# Maritime Co-simulation Framework: Challenges and Results

**Arnold Akkermann, Bjørn Åge Hjøllo, and Michael Siegel**

## 1 Introduction

This technique, which relies on the exchange of data between separate executables, has been referred to as co-simulation or external coupling [1]. Various co-simulation realizations have different implications regarding stability, convergence, accuracy, efficiency and ease of implementation.

The maritime co-simulation is a spatially separated co-simulation between a MiL/SiL[1] based maritime runtime environment (OFFIS[2]), a physical satellite emulator (GUT[3]) and a HiL[4] own ship simulation (AVL SFR[5]).

The main task of our co-simulation is to test the navigational safety of a navigation planning system. In addition, the possibilities and limitations of this system with regard to remote control of a ship from shore will be investigated. The

[1] MiL: Model in the Loop, SiL: Software in the Loop.

[2] Institute for Information Technology (Oldenburg, Germany).

[3] Gdansk University of Technology (Gdansk, Poland).

[4] Hardware in the Loop.

[5] AVL Software and Functions GmbH.

A. Akkermann (✉) · M. Siegel
OFFIS e.V., Institut für Informatik, Oldenburg, Germany
e-mail: arnold.akkermann@offis.de; Michael.Siegel@offis.de

B. Å. Hjøllo
NAVTOR AS, Egersund, Norway
e-mail: bjorn.hjollo@navtor.com

A. Leitner et al. (eds.), *Validation and Verification of Automated Systems*,
https://doi.org/10.1007/978-3-030-14628-3_19

associated ship (hereafter referred to as own ship) must correspond in dynamics and behavior to a real ship.

In this context, OFFIS and AVL SFR have developed a maritime co-simulation which simulates the own ship including its physical behavior (AVL SFR), its sensory behavior (OFFIS) and its intelligent behavior (OFFIS). In addition, the maritime co-simulation is used to provide the surrounding traffic that could be observed from the perspective of the own ship.

Skjong et al. [2] and Sadjina et al. [3]: project Virtual Prototyping of Maritime Systems and Operations (ViProMa) developed a framework for overall maritime system design by use of distributed co-simulation technology based on FMI. They developed an open-source platform (Coral) with master/slave network communication. As the network communication is not standardized via the FMI, a proprietary interface with a slave provider is used within their framework.

Also the European Joules project[6] analyzed the applicability of FMI in the maritime domain.

Chu et al. [4] use FMUs for developing a maritime crane model. For distributed simulation they put a Java wrapper around the FMUs to export functionality via RMI.[7]

Methods for the development of platforms for real-time simulation have been extensively studied for many years. However, research in this area has recently accelerated following advances in terms of speed and the ease of development associated with new hardware platforms. Methodologies for real-time simulation have included the use of hardware, such as digital signal processors, general-purpose processors and even reconfigurable computational solutions employing FPGAs[8] [5–7].

However, there are more approaches to combine FMI with HLA for distributed co-simulation. There are two approaches, use HLA RTI as master, which is not defined in FMI or Awais et al. [8] propose two algorithms for using HLA in conjunction with FMI. One based on fixed time steps and one based on discrete event-based simulation.

Overviews of commonly used inter-process communications protocols and co-simulation frameworks can be found in Yahiaoui and Trcka [9].

A challenge is to integrate the data exchange with the internal data structures, time integration algorithms and program flow of the individual simulators.

[6] http://www.joules-project.eu/Joules/index.xhtml

[7] Remote Method Invocation.

[8] Field Programmable Gate Array.

# 2 Demonstrator Setup for Distributed Co-simulation

## *2.1 General Requirements for Co-simulation in Maritime Domain*

The development of vessels is a complex task as a vessel comprise a lot of subsystems from a wide range of engineering domains. Further, the designer must consider extreme requirements, e.g. significant power consumption, and has to find a tradeoff between costs, risk factors and environmental impacts. In addition to the difficult development, the maritime domain typically provides one-of-a-kind solutions. While the automotive or aerospace industry can invest a lot of resources in one prototype, since it lays the ground work for mass production, in the maritime industry a vessel is tailored and rarely mass produced [2].

Moreover, testing costs a lot and is time-consuming. Traffic and environmental situations required in this context are often not producible and even less reproducible. A traditional monolithic simulation approach is inflexible, costly and inefficient. A modular approach, in which models of the relevant subsystems are interconnected and simulated together, is needed to reduce costs and development time. The use of co-simulation enables multi-domain simulations, which makes it possible to test a vessel design, including all its subsystems and equipment, using different modeling and simulation software suited for specific systems [2].

Full-system simulation nevertheless remains challenging. Typical maritime systems are complex and difficult to model and simulate as there are interactions between several different domains having different time scales (e.g. slow dynamics of large mechanical ships and fast response of electronic bridge components). Further, models span a wide range of complexities and accuracies. Thus, understanding how different subsystems interact and how they influence the overall system behavior is a demanding task [3].

These facts, and our geographically distributed simulation, form the fundament for our general requirements for a co-simulation framework:

- interoperability of different simulation models, simulation tools as well as hardware components
- reusability of simulation components
- support of soft real-time operation
- possibility to use already existing software (also commercial of the shelf software)
- integration of geographically distributed components
- integration of time independent submodules (e.g. for coordinate transformations)
- possibility to use simulation components as black boxes (to protect IP).

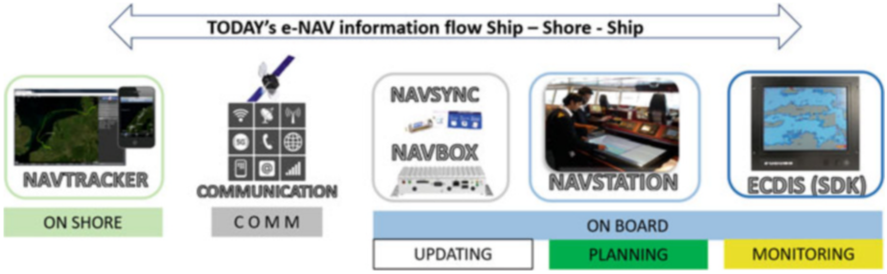

**Fig. 1** Coarse Architecture NAVTOR SBB. Source: NAVTOR

## 2.2 *Description of the System Under Test*

The System under Test (SuT) is the Shore Based Bridge (SBB) from NAVTOR[9] (Fig. 1).

