INTEROPERABILITY IN CO-SIMULATONS OF MARITIME SYSTEMS

Christoph Dibbern
Axel Hahn
Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstr. 114-118
D-26129, Oldenburg, Germany
E-Mail: {dibbern, hahn}@wi-ol.de

Sören Schweigert
OFFIS - Institut for Information Technology
Escherweg 2
D-26131, Oldenburg, Germany
E-Mail: soeren.schweigert@offis.de

KEYWORDS

Distributed Simulation; Maritime Traffic Simulation; Interoperability; Model Transformation; Semantic Model; Conceptual Architecture; Layers for Simulation-based Analysis; HLA; RTI;

ABSTRACT

Powerful, flexible and cost-effective analyses of maritime systems can be conducted in an appropriate way with distributed simulations. Co-simulation components require interoperability like specified by the IEC TC/65/290/DC to insure information exchange and synchronization. Therefore this paper describes a way to insure the interoperability of co-simulations of maritime systems. User- and technical requirements for these co-simulations are derived to create a deeper understanding for especially the necessity of a common semantic model. Furthermore, six integration layers are presented for co-simulation-based analysis. They are derived from IEC TC/65/290/DC and the conceptual architecture approach for co-simulation systems. The paper introduces a semantic model approach to reflect the requirements formal description of the simulation model, observability, controllability and interoperability. The model is used to configure the inter-process communication of the used co-simulation architecture. This is done by an automatic model transformation from the semantic model to the platform specific implementation.

INTRODUCTION

Seafaring has always been a joint undertaking between humans and their technology. The reliability of the technical equipment and its correct usage ensure safe travelling. This is still true with the implementation of eNavigation technology.

The eNavigation implementation process is accompanied by IMO's NAV and COMSAR sub-committees, as well as the International Hydrographic Organization (IHO) and the International Association of Lighthouse Authorities (IALA). These institutions did a comprehensive gap analyses as part of their development of a joint implementation plan for eNavigation. In a ten year survey (Gale 2007) investigated the causes of collisions and groundings, in which human error was the primary cause with 60%. Therefore the gap analysis of the IMO addresses numerous aspects of human machine interaction (IMO 2012), e.g. absence of structured communication to report incorrect operation of both shipboard and/or shore-based systems together with a lack of intuitive human-machine interface for communication and navigation means. Furthermore, the analysis revealed that existing performance standards are not applied or are missing such as guidelines for usability evaluation (IMO 2012). This requires that equipment providers have to do a comprehensive usability and risk assessment of their products. IMO MSC Circular 878 states: “A single person’s error must not lead to an accident. The situation must be such that errors can be corrected or their effect minimized. Corrections can be carried out by equipment, individuals or others.”

Cognitive Simulation Based Test Bed

To fulfill the announced IMO requirements the authors propose a cognitive simulation based test bed for human machine interaction, to provide a test bed for experiments during early design phases of an eNavigation System like an Integrated Navigation System (INS).

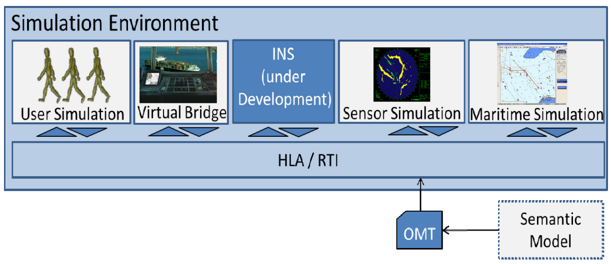

Figure 1: Simulation-based analysis of human machine interaction

Figure 1 shows the proposed test bed containing the system under development that is the tested in a co-simulation environment: A maritime traffic simulation provides the context for the experiments. A sensor simulation uses the traffic model to generate AIS and radar signals. The data is streamed to the *INS* for testing purposes. We use a simulated user (a virtual crew member implemented by a cognitive simulation) (Wortelen et al. 2006) which interacts with the not yet ready *INS* via another system: the virtual bridge simulator. It provides the 3D Environment of the ship

