Sci.* 20. pp. 45–66.
8. LÄSCHE, C., GOLLÜCKE, V. & HAHN, A. (2013) *Using An HLA Simulation Environment For Safety Concept Verification Of Offshore Operations*. ECMS. pp. 156–162.
9. LÄSCHE, C., PINKOWSKI, J., GERWINN, S., DROSTE, R. & HAHN, A. (2014) *Model-Based Risk Assessment of Offshore Operations*. ASME, p. V01BT01A010. doi:10.1115/OMAE2014-24018
10. LÜDTKE, A., WEBER, L., OSTERLOH, J.-P. & WORTELEN, B. (2009) Modeling Pilot and Driver Behavior for Human Error Simulation. Duffy, V. (Ed.) *Digital Human Modeling, Lecture Notes in Computer Science*. Berlin–Heidelberg: Springer. pp. 403–412.
11. NMEA (2002) National Marine Electronics Association, 0183-Standard for Interfacing Marine Electronic Devices, 3.01 ed. NMEA.
12. Plant Simulation (2015) Simul. Mit Plant Simul. [Online] Available from: [http://www.plant-simulation.de/?gclid=Cj0KEQjwzPSrBRC\\_oOXfxPWP6t0BEiQARqav2OQO8\\_Xb5B-iBS-zLZQ0TyqNfXTt7NH0KGFPEggQI8aAm-58P8HAQ](http://www.plant-simulation.de/?gclid=Cj0KEQjwzPSrBRC_oOXfxPWP6t0BEiQARqav2OQO8_Xb5B-iBS-zLZQ0TyqNfXTt7NH0KGFPEggQI8aAm-58P8HAQ) [Accessed: 14 June 2015].
13. SCHWEIGERT, S., GOLLÜCKE, V., HAHN, A. & BOLLES, A. (2014) HAGGIS: A modelling and simulation platform for e-Maritime technology assessment. Istanbul, Turkey. p. 10.
14. STASCH, A., HAHN, A. & BOLLES, A. (2014) *LABSKAUS – A physical platform for e-Maritime technology assessment*. Proceedings of 2<sup>nd</sup> International Symposium of Naval Architecture and Maritime. Istanbul, Turkey. pp. 742–752.
15. STEINBERG, D., BUDINSKY, F., PATERNOSTRO, M. & MERKS, E. (2008) *EMF Eclipse Modeling Framework*. 2<sup>nd</sup> ed. Boston: Pearson Education, Inc.