

OPEN FRAMEWORK FOR DIGITAL TWIN SHIP DATA: CASE STUDIES ON HANDLING OF MULTIPLE TAXONOMIES AND NAVIGATION SIMULATION

Reference NO. IJME 813, DOI: 10.5750/ijme.v165iA1.813

Í A Fonseca, F F de Oliveira and H M Gaspar, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU), Norway

KEY DATES: Submission date: 23.11.21 / Final acceptance date: 28.04.23

SUMMARY

The digital twin concept describes a comprehensive model used to monitor and simulate an existing asset in nearly real-time. In a maritime context, digital twins do not yet have a wide selection of standards to support interoperability among systems, undermining users' ability to create a cohesive digital twin that combines multiple sources of data. This study proposes an open framework for modelling and organizing ship, sensor, and operating context data in digital twins. The framework is web-driven, allowing digital twin content and simulations to be accessed by distributed users. Two case studies based on the framework are presented. The first one explains how to manage ship and sensor data using taxonomies suitable across different stages of the lifecycle. A dashboard for browsing a digital model of the research vessel Gunnerus according to taxonomies for early design and operation is developed as a proof of concept. The second case study illustrates how the proposed data standards can be used to develop simulations accounting for some aspects of the vessel's operating context. The digital Gunnerus model is used to develop a navigational simulator including surrounding terrain and nearby vessels. The challenges anticipated in scaling the framework toward industrial applications are discussed in terms of standards' capabilities, compatibility, and of data governance. Plans of integrating the case studies with sensor streams to obtain true digital twins are outlined.

KEYWORDS

Digital twin, digital thread, data standard, taxonomy, interoperability, ship, maritime.

NOMENCLATURE

API	Application Programming Interface	R/V	Research Vessel
AIS	Automatic Identification System	SBSD	Systems-Based Ship Design
BIM	Building Information Model	SOLAS	International Convention for the Safety of Life at Sea
CAD	Computer-Aided Design	VIS	Vessel Information Structures
CG	Centre of Gravity	WebGL	Web Graphics Library
DEM	Digital Elevation Model	1.	DIGITAL TWINS APPLIED TO THE SHIP INDUSTRY
DNV	Det Norske Veritas	1.1	DEFINITIONS AND ROLE THROUGH THE LIFE CYCLE
FMI	Functional Mock-up Interface		
glTF	Graphics Language Transmission Format		
HTML	HyperText Markup Language		
IFC	Industry Foundation Classes		
ISO	International Organization for Standardization		
IMO	International Maritime Organization		
JSMEA	Japan Ship Machinery and Equipment Association		
JSON	JavaScript Object Notation		
JT	Jupiter Tessellation		
MIT	Massachusetts Institute of Technology		
MQTT	Message Queuing Telemetry Transport		
NTNU	Norwegian University of Science and Technology		
PDM	Product Data Management		
PLM	Product Lifecycle Management		
PSV	Platform Supply Vessels		

A digital twin is an integrated simulation that uses sensor data and other measurements to mirror and predict the behaviour of a physical system during its built life (Defense Acquisition University, u.d.). Use of operational data to feed simulation models is not new in specialised domains. However, digital twins have been present in their current formulation only since a 2010 NASA draft report, which outlines an ambitious vision where it acts as a central data hub to monitor and assess system operation in real-time (Shafto *et al.*, 2010). Among the use cases outlined for the digital twin, are simulating an operation before it is executed and aiding forensics of a potential fault or damage. Based on this definition, Erikstad (2017) explains digital twins must include models and data for ship representation, ship states, behaviour, and operating contexts. Ship representation

is roughly a product model that contains relevant data about a vessel. State data describe its operating situation; it is derived from logs measured by instrumentation systems installed in ship machinery and equipment. The operational context contains data and models required for evaluating the vessel's interaction with the environment surrounding it. The digital twin takes these contents as inputs to execute simulations or analyses of behaviours in various domains, aiming to provide stakeholders with actionable information about the system. According to this definition, the digital twin does not involve handling of data throughout the lifecycle but focuses specifically on aiding system operation. An expanded view of data management that establishes guidelines and processes to control interplays among data, software, information, and knowledge to inform decision-makers throughout a system's lifecycle has been gaining popularity under the term *digital thread* (Defense Acquisition University, n.d.). Under this framework, a common ship representation (i.e., product model) supports activities throughout all lifecycle stages, up to launch. It is then augmented with sensor data collected from the operating vessel and linked to behavioural models, yielding a digital twin. This vision aims to enable not only a smoother knowledge flow of a given vessel's lifecycle but reuse of knowledge and models from previous builds during future ones.

For instance, van den Hamer and Lepoeter (1996) present five dimensions relevant to the problem of managing product data: versions, views, hierarchy, status, and variants. The authors previously argued the dimensions of views and hierarchy were identified as central to effective management of system representation and sensor readings in digital twins, as they allow hierarchical mapping of content to, e.g., ship's systems, weight groups, or compartments, depending on the adopted view (Fonseca and Gaspar, 2020). While the other dimensions have reduced importance given the particularities of the ship industry, they might still play significant roles. Since shipbuilding is commonly based on individual tenders, rather than product platforms, each digital twin represents a unique system. Thus, the variants dimension is not so useful to describe differences between multiple ships as it is to consider possible variants of systems and components inside them (e.g., types of propulsion arrangements, such as diesel or diesel-electric). The dimensions of versions and status are particularly important when managing the evolution of a product model leading to construction. This is useful to support approval by class societies, building status, and other features. During operation, they can be used to issue updated versions of the product model (e.g., after a retrofitting), displaying systems' condition and uptime, and managing eventual maintenance notes for an asset.