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

bridge in which the new *INS* is used and implements the human machine interaction (Puch et al. 2012).
Each simulation component is implemented as a separate simulation that must be synchronized in terms of time and communicate the dynamic information like vessel position, headings etc. with all the others.
All simulators must have the same understanding of their shared environment in order to carry out a joint simulation. Therefore a vessel within the *Maritime Traffic Simulation* is the same entity of a vessel as in the *Sensor Simulation*, *Integrated Navigation System* and the *User Simulation*. The common *Semantic Model* shown in Figure 1 is used to describe all items and the shared environment of all simulators.
In our case, this is technically archived by using a High Level Architecture (HLA) compliant implementation (Noulard et al. 2009) like described in Läsche et al. (2013). This implementation requires a formal description of the communicated data, in form of an Object Model Template (OMT).

INTEROPERABILITY REQUIREMENTS FOR THE CO-SIMULATION

To define the co-simulation requirements first bottom-up and top down practical requirements are identified. Secondly the authors take the IEC requirements on interoperability into account and finally the authors derive requirements from layered integration architecture for co-simulation systems.

Practical Requirements

Bottom-up it is necessary to find the technical requirements on the system to derive a suitable *Semantic Model* and architecture so that the user requirements can be fulfilled. In addition, the analysis of maritime systems generates user requirements.
These requirements of co-simulations of maritime systems can be emphasized with a top-down, problem-oriented analysis. Its goal is to derive the user requirements which have to be satisfied so that the co-simulation is able to support the usage scenarios. This focusses on the question: Which user requirements have to be satisfied to support the needed scenarios for analyzing purposes? The requirements of the usage scenario "analysis for maritime systems" can be split into two groups (s. Figure 2). The related user requirements are visualized as inner rectangles and are derived during the requirements analysis. The first group comprises usage scenarios like the Simulation Description and the automatic *Formal Analysis* of offshore operations as described in Läsche et al. (2014). The *Simulation Description* is e.g. necessary to describe the maritime environment like buoys and havens on waters like the Weser. Moreover static vessel charackteristics like the power of its engine and its maximum draught can be described. An example for the automatic *Formal Analysis* is the analysis of hazard events like an injured person at sea as described in Läsche et al. (2014). Their approach allows assessing the reasons for hazards. Relevant reasons can be e.g. environment factors (wind, waves etc.), broken equipment and human failures. Therefore a complete and correct *Formal Description* (e.g. of the agents behavior and the environment) is required. Because of that the *Semantic Model* has to provide all data types and parameters for the *Formal Description*. The second group includes the requirements regarding the *Controllability, Observability* and *Interoperability*. *Controllability* means the possibility to influence the behavior of the simulation components as well as the whole co-simulation with commands like start and stop. Moreover, the controllability also covers the injection of manual and automatic failures like described in Läsche et al. (2013). Furthermore, the co-simulation has to be *observable* because the user has to be able to monitor the internal states of the simulation components. An example is the logging of GPS positions of the simulated vessels for collision detection.

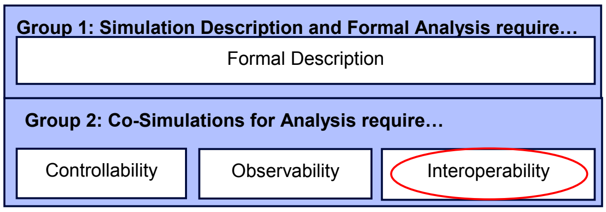

Figure 2: Problem-oriented View for the User Requirements

The user requirement *Interoperability* and its technical requirements are explained in the following chapters.

Interoperability Requirements

According to IEC (2002) a system is interoperable if: "The application data, their semantic and application related functionality of each device is so defined that, should any device be replaced with a similar one of different manufacture, all distributed applications involving the replaced device will continue to operate as before the replacement, but with possible different dynamic responses." The insurance of the *Interoperability* is necessary for the simulation consistency during runtime. Table 1 shows six *System features* which are necessary for an interoperable system according to IEC (2002). This means, *Interoperability* stands for the interaction of two or more systems which fulfill all of the listed system features.
Summarized, technical requirements for the *Interoperability* are according to the IEC a common *Communication Protocol*, *Communication Interface Data Access* and *Data types.* These are necessary to provide the network infrastructure of distributed systems. Furthermore, the definition of *Parameter Semantics* and dependency and consistency rules is required to insure the *Interoperability* of the system (s. *Application Functionality*). The IEC *System features* help to derive the technical requirements for the co-simulation integration.