The SBB is designed for:

- "One click" user interface for navigator to exchange routes between ECDIS (Electronic Chart Display and Information System) and Planner (NavStation)
- Highly automatic passage planning (detailed navigational plan made before leaving port)
- High quality routing due to full resolution of weather/wave/sea data (NavStation)
- Complete 2D wave spectra (not a few wave trains only) (NavStation)
- Benefit from probability-based weather forecasts (NavStation).

The SBB ECDIS integrates a variety of real-time information, making it an automated decision aid capable of continuously determining a vessel's position in relation to land, charted objects, navigation aids and unseen hazards. An ECDIS includes electronic navigational charts (ENC) and integrates position information from the Global Positioning System (GPS) and other navigational sensors, such as radar, fathometer and automatic identification systems (AIS[10]). It may also display additional navigation-related information, such as sailing directions. Further, the SBB includes a NavStation which includes the key features of passage planning. As a subscriber of the NAVTOR ENC Service, the NavTracker allows to track own vessel's position, costs of e.g. ENC and review update reports. With NavBox integrated in the bridge networks, the vessel will automatically receive the latest

[9] https://www.navtor.com

[10] Like almost every technology, AIS is subject to specific restriction and limitations, too. Because of the dual character of AIS data (disengageable, dependent on the human initiated processes) and the dependency on other onboard devices (for example the GPS receiver) there is still a margin for errors in both the static as well as the dynamic data. Insofar, a possibility cannot be ruled out, that AIS data is wrong or not meaningful during important maneuvers of a vessel. Source: ANNUAL OF NAVIGATION 19/2012/part 1. DOI: https://doi.org/10.2478/v10367-012-0001-0.

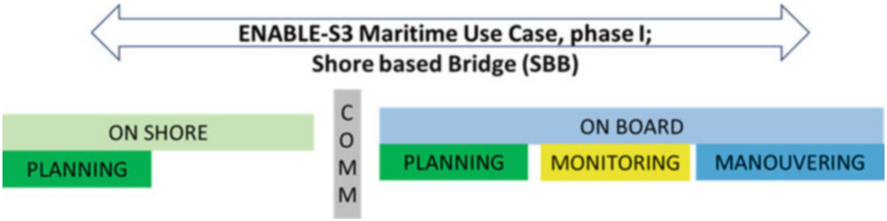

**Fig. 2** Deployment SBB Phase I

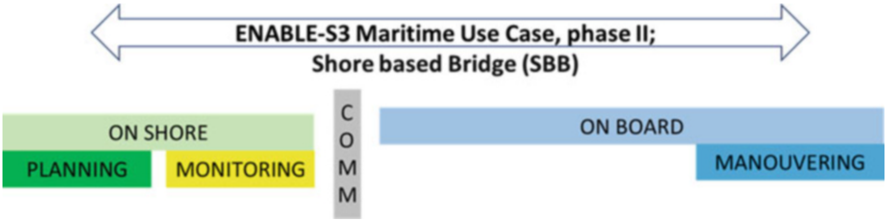

**Fig. 3** Deployment SBB Phase II

updates of charts and navigational publications. This saves time and always ensures full compliance to maritime regulations. Furthermore, the routes and passage plans for the own ship, together with sensor data available on board, are transmitted ashore via the NavBox. The integration of SBB in the co-simulation ashore and on the own vessel is shown in Fig. 5.

Today the ships voyage is planned on board with the NavStation and afterwards this route is transferred to the (ECDIS). This system is a navigation information system which, with adequate back-up arrangements, can be accepted as complying with the updated chart required by regulation V/19 & V/27 of the 1974 SOLAS (Safety of Life at Sea) convention, by displaying selected information from a system electronic navigational chart (SENC) with positional information from navigation sensors to assist the mariner in route planning and route monitoring, and by displaying additional navigation-related information if required. In ECDIS a safety check for the planned route is accomplished. The voyage of the ship, both track and planned route, can be observed by a web-based NavTracker via communication link (see Fig. 1).

Within the European project ENABLE$_*$S3 the SBB is developed further in the following stages:

- Duplicate planning function to Shore (Phase I, see Fig. 2)
- Move permanently the Planning and Monitoring function to shore (Phase II, see Fig. 3)
- Steering a ship autopilot from ashore (Phase III, see Fig. 4).

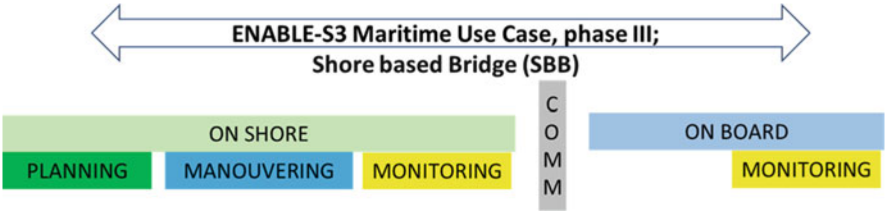

**Fig. 4** Deployment SBB Phase III

In phase I, the complete planning process for the required route is carried out by SBB ashore and finally transferred to the ship. Here the planning is checked again and transferred to the ECDIS. The steering of the ship and the monitoring of the route is carried out exclusively on the ship.

In Phase II, both planning and monitoring of the route will be carried out with support of SBB ashore. For this purpose, all the ship's sensor information must be available ashore. The ship's steering remains on the ship.

In Phase III, the planning, steering and monitoring of the vessel's route and the surrounding traffic will be carried out ashore. The crew of the ship will be substantially de-stressed and can potentially be reduced in numbers. For double navigational safety, the monitoring function additionally remains on board the ship.