1.2 STATE OF RESEARCH

Since then, sustained advances in computational capacity, wireless connectivity, sensor technology, and internet of

things are enabling the elevation of the scope and detailing of such tools to a higher level. Up to this day, there is still a considerable production of theoretical research on the digital twin concept and its intended implementation. Discussion on definitions and potential applications, assessments of challenges and enabling technologies (Fonseca and Gaspar, 2020), proposal of high-level architectures (Giering and Dyck, 2021) and of design patterns (Erikstad and Bekker, 2021). The need for this body of research is attributed to a combination of factors. Some of them are the novelty of the digital twin concept, its ambitious scope and relatively vague definition. While there is interest in using digital twins, or at least high frequency sensor data, to improve processes and operations, it is not always clear how to realise this interest. Experience in other industries, even those that are household names of information technology, suggests such identification is largely a process of trial and error as researchers and practitioners explore and understand the potential of new technologies (Nokkala, Salmela and Toivonen, 2019). Another factor are the specific infrastructure challenges associated to maritime applications (Section 1.3). Together, they raise the need to pair useful digital twin concepts with feasible maritime implementations.

In addition to the theoretical basis, this novelty is also reflected on the applications of digital twins presented on the maritime literature, meaning they are currently not on par with the comprehensive hubs digital twins are aimed to become. For instance, applications of digital twins to broader life cycle services tend to be concepts at early stage of development. Taylor *et al.* (2020) describe existing initiatives to develop a digital twin shipyard and a smart port. Applications at a later stage of development usually address specific operation support concerns. Some focus on providing real-time guidance to operation. Lee *et al.* (2022) present a digital twin with the objective of guiding ship operations through routes which minimise resistance and motions. The digital twin developed by Wei *et al.* (2023) provides continuous routing recommendation with the objective of obtaining compliance with environmental regulations such as the Carbon Intensity Indicator. Other cases focus on supporting maintenance, such as estimating hull condition by measuring speed loss due to marine fouling (Coraddu *et al.*, 2019). Yet others are bridging the gap between theoretical models and sensor-gathered data. Nielsen, Mounet and Brodtkorb (2021) present usage of data acquired during operations to tune motion transfer functions originally obtained with simulation packages. Liu and Ren (2022) propose a method to correlate local stresses measured through sensors with evaluation of ship hull structure strength.

A determinant factor for future rollout of concepts like these is their ability "to scale", meaning both the feasibility of implementing them on industrial settings (as opposed to only controlled research setups) and of replicating this implementation with the amount of dispended resources

(both human and computational) growing at maximum at a linear rate in relation to the number of supported vessels. On that note, a previous study by the authors presented a digital twin of an experiment with a scale model PSV that navigated in a wave basin (Fonseca *et al.*, 2022). It provided real-time simulation, control, and experiment rewind. While the experiment was functional, its full-scale implementation is far-fetched given current technologies for remote vessel operation. This work takes a more pragmatic approach of leveraging available resources to develop digital twins which are less functionally mature but can be implemented and replicated in full-scale.

1.3 CHALLENGES TO REALISATION IN INDUSTRY

Several challenges exist to realizing cohesive digital twins that account for varied behaviour domains in the maritime industry. At the infrastructure level, widespread sensor systems that are supported by robust networks and data exchange protocols, while reconciling databases owned by distributed stakeholders through the maritime value chain, e.g., ship owners, classification societies, and port authorities, are required. Regarding software and data, it is necessary to overcome the segmentation of digital tools in the maritime industry toward greater interoperability among such systems. After a few decades, during which multiple players attempted to establish closed systems of increasing scope (Nowacki, 2010), it is now possible to observe two major trends in the sector — PLM/PDM suites that began catering to maritime by offering ship design tools, and specialized computer-aided ship design software that allows more flexible connections among external suites (Gaspar, 2019). Greater interoperability among systems reduces the need for rework when exchanging models among suites, leading to smoother workflows when chaining multiple software packages. However, to realise this possibility, standards that allow data modelling, archival, and exchange are needed.

A previous report by the authors gave a brief account of digital twin standardisation challenges compared to previous

standardisation initiatives in the ship domain (Fonseca and Gaspar, 2020). While there is a growing perception about the need for open standards as enablers to digitalisation in maritime, it has traditionally been challenging to achieve their adoption on a purely voluntarily basis. Take for comparison the Industry Foundation Classes (IFC) standards, which support Building Information Models in the civil engineering sector (ISO, 2018a). Adoption of IFC standards in various countries is incentivized by regulatory measures which require public contractors to hand over BIMs for the projects they execute (Edirisinghe and London, 2015). In the absence of such regulatory devices, an alternative for the maritime sector would be to compromise the scope and strictness of standardisation to accommodate variations desired by potential adopters. Table 1 compares the traditional approach to compatibility in computer-aided design and engineering systems, including those aimed at ship design and operation (middle column) and an emerging approach which can be already observed in the design of novel standards and in the commercial offerings of some digital vendors (right column). The last two rows describe a design choice observed in some recent standards and in the development of maritime simulations based on the Functional Mock-up Interface (FMI) (Hatledal *et al.*, 2021). The adherence to the standard specification ensures some level of compatibility, however, stakeholders will need to further coordinate the implementation to achieve full interoperability in each project. E.g., FMI-based simulation models will be more easily linked if the involved parties have agreed in advance to an architectural template of the system they wish to simulate and to adopting the same physical quantities and units for corresponding interfaces.