Table 1: System Features for *Interoperability* (IEC 2002)

System feature	Description
Application Functionality	"This feature is defined by specifying the dependencies and consistency rules within the Functional Elements. This is done in the data description part or in a separate behavior section."
Parameter Semantics	"This feature is defined by the characteristic features (parameter attributes) of the application data this can be data name, data descriptions, the data range, Substitute value of the data, default value, persistence of the data after power loss and deployment."
Data Types	"The data type of the object attributes or block data input, data output or parameter defines this feature."
Data Access	"This feature is defined by the object operation definition or the access parameter attributes of the block data input, data output and parameters."
Communication Interface	"This feature is defined by the communication service definition of application layer including the services and the service parameters. Additional mapping mechanisms can be necessary. The dynamic performance of the communication system is part of this feature."
Communication Protocol	"This feature is defined by all protocols of layer 1 to 7 of the OSI (Zimmermann 1980) reference model, i.e. from the physical medium access to the application layer protocol."

Table 2: Layers for Co-Simulations of Maritime Systems based on (Schütte 2013)

Category of TRs	Layer	Artifact
Control of the Simulation	6) CONTROL	Controlling Applications
Modelling / Scenario Composition	5) COMPOSITION	Scripts to configure the system to simulate e.g. a ship's bridge
Decomposition	4) SCENARIO	Distributed Models in Domain Specific Simulators
Definition of the Semantics	3) SEMANTIC	Model for the **Interoperability**, Observability and Controllability
Configuration of the RTI	2) SYNTACTIC	RTI-Configuration File
Connection by Network / Infrastructure	1) TECHNICAL	RTI (IPC, Time-Synchronization), Observer

Co-Simulation Integration

The conceptual architecture for simulation-based analysis from Schütte (2013) includes six layers (s. Table 2). The following description focusses on the first three layers because they are necessary to insure the *Interoperability* (Schütte 2013). The authors extended the conceptual architecture by technical requirement (TR) categories and on the right site of the table, the artifacts per layer for co-simulations of maritime systems. The **first** layer provides the network connection and infrastructure. Therefore the system features *Communication Protocol*, *Communication Interface* and *Data Access* from Table 1 can be assigned to this layer. The associated artifact is a HLA-based Runtime Infrastructure (RTI) to provide an inter-process communication (IPC) as well as a time-synchronization. These are sensible to synchronize the distributed simulation components regarding their simulation states so that their communication is insured (Läsche et al. 2013). Another element on the first layer is the *Observer*. It is sensible for the infrastructure because it helps to analyze maritime systems through the recognition of predefined events during co-simulations (Läsche et al. 2013). Therefore, *Observer* are only able to listen on the HLA communication interface in this case (Läsche et al. 2013). The artifact on the **second** layer is necessary for the configuration of the RTI. This is a configuration file which contains the exchangeable data types, their units and other necessary information. This means, the system feature *Data Types* can be assigned to this layer. The **third** layer has to contain all data types and parameters with their semantic to insure the correctness and completeness of the co-simulation states. This helps to insure the communication respectively the data exchange between the co-simulation components. Furthermore, the layer includes all data types and parameters which are necessary to describe the possible behavior of the simulation components (commands like start, stop, pause etc.) so that it fulfills the system feature *Application functionality*. Therefore it includes the *Parameter Semantics* and the *Application Functionality* from Table 1. It follows that the satisfaction of the requirements from the first three layers insures the *Interoperability* of a co-simulation as defined by the IEC.

The artifact is the *Semantic Model* for both system features with all necessary data types and the *Parameter Semantics*. Summarized, it can be defined as follows:

> The *Semantic Model* insures the *Interoperability*, *Observability* and *Controllability* of the simulators. Furthermore, it contains entities and parameters to describe static information like environments.

The design rationales for the creation of the *Semantic Model* are described in the following chapter.