To test the SBB the following six test cases have been defined:

- Test case 1: Passage planning and deployment (reduced crew)
- Test case 2: Passage monitoring (sensor & traffic data in near real time)
- Test case 3: Reaction to monitoring information (sensor data & alarms may lead to change of route)
- Test case 4: Acknowledgement on board (controlled transfer of autopilot control from shore or own ship)
- Test case 5: Remote vessel guidance (remote control of autopilot/virtual handles)
- Test case 6: "Fail-safe" and "fail-operational "(fail-secure to avoid a disaster).

In this paper we focus on test cases 1, 2 and 5.

## *2.3 Maritime Co-Simulation Framework*

In the maritime sector, in contrast to automotive engineering, only very small series are developed and built. The development costs amortize therefore only very slowly and there is permanently a high cost pressure. Due to this pressure, the development of these systems must be parallel and distributed, i.e. distributed among different teams and/or external suppliers, each working in their own domain with their own specific tools. Each participant, who is often far away, develops a partial solution into a system that has to be integrated with all other partial solutions. If the integration is carried out later in the development process, it will be less effective

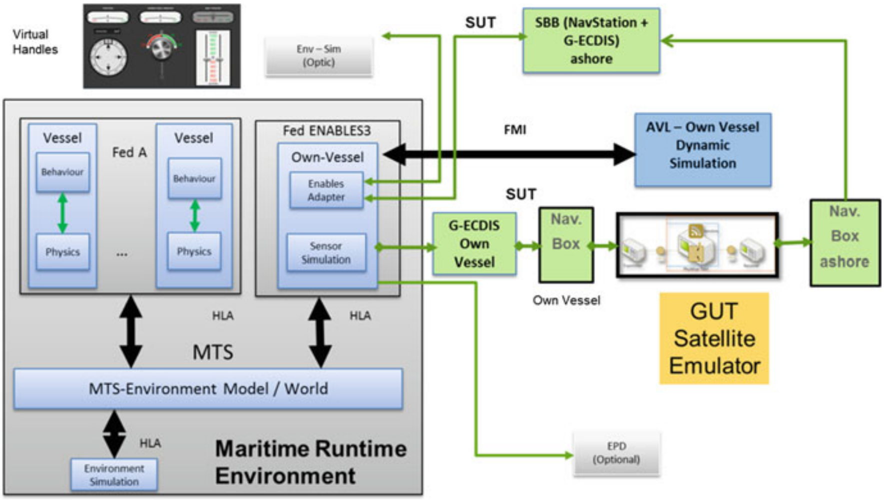

**Fig. 5** Architecture co-simulation with integrated SUT

[10]. Here we see a broad field of activity for the use of a distributed co-simulation to reduce costs on the one hand and to increase the quality of the products on the other. This means that not only finished products such as autopilots, GPS or AIS can be integrated into the co-simulation, but also their models. This does not enable only the early detection of the undesirable developments and errors, it also allows existing requirements to be validated and verified. In order to fully exploit the theoretical possibilities of a distributed maritime co-simulator (Fig. 5) and to test it in practice, it was distributed over two locations and a flexible, open, standards-based architecture was chosen to establish it.

Further for effective testing of the SBB, it must be integrated into the co-simulation so that it is possible at any time to have a:

1. Time synchronization between the Maritime Runtime Environment and the SBB
2. Comparison of the current data of the Maritime Runtime Environment with the target data/results of the SBB
3. Fast time simulation of longer ship approaches within special scenarios
4. Accurate evaluation/comparison of individual NMEA[11] sentences.

Sensor measurements from the own ship are provided either to the on board ECDIS system using standard maritime communication channels, or to the shore based ECDIS using similar standard communication channels. Those measurements

[11] National Marine Electronics Association. The NMEA 0183 Standard, a communication standard defined by the NMEA organization (www.nmea.org), defines a communication protocol that enables navigation instruments and devices to exchange data with each other. The NMEA 0183 Interface Standard defines electrical signal requirements, data transmission protocol and time, and specific sentence formats for a 4800-baud serial data bus.

provided directly to the SBB Planning Station on shore are the same measurements that are globally available for example through the AIS distribution system (Satellite based AIS, see also Marine Traffic), but with a NavBox you may see the complete coverage of surrounding vessels.

The ECDIS on board of the simulated own ship processes the sensor information in the same manner as it is done on real ships. In addition, it shall forward the messages to NAVTOR's backend system, using the already existing NavBox. Messages between two NavBoxes will be transmitted using satellite based IP services. The satellite based IP communication is simulated by GUT.

After passing the satellite simulation the available information will be presented at the SBB, allowing a nautical officer to decide if he need to intervene and take any action.

### 2.3.1 Maritime Runtime Environment

The Maritime Runtime Environment (Fig. 5) is the fundamental part of our distributed co-simulation. It is realized as a Model in the Loop (MiL)/Software in the Loop (SiL) simulation based on HLA (High Level Architecture), compatible to the eMIR-platform.[12] The data model does contain the current state of simulation technics.[13] It contains a world editor which is an Eclipse-based editor that allows set up a static scene according to a predefined scenario. This model contains the fundamental components of all used resources, actors and environmental factors. Further, the part Maritime Traffic Simulation (MTS) is a flexible simulator for implementing, executing and observing the behavior of multiple vessels in a realistic context. Routes for vessel traffic can be modelled or imported via standards defined by NMEA. Different infrastructures are available for the individual blocks of the Maritime Runtime Environment and also for the entire co-simulation in order to optimally fulfil the respective requirements for the simulation. Simulation of vessel movements can be coupled with sea charts and environmental simulations. The environment simulation provides a simulation of maritime environmental factors such as wind, wave, current, tide, etc. Further, the MTS contains an unlimited

[12] Open initiative of German maritime industry for improving safety and efficiency in maritime transportation systems.