2. WEB-DRIVEN DEVELOPMENT

2.1 MOTIVATION AND CORE TECHNOLOGIES

In response to interoperability concerns, a standards-based, web-driven approach is adopted to model and develop digital twin ships. The web is an example of successful standardisation, with its core technologies enabling

Table 1: Summary comparison of compatibility approaches in earlier and novel digital systems

	Monolithic digital systems	Heterogeneous digital systems
Digital content consumption and exchange	Data and models stored in proprietary formats tied to specific proprietary solutions.	Data and models stored in broadly supported formats compatible with various tools, even open source ones.
Commercial tactic and scope	Licensed software with self-contained functionality and data content.	Subscription services chained with each other and sharing a common data pipeline.
Standardisation initiatives' scope	Strict specification of ship models and data.	Compromise solution allowing for customized implementations and use cases.
Standardisation initiatives' aim	Full compatibility of a given model across systems. Did not gain adoption in practice.	Partial compatibility of model across systems, plus consistency respective to the chosen implementation.

compatibility of software across computer systems. HTML and CSS allow developers to format hypertext and apply graphic styles to websites, respectively. JavaScript was originally designed for execution of scripts inside websites, but has now reached a much broader scope. It is based on the ECMAScript specification developed by Ecma International, a non-profit standards organisation. Such languages are tested extensively, and their broad adoption across web browsers allow websites and web applications to be compatible with any device capable of running these browsers, such as smartphones and laptops. This compatibility does not come at a great performance cost, since benchmarks show that for many tasks, JavaScript performs similarly to Python, which is a popular choice for scientific app development (Gaspar, 2017).

Web-based development offers advantages in terms of app testing and distribution. Since algorithms are written in a scripting language, it is unnecessary to compile software before executing it. Thus, browsers function also as a responsive environment for testing and debugging web applications. Since apps are always accessed online, they are deployed immediately to users as soon as a developer updates the source code hosted on a server. Regarding interoperability, a web-based approach offers not only compatibility for execution of scripts across devices, but is compatible with technologies for data serialisation, storage, and exchange across applications. For example, JSON is a notation format that organises data into either objects that consist of unordered pairs of names and properties, or arrays that are ordered lists of values. JSON is syntactically simple, intelligible to human readers, and supported widely across programming languages. Besides exchanges of serialised data, it is possible to link applications to protocols for real-time data exchanges, enabling development of simulations based on live data streams. For example, WebSocket provides bidirectional data streaming between two devices (i.e., full-duplex communication). Such technologies offer the flexibility to connect web applications with other platforms, including proprietary ones. While a web-driven approach was adopted, the standards allow interoperability with software that is unrelated to web-development. For example, text-based JSON files make data accessible and easily readable. The selected 3D formats can also be opened using several lightweight applications, avoiding the need for complex software suites for simple visualisation tasks.

2.2 DEVELOPMENT AND GRAPHICAL USER INTERFACE

Web development has a strong culture for open source development, backed not only by small projects that independent groups manage, but for-profit companies that contribute with open source projects as part of their business strategies. Such culture might simplify development by allowing reuse of code contained in existing libraries or web applications. There exist several

JavaScript libraries for creation of graphical user interfaces, including, creating graphical interfaces, rendering 2D and 3D visualisations. Two of them have been particularly important to development of case studies. The first one is Three.js for creating and displaying 3D graphics using the standardised WebGL programming interface. It simplifies rendering of meaningful web-based graphics and animations. The library repository provides loaders and exporters for several visualisation formats, allowing users to easily pre-existing models in a supported format to web applications. The second library is OpenBridge, a user interface architecture aimed at increasing design consistency in ship bridges integrating systems from different vendors (Nordby, Mallam and Lützhöft, 2019). During development of web and augmented reality applications, OpenBridge allows inclusion of indicators which convey the digital twin's navigation states to users.

3. DATA STANDARDISATION

3.1 DIGITAL TWIN SHIP DATA CONTENTS

Taking clue from the principles stated in Table 1, this section outlines an extendable framework for modelling digital twin ship data. The selection of standards prioritises alternatives that are open, neutral, human-intelligible, and supported broadly. However, in some cases it is needed to compromise some such characteristics, e.g., by using ISO standards that are copyrighted and licensed under fees. The exposition follows the characterization of digital twin content in Section 1.1: ship representation, shipboard data servers, sensor logs, operating context, and taxonomies for organising ship data. Standardisation of ship data servers and metocean data are not addressed in the case studies, so only a brief overview is given to those interested in obtaining references for further reading. The sections do not discuss the topic of behavioural model standardization, as it is judged too extensive for this work and would require further research. However, Case Study 2 illustrates how the framework can link to digital twins including simulations of behaviour, even if the sim themselves are not standardized.