The **fourth** layer includes the domain specific scenario simulations. A scenario is runnable by an individual simulator to simulate e.g. that a human is able to move on a ships bridge. The **fifth** layer is responsible for the scenario configuration. As described, each simulator has its own scenario. The scenario in a co-simulation is a composition of each scenario of each simulator to simulate e.g. ship's bridge in combination with the maritime traffic on waters like the Weser.

The **sixth** layer includes applications to control the whole co-simulation and its components. They use the defined data types and parameters of the third layer which are necessary to fulfill the system feature *Application functionality*. The *Controllability* is insured if the requirements of all six layers are fulfilled.

Conclusion for the Co-Simulation Integration

Overall the following technical requirements on the simulation components are primary relevant for the *Interoperability* according to Schütte *(2013):* The simulation components have to be connectable to a *RTI (*s. layer 1 and 2 of Table 2*)*. They must have a common understanding of the time so that the co-simulation states can be resumed to a defined point in time (s. layer 1). The simulation components use a common *Semantic Model* for their communication (s. layer 3). Their *Control Unit* uses unique, interoperable control commands like *start*, *stop (*s. layer 3*)*. The Control Unit is able to synchronize the step size of each simulator as defined. This is necessary to insure the consistency of the co-simulation states. In addition, data type and range matching has to be performed (Schütte 2013). The data types have to be matched so that it is insured that the internal data types of the simulation components are compatible with the data types which are specified in the common *Model*. Range matching is the check of the data type ranges. This is necessary to insure that the possible value ranges have the required level of granularity.

DESIGN RATIONALES FOR THE MODEL

This chapter focusses on the design rationales for the model from the third layer (s. Table 2). The model design is based on API design rationales as described in Bloch (2006) and Tulach (2008) to increase its reusability and quality. The following design rationales are explained below: the use of standards, object oriented design and transitive dependencies**.**

The use of established **standards** (ISO, IEC etc.) helps to cover the correctness and completeness of the model. This allows to consider e.g. the correct description of the environment with the ISO19125 (ISO 2004) which is a specification for geographic information meta data. Moreover the appropriate use of standards increases the probability that the model can be reused for projects with the focus on maritime risk and efficiency analysis. Furthermore the model is based on standards for the system engineering like the Department of Defense Architectural Framework (DoDAF 2011). This specifies e.g. the vocabulary for the description of the DoDAF meta models. All used meta- model standards are explained more detailed in the next chapter.

The **object oriented design** with its hierarchical structure allows an effective reuse of model elements. This design comprises e.g. the use of polymorphism and encapsulation. The practical use for polymorphism is e.g. to be able to describe that there are several types to specify a point in space that could be either a two dimensional space like it is used in a traffic simulation on the water plane, or a three dimensional space required in an underwater simulation.

Encapsulation means in this case that the related data types and parameters are logical organized in model parts (s. Figure 3). Therefore, the encapsulation helps to consider the separation of concerns principle (Gurp and Bosch 2003). For example, the *Traffic System* has no information about elements defined in the *Sense* model. As consequence, changes in the *Sense* model have no impact on the *TrafficSystem* and vice versa. In addition this covers the need of a slim model. The **transitive dependencies** of the model parts ensure that all types used in a more general package are available, even if the dependency is not modeled explicit.

THE SEMANTIC MODEL

This chapter focusses on the structure and used standards of the model from the third layer (s. Table 2). The basic model structure is visualized in Figure 3. It includes two layers. The top layer contains model parts with general data types and concepts. The layer below is for the two model parts with domain specific data types. The meta-model standards *ISO19125*, the *SensorML* and *Safety ISO26262* are used for this layer. The standards are explained together with the layers below.