[13] HLA standards: IEEE 1516-2000: High Level Architecture (Framework and Rules), IEEE 1516.1-2000: High Level Architecture (Federate Interface Specification), IEEE 1516.1-2000: Errata (16. Oct. 2003), IEEE 1516.2-2000: High Level Architecture (Object Model Template (OMT) Specification, IEEE 1516.3-2003: Recommended Practice for HLA Federation Development and Execution Process (FEDEP). Our model is based on the S-100 standard. S-100 is the document that explains how the IHO will use and extend the ISO 19100 series of geographic standards for hydrographic, maritime and related issues. S-100 extends the scope of the existing S-57 Hydrographic Transfer standard. Unlike S-57, S-100 is inherently more flexible and makes provision for such things as the use of imagery and gridded data types, enhanced metadata and multiple encoding formats. It also provides a more flexible and dynamic maintenance regime via a dedicated on-line registry.

set of vessels. Each of those vessels can be characterized with different features such as shape describing the hull of the vessel, current vessel velocity, position and orientation in global coordinate system. Other features include for example the vessel dynamics that is manipulated through the powertrain simulation on HiL.

The sensor simulation (Fig. 5) is a real-time generator for AIS/ARPA/GPS data for simulating data of real sensors on board. The simulation includes on the one hand sensors that give information about the vessel state, such as the actual rudder position, rotational speed, orientation, position, etc. On the other hand, there are sensors for environmental perception, e.g. the AIS which is commonly used in the maritime sector. Both types of sensors deliver their data to the SBB in the form of an NMEA data set.

The ECDIS of the simulated own vessel is fed with AIS information by the Maritime Runtime Environment. The radar and sensor data of the own ship and the surrounding vessel traffic is in the form of NMEA 0183 sentences, so no real sensors are required. Additionally, it forwards the NMEA messages to NAVTOR's backend system at shore for monitoring this data on the G-ECDIS (Generic ECDIS). The G-ECDIS processes the sensor information onboard as real ships. Messages between two NavBoxes (one ashore and one on board) will be transmitted using satellite based IP services.

The "Lobo Marinho" (Ro-Ro Cargo Ship) is specified as own vessel. This ship with its maneuvering behavior is simulated by the HiL simulation. The surrounding ship traffic, the sensor data of the own ship and the entire environment of the respective scenario are simulated by the Maritime Runtime Environment. The influences on the communication like run times and disturbances between the SBB (shore station) and the own ship, are taken into account by a satellite emulator (GUT).

### 2.3.2 HiL Simulator at AVL SFR

The following figure (Fig. 6) illustrates the HiL Simulation of the own ship simulation.

The HiL simulator consists of the ship and engine model in loop with the Electronic Control Unit (ECU) for engine control. It is real-time capable and thus a CAN interface (Controller Area Network) was implemented for the signal transfer between HiL and Model.CONNECT. Model.CONNECT provides then the communication with other components, running by OFFIS.

The concept of "master" and "slave" in the FMI is very different from that of HLA's. When a simulation package imports an FMU, it becomes its "master" and the loaded component becomes its "slave". FMI for co-simulation provides a mean to utilize models using an API, in a form where slave acts as a black box to master. It can react to inputs and gives outputs at discrete time steps. While using an FMU conforming to FMI for co-simulation, one does not need to know which integration method is actually applied to solve the model. Parameters to this black box can be set in initialization phase and cannot be changed afterwards, while the inputs can

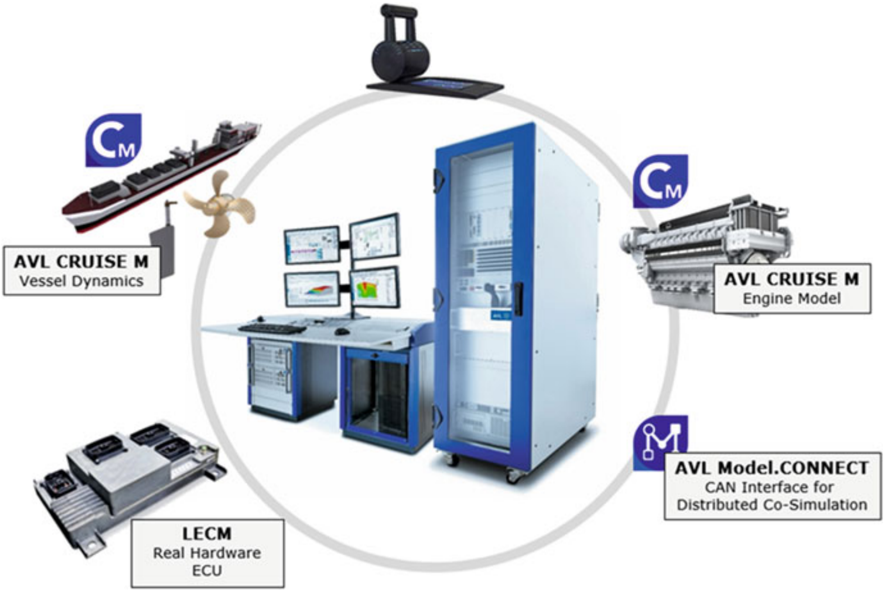

**Fig. 6** HiL simulator AVL

be changed between discrete time steps. In our case the HiL is set up so that the models on the HiL start the simulation as soon as the HiL with the zero set points is switched on.

### 2.3.3 Interface Between the Maritime Runtime Environment and HiL

A dedicated VPN network has been established between AVL SFR and OFFIS for the distributed co-simulation platform. The VPN is encrypted to ensure data security at both sites.

To setup the simulation of the test scenarios, Model.CONNECT is used as co-simulation environment. Model.CONNECT controls the simulation and provides mechanisms for an intelligent coupling and advanced simulation mechanism by using of the ICOS[14] blocks. The following figure (Fig. 7) shows the first implementation (Var. 1) of the distributed co-simulation architecture for this part in detail.

Via a VPN connection data will be exchanged by a proprietary UDP connection. The Maritime Runtime Environment only needs to create an interface to Model.CONNECT ①. This interface is realized as FMI which is supported by Model.CONNECT by default. The connection to the ICOS-Remote-Server

[14] Independent Co-Simulation.