3.2 SHIP REPRESENTATION

3.2 (a) Existing Alternatives in the Ship Industry

One of the most significant initiatives for standardisation in the ship industry has been the development of ship-specific application protocols (APs) for the ISO 10303 standard, namely Automation systems and integration — Product data representation and exchange, and informally known as STEP. STEP's objective has been to establish common formats and data structures for exchange of data about engineering systems. The provision of ship-specific APs allows the standard to model data covering ship arrangements, moulded forms, structures, and mechanical systems. Other general APs can be applied to modelling

of ship systems, e.g., the plant spatial configuration and the electromechanical design and installation APs, which together cover piping, HVAC and electrical distribution systems (Fonseca and Gaspar, 2020). The literature review in (Fonseca, 2022) explains there has been a low industrial uptake of STEP. The reasons for this include the burden of implementing the standard together with its supporting information modelling language (i.e., EXPRESS) and the split incentives of this implementation between software vendors and users. It seems thus that the ongoing interest in digital twins is more likely to bring about new standardisation initiatives for ship representation rather than reviving interest in STEP renovation and adoption. In that sense, a relevant candidate format for implementation of future digital twins is the Open Class 3D Model Exchange, which aims to establish a standard in the ship design and construction sector (OCX, u.d.). Unfortunately, the public specification only came to the authors' awareness after this study had already been developed. The investigation of its trade-offs remains an interesting topic for future research.

3.2 (b) Vessel.JS' Ship Model

Ship representation plays several roles in a digital twin. It helps users acquire awareness about a vessel by allowing inspection of its appearance and spatial arrangement, identification of components, and browsing of sensor logs connected to each component. It also collects physical and geometric data that can later be used as inputs for behaviour simulations. Considering such requirements, ship representation must include a 3D model of the vessel and metadata about weight and materials. In a previous report, the second author presented Vessel.js, a library that already incorporates the principles of the ship representation proposed here, but it was scoped toward simplified 3D models for early stage ship design (Gaspar, 2018). The library is explored here as an alternative for modelling digital twin ships even if it does not fulfil all formal and operational requirements necessary to qualify as a standard. Formal requirements include extensive specification of mandatory and optional data properties, definition of physical units, etc. Operational concerns include publishing and long-term support, for example. Thus, the library is studied as a modelling approach which might inform standardisation initiatives.

Since the library was developed for the web, it prioritises 3D formats based on polygons (i.e., STL) rather than splines, since they do not require a complex geometric modelling kernel to be rendered. Even with the new objective of creating a digital twin to gather live insights during operation, it continues to be an appropriate choice because the lightness of polygonal formats makes them better suited to real-time rendering. The library allows users to create an object-oriented ship model stored as a JSON file that contains geometry and weight of the hull, decks, bulkheads, tanks, and components. Each object contains a property designated to store identification tags

according to one or more taxonomies. Hull geometry is generated automatically from a table of offsets, and its weight is calculated from an empirical formula. Structural elements are created parametrically as flat planes without reinforcers, and their weights are calculated based on material volume.

Besides the hull and structure, Vessel.js also defines concepts of base and derived objects, which are essential to the framework's flexibility. A base object can represent a tank, compartment, assembly, or component, listing the dimensions and weight of an element. Geometry is loaded from an STL file, and if a file is unavailable, a box of corresponding dimensions is generated automatically as a placeholder. For components and compartments, weight data are lightweight and CG. In case the element being modelled is a tank, data include volume capacity, content density, and a list of CG coordinates that correspond to levels of tank filling (i.e., from empty to full). A derived object replicates a given base object in a position inside the vessel, reducing the data necessary to model identical components. The library's general focus on describing vessel spaces and parametric geometry is a limitation to modelling detailed asset constitution. Thus, the library needed some modification to be usable with digital twins. A second ship model that is suitable during operational stage and loads detailed 3D models of components is introduced. The approach with base and derived objects still applies, with a JSON file listing every component in the digital twin with position and a link to a 3D model. The latter are stored separately in unique files that multiple derived objects can invoke when placing identical components inside of the vessel. The glTF format is useful to model detailed components because it allows exchanges of models with colours, textures, and assemblies (Robinet *et al.*, 2018). Both STL and glTF formats can be easily loaded to web-based environments with Three.js.

3.3 SHIPBOARD DATA SERVERS AND SENSOR LOGS

A position paper by DNV (Låg, Vartdal and Knutsen, 2015) gives a comprehensive overview of status and challenges to ship connectivity. It predicted that new communication capabilities and increases in available bandwidth would become enablers to applications transferring between ship and shore in real-time such as condition and environmental monitoring, energy efficiency optimisation, and remote maintenance. Two recently published ISO standards might aid digital twin interoperability in this regard. The first one is ISO 19847 standard, namely Shipboard data servers to share field data at sea (ISO, 2018b). While exchange of navigational data is commonly enabled by the IEC 61162 series of standards (IEC, 2016), machinery data is often non-standardised. Thus, ISO 19847 specifies requirements for servers which aggregate, store and retransmit ship machinery data with respect to performance, function, service, and safety.

The second standard is ISO 19848, namely Standard data for shipboard machinery and equipment (ISO, 2018c; Ando, 2019). It defines machine and human-readable identifiers and data structures for exchange of such machinery and equipment data. While the standard has been aimed primarily at data exchanges onboard ships, it has implemented features for universal data identification on the internet. The document defines package templates of two types:

- A data channel list that contains metadata about the data channel—descriptions of variables, data channel types (e.g., instantaneous, average, calculated, and manual input by the crew), update cycles, measuring units, and expected ranges and various identification tags (e.g., ship ID, naming rules for the taxonomy, and local and short IDs for each variable).
- Time series data, with measurements in the form of pairs that list respective time stamps and sensor readings. The package contains only a few identification properties, such as author, ship ID, and short ID.