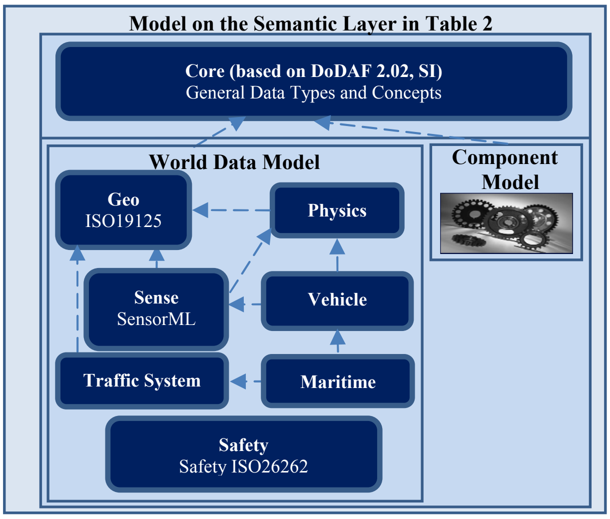

Figure 3: Structure of the Model with Transitive Dependencies

The *Core* is an elementary, object oriented model part with general data types and concepts. *Concepts* are empty shells. They cannot be instantiated and have to be specified in sub models. In addition, they are useful to categorize data types and parameters. The concept Vehicle shows e.g. the relation between Vessels, Cars and Amphibious Crafts. The *Core* is oriented on DoDAF 2.02 (DoDAF 2011) and 'The International System of Units' (National Institute of Standards and Technology 2008). Therefore the *Core* contains general data types like SIUnit. DoDAF comprises general concepts like 'Activity', 'Capability' and 'Performer'. A 'Performer' comprises an 'Activity' as well as a

'Capability' (DoDAF 2011). An example for a 'Performer' is e.g. the captain of a vessel which can be simulated as a user like explained in Figure 1.

The *World Data Model* (WDM) and the *Component Model* (CM) are in the layer below. The CM contains all entities and parameters which are necessary for the control of the co-simulation components, like explained for the third layer of Table 2.

Data Types to Describe the Current Situation

The WDM is the *Semantic Model* for the description of data types and parameters which are necessary for the communication of the current situation between the simulation components. It is used to describe all necessary data types and parameters to integrate simulators which are necessary for the representation of the Simulation Environment like a shipping area in the case of the example mentioned in the introduction.

The *Geo* part follows the Simple Feature Access which is specified by the ISO19125. It is used to describe geographical data. Therefore it comprises data types like 'GeoPoint', 'Curve', 'Surface' and 'Polygon' that could be both two- or three-dimensional. Furthermore, the model needs to contain data types and parameters for the integration of sensor simulations and physical simulations which are included in the *Sense,* respectively the *Physics* part. In the case of maritime systems it is e.g. required that the *Sense* part includes data types and parameters for the integration of sensor simulations like GPS so that the Maritime Traffic Simulator is able to communicate the vessel positions to a radar simulator. Because of that the *Sense* part is oriented on the *SensorML* (Open Geospatial Consortium Inc. 2007). The Sensor Modelling Language is a XML based standard which is used to describe sensors as well as their measurement process. This standard focusses on the description of sensing devices. It also includes models to describe observations of these devices. The *Vehicle* part is necessary to integrate Vehicle simulators like for Vessels with data types like draught and length of a vessel. The *TrafficSystem* part includes data types like 'WayPoint' and 'Trajcetory'. The *Maritime* part is for the integration of Maritime Simulations with data types like 'Wave' and 'Ocean'.

Safety relevant Data Types in the Maritime Context

In addition, the *Safety* part contains data types and parameters for the recognition of hazards and failures so that rare events can be recognized like described in Läsche et al. 2013. Therefore the *Safety* part is oriented on the *Safety ISO26262* (ISO 2012). This is a specification for functional safety in the automotive area which includes, among others, an optimized vocabulary for its scope (Hagel 2011; Esposito 2010; LDRA 2011). The *Safety ISO26262* is based on the IEC 61508 and contains many general safety relevant elements for the automotive domain. The IEC 61508 is a specification for safety related systems (LDRA 2011; exida 2006). The *Safety ISO26262* can be used for scenarios regarding safety analysis for maritime systems. Because of that it is sensible to derive the relevant data types in the context of safety analysis for maritime systems. For example, data types like hazards and failures can be used to simulate that a failure consists of a concatenation of events. In the case of automotive, a hazard is defined as (Esposito 2010): "A potential source of harm. Harm is a physical injury or damage to the health of people". A failure can be defined as (Esposito 2010): "The termination of the ability of an element or an item to perform a function as required." In the case of maritime systems, an example for a failure is the grounding of a vessel in a curve because of an occurred hazard like the harm of the vessel engine).