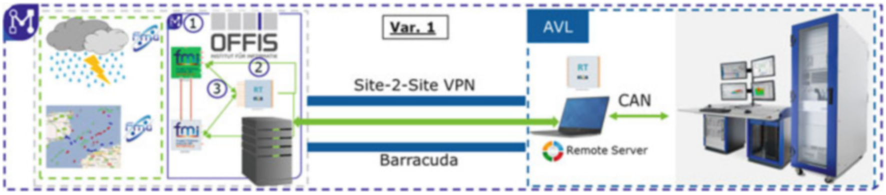

**Fig. 7** First implementation (Var. 1) distributed co-simulation architecture

(AVL-side) must be realized via RT[15]-ICOS-wrapper ②. In Model.CONNECT FMUs/other Models can be connected to RT-ICOS-wrapper ③. On AVL side the access to the HiL is realized via ICOS Remote Server thus there is the possibility of connection ECU, injectors, flaps etc. The Cruise M models are running on the HiL in real-time. However, the real-time capability of the entire architecture depends on the network infrastructure.

During the implementation of the Var. 1 architecture the question arose why a hard real-time capable HiL should be integrated over a secure VPN (Barracuda Server) with a soft real-time capable maritime runtime environment. With regard to the implementation of the planned test scenarios, no advantages of this implementation could be identified. This was especially true as the internal powertrain behavior should not be tested within the scenarios.

In general, it is not possible to be faster than the network infrastructure. One of the tasks of our research work is to investigate which time delays hinder or render impossible the execution of the panned test scenarios. Usually the lags were about 20 ms. In addition, the processes running in milliseconds within the powertrain simulation of the HiL are not required for the scenario execution, but the surge, sway and yaw velocity.[16] These speeds are essentially determined by set points rudder position, engine speed and propeller position of the Maritime Runtime Environment. Both the update rates for the settings of the Maritime Runtime Environment (setpoint data) and the feedback from the own vessel simulation are in the range of seconds. Thus, the HiL itself does not have to be integrated, but a software component which represents the real dynamics and behavior of the "Lobo Marinho". Consequently, a second communication architecture (Var. 2) was developed, which is illustrated below (Fig. 8).

[15] Real time.

[16] Three translations of a ship's center of gravity in the direction of the x-, y-, and z-axes:

- surge in the longitudinal x-direction, positive forward
- sway in the lateral y-direction, positive to the port side
- heave in the vertical z-direction, positive upward

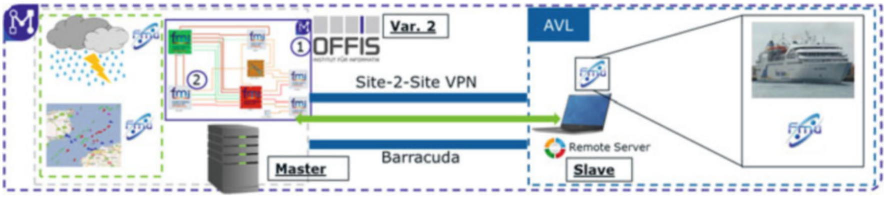

**Fig. 8** Second implementation (Var. 2) distributed co-simulation comArchitecture

On AVL-side the Cruise M models in form of a FMU are running on a simulation computer. The access to simulation computer is realized via ICOS-Remote-Server.

In the Maritime Runtime Environment Model.CONNECT is running ①. The Connection to the ICOS-Remote-Server is implemented via FMU-wrapper ②.

After loading the FMU the Maritime Runtime Environment as master becomes the complete owner of the FMU and there is no question of “entity transfer”. Similarly a “shared resource” does not exist in the FMI. An FMU can only read and write some values, it does not have an entity similar to a process. Naturally, there cannot be a concept of ownership management, when slave itself is owned by master.

Figure 9 illustrates the main components of the data exchange between the Maritime Runtime Environment and the AVL simulation computer to simulate the behavior of the “Lobo Marinho”. The following data are provided by the Maritime Runtime Environment as set point data

- weather data
- rudder position
- engine speed
- propeller position.

As feedback to these control signals, the AVL simulation computer sends

- Surge (forward and backward movement of the center of gravity of a ship)
- Sway (movement along the transverse axis)
- Yaw (movement around the vertical axis)

speed (actual data).

The current speed of the own ship or its current position and orientation (pose) within the world are simulated in the Maritime Runtime Environment.

### 2.3.4 Satellite Emulation

The simulation of a proportionate real maritime world includes the simulation of the satellite connection between the ship and the shore station. This simulation is performed by a satellite emulator, which simulates the occurring delays in the communication as well as the possible disturbances within this connection. This emulator is located in Gdansk.