The document specifies JSON and Extensible Markup Language (XML) implementations for data channel lists and time series data, and Comma-Separated Values (CSV) for times series data, recommended for long log sequences. Available data channel types allow a shipowner various strategies of data storage when implementing a digital twin. For example, if a vessel state derives from calculations based on one or more sensor logs, it is possible to choose between storing the raw data as instantaneous or resulting values as calculated. The time series package allows modelling tabular or event data, with tabular used for sequences of values expected at regular intervals (e.g., sensor logs) and event for data occurring irregularly (e.g., alarm warnings or manual inputs by crew). Identification tags included in a package allow existing data to be organised according to a naming rule (i.e., a taxonomy as Section 3.6 discusses). It is still early to assess the impact both standards will have in exchange of ship machinery data due to at least two factors. The first is the lead time between standard publication, implementation, and launch in commercial systems. The second is the relatively long operational life cycle of vessels, meaning the uptake of new technologies might not be observed as a swift trend. In any case, development and maintenance of both standards is ongoing, and preview drafts of their new editions have already been published.

3.4 OPERATING CONTEXT

3.4 (a) Topography

Surrounding topography provide remote crew location awareness in coastal and short-sea shipping, the navigation modalities which are the strongest candidates for automation, and by extension provide strong opportunities

for application of digital twins. Water depth profiles can occasionally assist in the identification of suitable draught in ports or shallow water regions. Fonseca *et al.* (2022) mention the Norwegian Mapping Authority provides free topographic data sets of Norway in the Digital Elevation Model (DEM) standard by the U.S. Geological Survey, a legacy and still used format. Sandvik (2013) provides a series of tutorials explaining how to use these data sets to generate web-based 3D models with Three.js. This work applies his procedure to obtain environments inside which vessel operations can be simulated. The first step of the process consists in exporting the data contained in the DEM files to a height map. The height map is used to generate colour and shading profile of the terrain, which are merged into a texture file. The elevation data is stored within JavaScript typed arrays, a data structure suitable for encoding and manipulating raw binary data in web applications. When displaying the 3D models, a script generated the region profile as a tessellated geometry based on the elevation data, then loads the texture to obtain a final render.

3.4 (b) Surrounding Vessels

The International Convention for the Safety of Life at Sea (SOLAS) enacted by the International Maritime Organisation (IMO) requires ships of 300 gross tonnage and upwards engaged on international voyages, ships of 500 gross tonnage and above, and all passenger vessels irrespective of size to be fitted with an Automatic Identification System (AIS) onboard. The AIS provides information about the ship's identity, type, position, course, speed, among others. It also receives the same information from AIS installed on other ships. While the AIS and infrastructure was originally devised to monitor and track vessels with the objective of preventing accidents, the fact that its data is exchanged with shore-based facilities through satellites has spanned a series of other applications such as estimation of carbon dioxide emissions, study of maritime trade patterns, and identification of vessels operating stealthily (i.e., "dark vessels"). There are several services which sell subscriptions to Application Programming Interfaces (API) providing access to live AIS data. In the context of digital twin development, these resources can be used to provide an updated overview of the vessels operating in a region for remote monitoring purposes.

3.4 (c) Weather Condition

The case studies in this work do not include modelling of metocean data (such as wave condition, current and wind speeds). Fonseca *et al.* (2022) explain that metocean data has traditionally enjoyed more standardisation than ship data, in part triggered by its shared use between public and private stakeholders. However, access to live data for a certain operating region might not be straightforward, nor is the aggregation of different data

sources to extend coverage over different regions. In terms of visualisation, the same work shows how a 3D model of water elevation might be linked to real-time data to calculate and display regular wave frequency and period. Future work could improve the visualisation to render irregular waves, readying it for linking to data on realistic sea states.

3.5 TAXONOMIES FOR ORGANIZING SHIP DATA

Adequate choices of views and hierarchies play a role not only for a digital twin, but ship data through its entire lifecycle (Pal, 2015). Existing schemas for handling such data are referred to as taxonomies (Låg and With, 2017), which adopt a view of a ship and organise data hierarchically according to it (Figure 1). An example of a traditional view used in the industry is one based on weight, in which ship lightweight is divided into groups for structure, machinery, outfitting, etc. However, during early design stage of complex, innovative warships or offshore support vessels, designers might use a systems-oriented view that is readily translatable to the functional requirements that a design should fulfil. Since, at this stage, designers are concerned primarily with allocating required functions at an architectural level, accompanying hierarchies do not allow great detailing, which is the case of taxonomies employed by early design methods, such as System Based Ship Design (Erikstad and Levander, 2012) and the Design Building Block Approach, the most comprehensive and updated exposition of which being available in (Andrews, 2018). Such methods arrange a ship in major spaces, intended to accommodate a system without identifying each individual component contained in them. As the lifecycle progresses, engineers gradually detail the initial design, culminating in a viable proposal for building. During this stage, the view shifts from functional systems to one suitable to manufacturing. One approach is to divide a ship into spatial zones that encompass all built and installed components in the region. The corresponding hierarchy divides the vessel in terms of blocks, assemblies, and components suitable for construction. If a shipyard adopts group technology, it is adequate to divide the build into groups that share the same set of manufacturing

challenges, even if they are not placed proximately inside of the vessel.