The *Semantic Model* is described using the Eclipse Modelling Framework (EMF) (Steinberg et al. 2008) to meet the user requirement for a *Formal Description*. Therefore the EMF ability to generate a valid implementation out of the *Formal Description* can be used. Furthermore, this allows the use of the EMF tool chain for further development.

SEMANTIC MODEL INTO OBJECT MODEL TEMPLATE TRANSFORMATION

This chapter focusses on the transformation of the *Semantic Model* from the third layer of Table 2 into an OMT compliant format. The *Model* describes the common data types and concepts of the various simulators involved in a joint simulation. It is constructed object-orientated, on the basis of the requirements of the previous chapter. A strong use of polymorphism ensures among others the reusability, as described in the chapter above. In contrast, most RTIs are based on a much simpler data model, e.g. without multiple inheritance. This also applies to the HLA implementation used by Läsche et al. (2013) which only allows the use of fundamental types (integer, float, etc.) as well as their array representation. Moreover, the HLA implementation utilizes XML files to describe the OMT. Therefore model driven architecture (s. Brambilla et al. 2012; Stahl et al. 2007) techniques are used to generate the platform specific OMT files from the platform-independent *Semantic Model*. This also eliminates the need to maintain two versions of the same model which is the EMF file of the *Semantic Model* and XML file in the case of the OMT. Figure 4 shows a possible modelling of a dative called *Pose*.

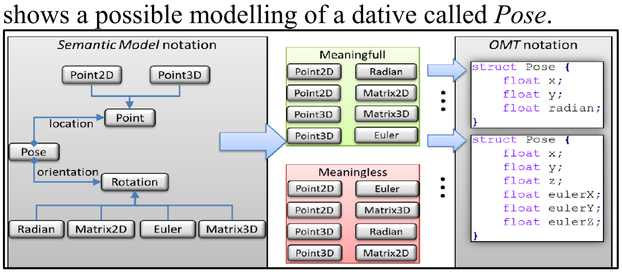

Figure 4: Breakdown of complex data structures to a set of fundamental data types

The *Pose* contains a location (*Point*) as well as an orientation (*Rotation*). Both types can be specialized for a two or three dimensional simulation, like mentioned in the introduction (upper side of the figure).

A *Pose* can exist in at least eight different variations, like displayed on the lower left part of Figure 4, where four meaningless combinations (mixing of dimensions) and four meaningful combinations can be found. The premise is in this case, that an HLA Object cannot contain a complex data type. Therefore the last step, each of these combinations has to be decomposed into a set of fundamental data types (like displayed on the rightmost side of the figure) to fit the requirements of the platform specific OMT. Moreover, it involves that each object of the *Semantic Model* which constitutes a *Pose*, e.g. a vessel can also exists in at least eight different "flat" variations of a *Pose* like mentioned in Figure 4. In a more complex model this would lead to a huge number of possible variations for a single object, also containing a huge amount of meaningless combinations of attributes like in Figure 4.

The simulators involved in a joint simulation need to agree on a combination of attributes in reality since in most cases each data type need a special treatment. For example, a *Pose* is the combination of a *Point3D* as *location* and an *Euler* as *orientation*.

Manual Selection of Concepts

In addition, not all information available within the model is needed to couple two or more simulations. For example, the maritime simulation mentioned in the introduction does not require knowledge about the number of crew members on the vessel, since it does not affect its dynamic behavior.

Taking these arguments into account, the authors propose a manual selection of concepts and their attributes, to perform the step from the semantic layer to the syntactic layer of Table 2. As additional benefit this will also lead to a lower usage of network bandwidth, to synchronize the simulations.

The selection of Concepts and attributes is then done based on the *Semantic Model*, in a form like:

> From Concept *Vessel* take the attribute *pose* as type *Pose*, from this *pose* take the attribute *location* as a *Point2D* and finally take the attribute *X* as the communicated value.