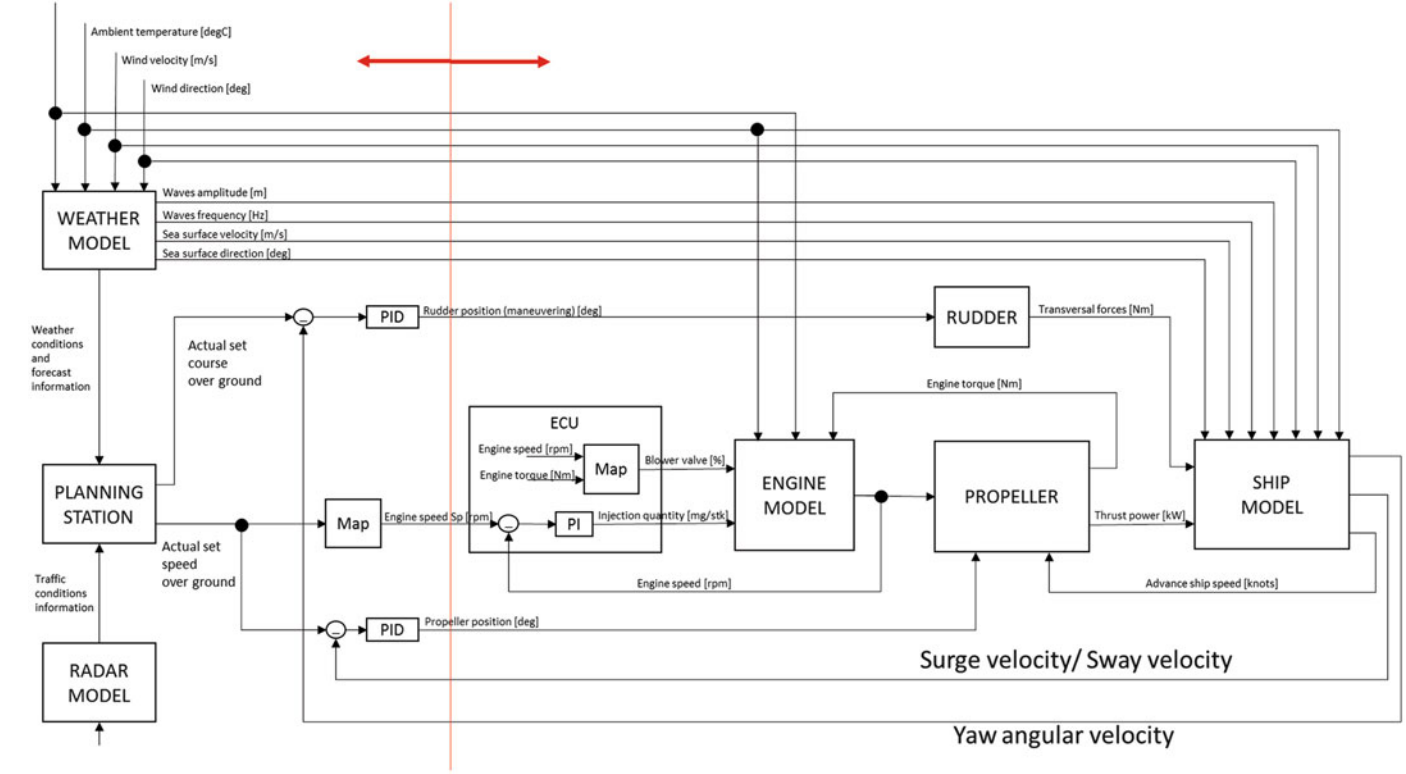

**Fig. 9** Signal flow between Maritime Runtime Environment and AVL simulation computer

The simulation will be done in the PhyWise Tool. Two different configurations will be available: software simulation and hardware simulation with devices prototypes. Additionally each of the configurations will allow simulation of communication with additional jamming signal source.

In Fig. 10, the PhyWise tool is placed between two components of a given ACPS system. In the case of maritime co-simulation, this tool is located in the communication channel between the two NavBoxes (ship and shore) (Fig. 5). The two NavBoxes display their data via the PhyWiSe-Tool, which has the main task of checking the security at the physical level. The PhyWise tool contains a special communication simulation component that is responsible for reflecting the characteristics of the communication channel as a model, enabling both MiL and HiL testing. The jammer simulation is responsible for generating deliberate interference for the needs of evaluation of secure communication and SBB resistance to such a threat. The interference detection receiver is a component that can detect, detect and locate both intentional and unintentional interference.

# 3 Experiences and Results

## *3.1 System Under Test*

We have tested the SBB with over 60 scenarios using our co-simulation setup. Based on the results, we can say that the route planning of the NavStation is carried out fully navigationally safe (test case 1). The tools installed on the NavStation, such as Admiralty Total Tide[17] (ATT) or NavArea,[18] have proved to be extremely helpful and efficient. As an example, a time saving of more than 66% compared to conventional planning of a route from Barcelona to Las Palmas (Gran Canary) can be mentioned. In our opinion a network based route optimization tool is missing in the weather forecast. A corresponding concept for this has already been developed.

According to Winner et al. [11] and Winner and Wachenfeld [12], following basic statistics, between 100 million and 5 billion km of test driving are required in order to ensure that software for autonomous driving or in our case for voyage planning and remote vessel guidance is at least as save as a human driver. The navigational/mathematical analysis and reconstruction of the scenarios enabled to determine both the causes of the accidents (collisions, groundings, etc.) and their avoidance actions. Test case 5 (remote vessel guidance) shows that these accidents can be avoided by means of this function. The NavStation, as part of SBB, was continuously further developed and optimized based on results of scenario-based

[17] ADMIRALTY TotalTide (ATT) provides bridge crews with fast, accurate tidal height and tidal stream predictions for more than 7000 ports and 3000 tidal streams worldwide.

[18] NAVAREAs are the geographic areas in which various governments are responsible for navigation and weather warnings.