During commission and operation, the view returns to a functional perspective, as the shipowner and classification society attempt to ensure that the vessel is operating effectively and in good condition, whether they use a digital twin to accomplish that objective. Since the ship is already built and launched, the hierarchy, at this stage, should provide great detailing — from major systems down to subsystems, components, and parts. In the geographic location of the authors. For civilian transport and service vessels such principles are found, e.g., in the SFI group system, a code-based classification system developed by the Norwegian Ship Research Institute and now licensed commercially by SpecTec (SpecTec, 2001), and in the Vessel Information Structures, a primarily functional description with which DNV supports its classification activities (Låg, Vindøy and Ramsrud, 2021). VIS describes functions that can be ultimately assigned to an adequate component, accounting for system variants by prescribing function selections of possible alternatives. Moving toward standardisation, DNV has uploaded and documented VIS on the web (DNV, u.d.). Similarly, the Japan Ship Machinery and Equipment Association provides the JSMEA Codebooks for various operational domains as alternative naming rules publicly documented on the web (JSMEA, u.d.). Annexes B and C of the ISO 19848 document describe implementation of JSMEA and VIS as taxonomies for sensor data, respectively. Both rules follow different strategies for construction of data channel names. JSMEA-MAC lists a smaller number of names for commonly used data channels, such as main engine load, running hours, and exhaust gas temperatures. DNV VIS establishes common elements which can be used to concatenate identification names for virtually any data channel existing inside a vessel. This trade-off makes the rule’s naming coverage more ambitious, at the expense of possibly complicating creation of consistent data channel names. On a final note, when a ship is decommissioned and eventually scrapped, it is useful to have data that lists components and materials to be recycled, including notes regarding toxic and dangerous substances (Andrade, Monteiro and Gaspar, 2015).

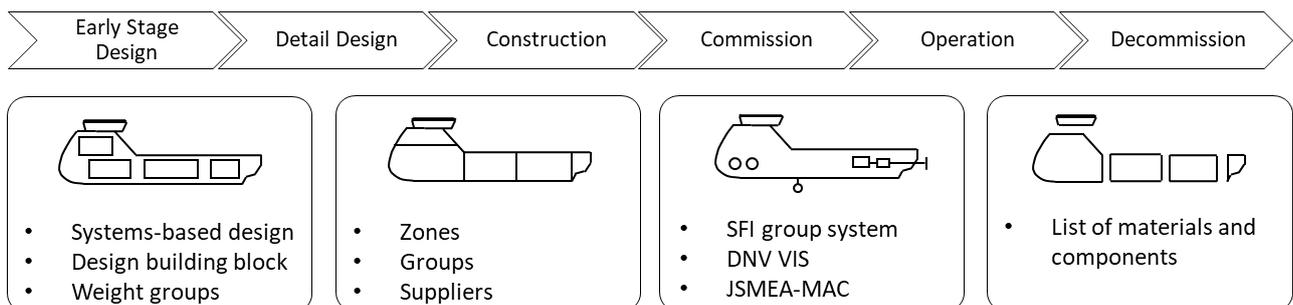


Figure 1. Examples of taxonomies appropriate to various lifecycle stages

4. DIGITAL TWIN OF R/V GUNNERUS

4.1 DIGITAL TWIN AND WEB-APPLICATIONS

An application was developed to illustrate how a digital twin can reconcile taxonomies and use data for asset visualisation and monitoring, detailed in the following case study. The interface was intended to organise digital twin data visually. R/V Gunnerus is a research vessel owned and operated by NTNU (Figure 2), measuring 36.25 metres in length, 9.90 metres in width, and 2.70 metres in depth. Gunnerus is outfitted with equipment for research across several disciplines, including biology, geology, and oceanography.



Figure 2. Starboard view of R/V Gunnerus (photo by Fredrik Skoglund)

With increasing interest in digital twins, Gunnerus is beginning to be used as a platform for research on that topic, and thus all data on the vessel were gathered in a file folder that is shared internally among employees. The folder contains 3D models, sensor logs, specifications, and results from simulations and model tests. The vessel is also equipped with an infrastructure setup for streaming sensor logs over the internet, allowing monitoring of operational states in nearly real-time. Given this context, Gunnerus became an appropriate case study to investigate how digital twin data can be organised according to data management principles.

4.2 SHIP REPRESENTATION

Two complementary Gunnerus models were created, one which would be adequate for early stage design and a detailed one for operation. The first includes a hull, two decks, three bulkheads, tanks, and compartments. The 3D visualisation in Figure 3 was generated automatically from parametric data contained in the JSON file for the vessel. At this stage, all hull form and weight distribution parameters used for simulations, including parametric estimation of lightweight, are extracted from this early stage ship model.

When creating a Gunnerus model suitable for the operation phase, the most prominent concern was gathering detailed

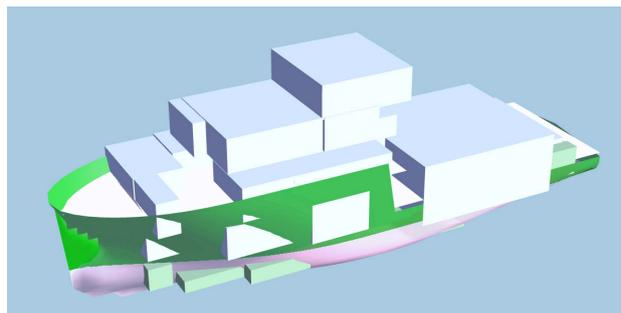


Figure 3. R/V Gunnerus model for early design, with colour code showing the vessel's design draft position (Oliveira, 2022)

3D files. A few resources were available among existing Gunnerus files that were useful for this purpose. The most important was a detailed CAD model created using a proprietary tool. The model included all vessel structures, furniture, and machinery, and was missing only wiring and piping installations. This detailed model had already been used to generate visualisation files in a few formats—Jupiter Tessellation (JT), OBJ, and glTF. The glTF file in Figure 4 was taken as starting point. Since its original purpose was simply to represent the vessel's external appearance, it did not include internal systems in the original CAD model, so it should be complemented with internal components available in the detailed file.