The formal description of the *Semantic Model* then may be used to detect the fundamental data type of the requested attribute. This allows the automatic generation of RTI-Configuration files, combining the *Semantic Model* as well as the selection of fundamental attributes. In case of the HLA-OMT Files mentioned in Läsche et al. (2013) this is done using a Model to Text transformation, based on the XPand Project of the Eclipse Modelling Tools (Seidl 2013).

Currently, the transformation from the semantic model into the HLA-OMT format is supported. In the future, other inter-process communication technologies, such as CORBA (OMG 2012) or ZeroC-ICE (Spruiell 2011) should be supported.

CONCLUSION AND NEXT STEPS

This article analyses the *Interoperability* in co-simulations of maritime systems. Therefore the requirements for these distributed simulations are derived by a problem-oriented as well as technical-oriented approach in the first chapter. The main part is the combination of the IEC compatibility level matrix with the conceptual architecture for simulation-based analysis. The paper introduces architecture with six layers for co-simulation *Interoperability* which defines requirements for co-simulations of maritime systems. A central role has the *Semantic Model* because it helps to insure the satisfaction of the derived user requirements: *Formal Description, Observability, Controllability* and especially the *Interoperability.* Therefore the semantic layer contains the *Semantic Model.* Its structure is based on the described design rationales and standards. The automatic model transformation is necessary to build a useful bridge between the *Technical Layer* and *Semantic Layer*. Its practical use is to insure the platform specific implementations consistency to the semantic model. Furthermore this automatic process accelerates the implementations creation.

One of the next steps is to check additional standards like the S-100 data model (IHO 2010) which supports the integration of hydrographic data and applications. In this case, it will be analyzed regarding useful elements for the *Semantic Model* to improve its *Interoperability* for the integration of simulators of maritime environments. Furthermore, a case study will be used to evaluate the *Interoperability* of the co-simulation in the case of the integration of a maritime traffic simulation (MTS) and radar simulations. The goal is to extend the *Semantic Model* so that the necessary generated ship data of the MTS can be processed by components for sensor simulation and AIS data generation. This allows testing and improving sensor systems.

REFERENCES

Bloch, J., 2006. How to design a good API and why it matters. *Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications - OOPSLA '06.*

Brambilla, M., Cabot, J. & Wimmer, M., 2012. *Model-Driven Software Engineering in Practice: Synthesis Lectures on Software Engineering*, Morgan & Claypool Publishers.

DoDAF, 2011. The DoDAF Architecture Framework Version 2.02. Available at: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf [Accessed March 3, 2011].

Droste, R. et al., 2012. Model-Based Risk Assessment Supporting Development of HSE Plans for Safe Offshore Operations. , pp.146–161.

Esposito, C., 2010. *Hands on the ISO 26262 Standard*, Napoli: Università di napoli Federico II. Available at: http://www.critical-step.eu/index.php/meeting-