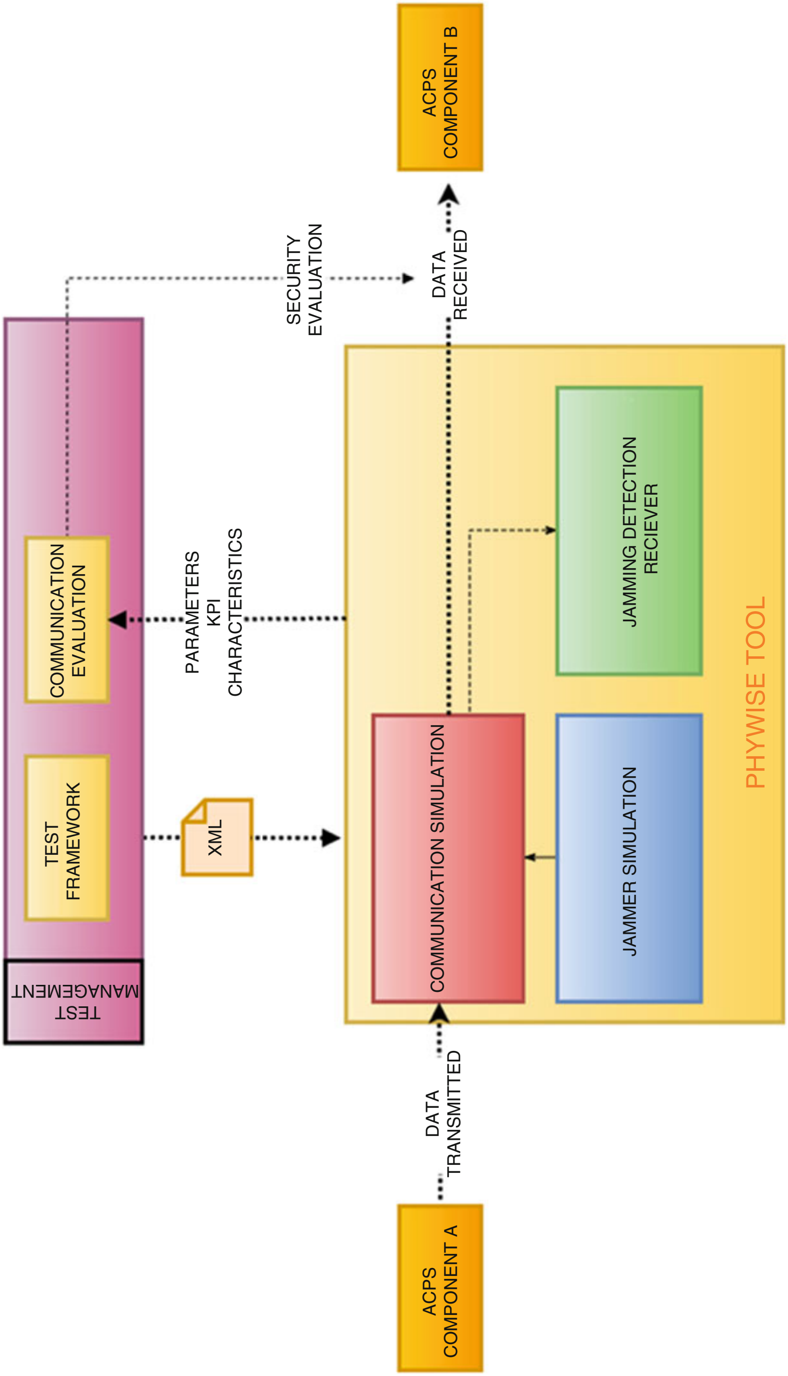

**Fig. 10** Basic architecture PhyWise Tool. Source: GUT

testing. However, the situation analysis on the ECDIS, which is based on the monitoring data of the own ship, must be carried out by the respective operator. The SBB does not contain a tool for situation analysis. In addition, there is no extensive support with regard to the COLREGs[19] (Convention on the International Regulations for Preventing Collisions at Sea) in implementing the remote vessel guidance.

The lack of a genuine handshake procedure (request/acknowledge) in the data traffic between the shore station and the ship has proved to be a serious deficit at SBB. This makes it impossible, for example, to transfer autopilot control from shore back to ship or vice versa (test case 5). This leads to confusion as to where the autopilot is controlled from. The necessary coordination was made possible during the test phase by conventional voice communication.

A further deficit lies in the pure safety aspects in the use of SBB. These concerns both access authorizations and the security of data exchange between ship and shore.

The pure evaluation of the sensor data of the own ship and the surrounding ship traffic ashore by the G-ECDIS proved to be error-free. Correct NMEA sentences were evaluated completely and correctly, faulty ones were detected and rejected.

## *3.2 Co-Simulation*

The co-simulation enables the testing of all scenarios. In particular, different systems such as assistance systems or ship types with different propulsion systems can be tested at numerous environmental conditions and conclusions can be drawn about the planning and monitoring process (test case 2) of the SuT. At the same time, the effects of delays and disturbances in communication on ship route monitoring and remote vessel guidance can be recorded and analyzed using satellite simulation. For delays >10 s neither monitoring nor the remote vessel guidance function is given. Further, the integration of vehicle dynamic simulation makes it possible to explicitly test ship engines, rudders and propulsion systems under all possible sailing and weather conditions on the planned voyage. The purpose of performance evaluation is to compare fuel efficiency or propeller performance at a given point in time, i.e. to compare the ship state at several conditions. Since a ship is exposed to external factors such as wind, waves, shallow water, changes in seawater temperature, etc., it is unlikely that the ship will ever be in exactly the same situation more than once. An outstanding advantage of the co-simulation is

[19] The International Rules shipping traffic rules were formalized in the Convention on the International Regulations for Preventing Collisions at Sea, 1972, and became effective on July 15, 1977. The Rules (commonly called 72 COLREGS) are part of the Convention, and vessels flying the flags of states ratifying the treaty are bound to the Rules.

the exact reproducibility, the starting and stopping of the scenarios at any time and the available analysis possibilities. This is rarely, if ever, not the case with real-world tests.

The different results occur between the mathematical-navigational calculation of the scenarios, the results within the maritime runtime environment and those in the SBB. In many cases, deviations occur in distances, bearings, courses and speeds above ground. To the best of our knowledge, these deviations can be attributed to the use of various projection models. While normal nautical charts and ECDIS use Mercator projection, the Maritime Runtime Environment uses the mathematical models of Vincenty.[20] The deviations that occur may not be relevant on the high seas with correspondingly low traffic volumes, but they are relevant in coastal waters or in narrow fairways. Remote vessel guidance is not permitted in these sea areas for this reason alone. It is an important question whether in the ECDIS time still Mercator projection is essential for marine navigation. Further research is required to move from a Mercator projection to calculations based on a spherical earth model. Vincenty's solution for the distance between points on an ellipsoidal earth model is accurate to within 0.5 mm distance and 0.000015″ bearing, on the ellipsoid being used.

Further safety-critical deviations in the results occur in the Closest Point of Approach (CPA) and Time to the Closest Point of Approach (TCPA) calculation. Comparing encounter parameters CPA and TCPA obtained from ARPA and decision-support system it can be stated that:

1. CPA presented by ARPA is always larger than the manual CPA calculation
2. TCPA values approximate each other.

Differences in CPA values are due to the fact that the length of the vessel is not included in the calculation. For example, information on the size of the vessel and the antenna position can take from the AIS system. In ARPA as well as in the formulas presented by Lenart [13], the ships are treated as points; therefore the falsification of the results, which can lead to a wrong assessment of the situation by the navigator, this is particularly true for the encounters of large ships. The largest CPA difference within the scenarios carried out was 0.21 nm at close range, which represents an error of more than 50% at a CPA value of 0.4 nm.