To provide a comprehensive Gunnerus representation on the web, it would be necessary to identify unique components inside the vessel and export them to glTF so that they can be inserted in the digital twin as base objects. Given the high number of elements in the complete Gunnerus CAD file, only internal components that can be linked to sensor logs were included, as they will be relevant to evaluating ship behaviours during operation (e.g., machinery and deck equipment). Thus, the office and research materials were not added to the digital twin at this stage. Absence of these systems might become a limitation if a researcher wanted to execute simulations that cover less-conventional aspects of ship operation, such as space utilisation and work ergonomics. Nevertheless, it remains possible to populate the model using missing objects in case such simulations are of interest in the future.



Figure 4. Gunnerus model used as basis for the ship representation (internal file at NTNU)

4.3 VESSEL-JS' SIMULATION FRAMEWORK

The ship model functions not only as a visualization and monitoring hub for ship components, but as a central element of behaviour simulations. Previous developments of the Vessel.js library provide various analyses and simulations that can be used for that purpose. We followed a progression of complexity during development of behavioural models that can be combined into applications that account for ship states at diverse update frequencies. The simplest are engineering analyses, in which a vessel's behaviours are evaluated statically. For example, data regarding hull geometry and weight distribution contained in the ship model are used to calculate the floating state of a loading condition, yielding corresponding hydrostatic and stability parameters. Results can be used as inputs to calculate other static analyses, such as resistance curves and response amplitudes to regular waves. The library can also handle discrete simulations, during which a simulation comprises a sequence of states that captures a vessel's behaviours across time. One example developed earlier is simulation of fuel consumption in transit, based on a JSON specification of the propulsion system. The last type are continuous simulations, which solve states in real-time at several Hertz. The mathematical model can be linked to dashboards with 2D and 3D visualizations that show the simulated scenario. In a design context, analyses can inform design decisions by evaluating a system according to required criteria.

and organise data according to flexible schemas, with the purpose of obtaining an extendable framework that can be applied across ship lifecycle phases. More specifically, the approach is applied to early stage ship design and to operation data with the SBS and VIS taxonomies, respectively. It comprises the following elements:

- The digital twin or product data. This work's scope focuses specifically on ship representation and states, based primarily on 3D models and shipboard logs, respectively.
- Identification tags for the data content in the platform. Every file or model must include identification tags for each taxonomy to which it should be mapped, listed in the content metadata.
- One schema for each taxonomy to be used with the digital twin. The schema is arranged in a hierarchical structure, listing identification tags associated with the respective title, description, and possible additional information. The schema must be generic; it must not be tied to a specific ship or digital twin so that it can be efficiently reused across projects.
- A hash map between each entry in the taxonomy and its respective model or file in the digital twin, if there is any. Such maps provide a quick mechanism to find which digital twin element corresponds to a given entry in the taxonomy; it avoids the necessity of repeatedly browsing through all models to find which contains the correct identification tag.

5. CASE STUDY 1: HANDLING OF MULTIPLE TAXONOMIES

5.1 GENERAL FRAMEWORK FOR HANDLING MULTIPLE DIGITAL TWIN DATA TAXONOMIES

The first case study focuses on how to handle ship representation and sensor logs according to multiple taxonomies. It proposes using existing standards to model

Figure 5 illustrates the mapping obtained with the approach. Given the primary digital twin scope, both taxonomies selected, i.e., SBS and VIS, are function oriented. The same principles could be applied with other taxonomies, be they functional (e.g., JSMEA), or based on views such as weight groups or construction zones. Each asset's data are identified according to tags, which link to respective entries in a taxonomy. Once digital twin content has been mapped this way, it can be accessed, browsed,

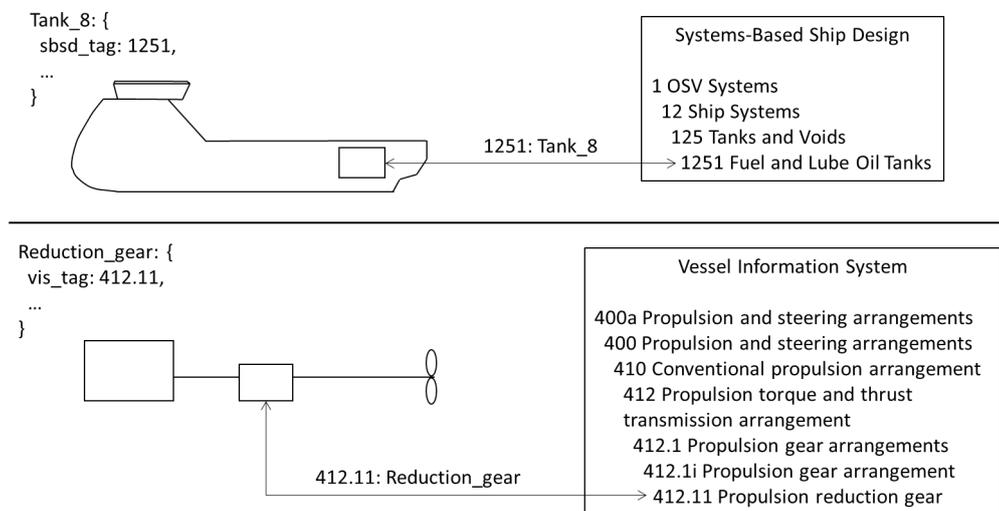


Figure 5. Use of tags and maps to identify ship systems or components according to SBS (top) and VIS (bottom)

and visualised according to the schema. This decoupled approach enables independent distribution and reuse of taxonomy schemas and allows linking of a single digital twin to multiple schemas, each suitable to a purpose. Such flexibility is useful to managing data changes by adding or removing taxonomies as the necessity arises during lifecycle phases (e.g., one for building and another for operation).