documents/doc_download/78-training-presentation [Accessed February 10, 2014].
exida, 2006. *IEC 61508 Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, Sellersville, Pennsylvania. Available at: http://www.win.tue.nl/~mvdbrand/courses/sse/1213/iec61508_overview.pdf [Accessed February 1, 2014].
Gale, C.H., 2007. Improving navigational safety. , pp.4–8.
Gurp, J. Van & Bosch, J., 2003. *Separation of Concerns : A Case Study*, Groningen. Available at: http://www.researchgate.net/publication/2563874_Separation_of_Concerns_A_Case_Study/file/79e4150c04dcf162ad.pdf [Accessed February 13, 2014].
Hagel, J., 2011. *Herausforderung Funktionale Sicherheit*, Sindelfingen: MBtech. Available at: http://www.mbtech-group.com/fileadmin/media/pdf/electronics_solutions/Herausforderung-funktionale-Sicherheit_Web.pdf [Accessed January 30, 2014].
IEC, 2002. *TC65: Industrial Process Measurement and Control*,
IHO, 2010. *S-100 Universal Hydrographic Data Model*, Monaco: International Hydrographic Bureau.
IMO, 2012. *Report to the Maritime Safety Committee, NAV 58/WP.6/Rev.1*,
ISO, 2004. *ISO 19125-2:2004 Geographic information - Simple feature access*, Geneva. Available at: http://www.ird.fr/informatique-scientifique/methodo/standards/normes_iso_ogc/iso_geo/afnor/ISO_19125-2_DE_2004-ANGLAIS.pdf [Accessed December 28, 2013].
ISO, 2012. *ISO 26262-10:2012 Road vehicles - Functional safety*, Geneva.
Läsche, C. et al., 2014. MODEL-BASED RISK ASSESSMENT OF OFFSHORE OPERATIONS. In *Proceedings of the 33rd International Conference on Ocean, Offshore and Arctic Engineering*. San Francisco, CA, USA: ASME.
Läsche, C., Gollücke, V. & Hahn, A., 2013. Using an HLA Simulation Environment for Safety Concept Verification of Offshore Operations. In *27th European Conference on Modeling and Simulation*. Alesung, Norway.
LDRA, 2011. *ISO 26262 the Emerging Automotive Safety Standard*, LDRA.
National Institute of Standards and Technology, 2008. The International System of Units [SI] B. N. Taylor & A. Thompson, eds. , 9(7), p.97.
Noulard, E., Rousselot, J. & Siron, P., 2009. CERTI , an Open Source RTI , why and how. In *Spring Simulation Interoperability Workshop*. San Diego. Available at: http://oatao.univ-toulouse.fr/2056/ [Accessed November 3, 2013].
OMG, 2012. Common Object Request Broker Architecture (CORBA). Available at: http://www.omg.org/spec/CORBA/ [Accessed February 5, 2014].
Open Geospatial Consortium Inc., 2007. *OpenGIS® Sensor Model Language (SensorML) Implementation Specification*, Huntsville.
Puch, S. et al., 2012. Rapid Virtual-Human-in-the-Loop Simulation with the High Level Architecture. In *Proceedings of Summer Computer Simulation Conference 2012 (SCSC 2012)*. pp. 44–50.
Schütte, S., 2013. *Simulation Model Composition for the Large-Scale Analysis of Smart grid Control Mechanisms*. Oldenburg: Universität Oldenburg.
Seidl, C., 2013. The Eclipse Modeling Framework (EMF): A Practical Introduction and Technology Overview. Available at: http://st-teach.inf.tu-dresden.de/wiki/images/The_Eclipse_Modeling_Framework_%2528EMF%2529_-_Christoph_Seidl_%2528commented%2529.pdf [Accessed October 10, 2013].
Spruiell, M., 2011. ZeroC Documentation. Available at: http://doc.zeroc.com/display/Doc/Home [Accessed February 6, 2014].
Stahl, T. et al., 2007. *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*, Heidelberg: dpunkt.verlag.
Steinberg, D. et al., 2008. *EMF Eclipse Modeling Framework* 2nd ed., Boston: Pearson Education, Inc.
Tulach, J., 2008. *Practical API Design: Confessions of a Java Framework Architect*, New York: Apress.
Wortelen, B., Lüdtke, A. & Baumann, M., 2006. Integrated Simulation of Attention Distribution and Driving Behavior.
Zimmermann, H., 1980. OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communication, Vol . COM-28, No.4 April*, pp.425–432.

AUTHOR BIOGRAPHIES

CHRISTOPH DIBBERN received his M.Sc. in Information Technology in 2012 at the University of Applied Science Kiel and started to work as a software developer for Business Process Management Systems at the company ESN innovo GmbH afterwards. Since 2013, he has been working at the University of Oldenburg within the project CSE (Critical Systems Engineering). His E-Mail address is: *dibbern@wi-ol.de* and the homepage of his group is http://be.wi-ol.de/.

AXEL HAHN is full professor at the University of Oldenburg and leads the working group Business Engineering and board member of the division Transportation at the research institute OFFIS. His research topics are safety and efficiency in marine transportation systems. His E-Mail address is: *hahn@wi-ol.de* and the homepage of his group is http://be.wi-ol.de/.

SÖREN SCHWEIGERT received his Dipl. Inform. in 2009 at the University of Oldenburg and started to work at OFFIS in the research group CMS (Cooperative mobile systems) afterwards. Currently he is working within the project COSINUS with focus on Maritime Simulations. His E-Mail address is: *soeren.schweigert@offis.de* and the homepage of his group is http://www.offis.de/en/start.html.