CPA results by the existing methods which only consider Speed over Ground (SOG) and Course over Ground (COG) are imperfect. A CPA calculation method taking account of SOG, COG, Change of Speed (COS) and Rate of Turn (ROT) is required.

[20] Vincenty's formulae are two related iterative methods used in geodesy to calculate the distance between two points on the surface of a spheroid, developed by Thaddeus Vincenty in 1975. They are based on the assumption that the figure of the Earth is an oblate spheroid, and hence are more accurate than methods such as great-circle distance which assume a spherical Earth (http://en.wikipedia.org/wiki/Vincenty's_formulae).

AIS can provide information on all four essential factors, as well as the position and other important information. AIS data is therefore suitable to be used to predict the positions and calculate CPA.

The results of CPA calculation in the Maritime Runtime Environment confirm this tendency. On smaller vessels the smallest passing distance on the level of 0.5 nm is often considered as safe. Within this short distance, the deviations described play a significant role. TCPA values are close to each other, as the vessel's size does not affect the moment of contact, only its value.

## 4 Conclusion

To test functionality in all possible situations, the automotive domain assumes that a high automation assistance system must drive 240 million kilometers, when the distance between two accidents with personal damage is 12 million kilometers. Such a number on test runs is hard to reach, even with fast-time simulation. One approach to reduce the number of necessary runs is to generalize situations and explicitly focus on critical, rare events. We have followed this approach in principle, but without ignoring "normal behavior" in maritime transport. The SBB was tested with the use of 60 scenarios that included both the simulation of:

- Close-range situations
- Collisions
- Groundings
- Hazards due to technical failures
- Normal traffic situations.

SBB's planning and monitoring function (evaluation of sensor data) could be carried out realistically and effectively by means of the co-simulation (test cases 1 and 2). Within the Remote Vessel Guidance function (test case 5), further research effort is required to develop and test target-oriented additional functions. The combination of scenario based testing and requirements based testing provides a more robust acceptance potential that gives the user and manufacturer confidence that the system meets their requirements. There is a risk that a legacy system which has not been subject of a validation program, will not meet regulatory expectations, e.g. IMO compliance. It is therefore necessary to check the conformity of existing systems. Basically, the owner of a legacy system should ensure that an appropriate validation package is available. Here we see another approach and advantage for scenario-based testing. In the further development of the NavStation, the conformity with the currently existing regulations was comprehensibly proven with the support of scenarios.

Co-simulation represents an enormous increase in effectiveness for the maritime domain. The first prototype is usually only available after 50–60% of the development time. With a development time of 5 years, the first 3 years would be without any experience of how the interaction of the new components works at all. This is exactly where co-simulation techniques come in: They bring together individual simulation models. Once the prototype of a component (e.g. the autopilot, the integrated bridge or the ECU) is finally available, the individual virtual model is removed from the overall simulation and replaced by the prototype on the test bench. This data then comes from the real component, but is fed into the same system and correlated with the other values still simulated. The model thus becomes more and more real step by step.

## References

1. Trcka, M., Hensen, J.L.M., Wetter, M.: Co-simulation of innovative integrated HVAC systems in buildings. J. Build. Perform. Simul. **2**(3), 209–230 (2009)
2. Skjong, S., Rindarøy, M., et al.: Virtual prototyping of maritime systems and operations: applications of distributed co-simulations. J. Mar. Sci. Technol. 1–19 (2017)
3. Sadjina, S., Kyllingstad, L., et al.: Distributed co-simulation of maritime systems and operations. J. Offshore Mech. Arct. Eng. **141**, 011302 (2018)
4. Chu, Y., Hatledal, L.I., Sanfilippo, F., Schaathun, H.G., et al.: Virtual prototyping system for maritime crane design and operation based on functional mock-up interface. In: OCEANS 2015, pp. 1–4. IEEE, Genova (2015)
5. Boukerche, A., Lu, K.: A novel approach to real-time RTI based distributed simulation system. In: Proceedings of the 38th Annual Simulation Symposium, pp. 267–274. IEEE, San Diego, CA, USA, 4–6 April 2005
6. Monga, M., Karkee, M., et al.: Real-time simulation of dynamic vehicle models using a high-performance reconfigurable platform. In: Proceedings of the 2012 International Conference on Computational Science (2012)
7. You, T., Zhu, Y., et al.: Applied research of delaminated real-time network framework based on RTX in simulation. In: Proceedings of the 2nd International Conference on Information and Computing Science, pp. 389–392. IEEE, Manchester, UK (2009)
8. Awais, M.U., Palensky, P, et al.: Distributed hybrid simulation using the hla and the functional mock-up interface. In: 39th Annual Conference on Industrial Electronics Society. IEEE, Vienna, Austria, 10–13 November 2013
9. Trcka, M.: Co-Simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings. Eindhoven University of Technology, Eindhoven (2008)
10. Tomiyama, T., D'Amelio, V., et al.: Complexity of multi-disciplinary design. CIRP Ann. Manuf. Technol. **56**, 185–188 (2007)
11. Winner, H., Wolf, G., et al.: Freigabefalle des autonomen Fahrens/the approval trap of autonomous driving. VDI-Berichte. **2106**, 17–29 (2010)
12. Winner, H., Wachenfeld, W.: Die Freigabe des autonomen Fahrens. In: Maurer, M., Gerdes, J., Lenz, B., Winner, H. (eds.) Autonomes Fahren, pp. 439–464. Heidelberg, Berlin (2015)
13. Lenart, A.S.: Manoeuvring to required approach parameters—CPA distance and time. Annu. Navigat. **1**, 99 (1999)