5.2 FRAMEWORK IMPLEMENTATION WITH OBJECT-ORIENTED DATA STRUCTURES

SBSD describes different types of systems depending on the type of vessel being designed (e.g., container carrier, oil tanker, or offshore support vessel). The SBSBD approach was originally derived for passenger ships (Levander, 1991); over time the approach has been applied to other commercial vessel types such as cargo tankers and offshore support vessels (Erikstad and Levander, 2012). The SBSBD taxonomy was thus adapted to R/V Gunnerus (Figure 6). VIS' Generic Product Model applies readily to cargo liners and includes a section covering special purpose equipment for offshore operations (namely, function group 340), implying that it applies to at least some offshore support vessels. The model suited the vessel in this case study well, with straightforward mapping of vessel parts to corresponding function groups and leaves in the schema.

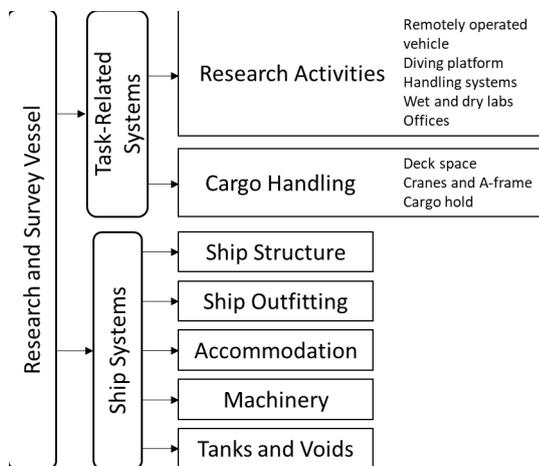


Figure 6. SBSBD taxonomy applied to R/V Gunnerus. Adapted from (Erikstad and Levander, 2012)

Mapping digital twin data to a taxonomy (VIS or SBSBD in this case) consists of the following steps:

1. Obtain the taxonomy as a JSON hierarchy.
2. Add corresponding identification tags to asset and sensor metadata.
3. Obtain a map linking digital twin data to corresponding entries in the taxonomies.

In step 1, the taxonomies are modelled as JSON files containing hierarchical structures of nested objects. The JSON file can be modelled manually or converted from

other formats (e.g., Excel spreadsheets provided by the original publisher). Once it has been obtained, it can eventually be reused in different projects. Each node in the hierarchy corresponds to an object containing at least one property (i.e., a key-value pair), with an identification code and another with a name. Optional properties might include descriptions or comments. Each sub-node in the hierarchy is modelled as a new object linked to a property of its parent node, and so on. Modelling VIS with this approach is straightforward because the Generic Product Model already links function names to corresponding numerical codes. The library has definitions attached to some functions, which can be added optionally as object properties. In the case of SBSBD, the taxonomy includes only names for existing systems, but no identification codes, so a numerical tag was assigned for each system listed (e.g., 1 for vessel systems, 1.1 for task-related systems, and 1.2 for ship systems). The following listing applies the JSON data structure to list propulsion thrusters according to VIS:

```

“433”: {
  “ID”: “433”,
  “CommonName”: “Propulsion thruster arrangement”,
  “433.1”: {
    “ID”: “433.1”,
    “CommonName”: “Propulsion thruster”,
    “Component”: “[C322] Propulsion thruster, azimuth”
  }
}
    
```

In step 2, the identification tags (“433” and “433.1”, in the listing) should be assigned to the corresponding digital twin data by the users. As Vessel.js and ISO 19848 designate properties for ID tags, the same approach can accommodate asset and sensor data. Step 2 should be taken as a shared task among project members while the product model and digital twin continuously evolve through the life cycle. This will ensure the framework supports different project stages adequately and avoid the accumulation of detached data to be identified later. In cases files are handed over from one supplier to another without consistent documentation, it would be possible to develop tools to query the taxonomy for tags with clues taken from the files’ names or contents. A user would then validate or correct the tool’s suggestions.

In step 3, a map is created to link the taxonomy’s JSON hierarchy to the tags contained in the digital twin data. Recent JavaScript versions implement a “Map” object

which is useful for that purpose. An instance of this object takes as input a collection of pairs with the taxonomy ID and digital twin data. In the propulsion arrangement example, the data would be references to the `Vessel.js` objects modelling Gunnerus's azimuth thrusters. These pairs can be created with an algorithm that scans all digital twin data for the IDs and stores these associations.

5.3 PROOF OF CONCEPT DASHBOARD

An interactive dashboard was developed as proof of the framework's key concepts. The simplified ship model in Figure 3 was linked to SBSB, which is suited to early design stage; the operational model in Figure 4 was linked to VIS, which provides greater functional detailing. For the scope of this case study, when applying VIS to this digital twin, only some of the fields were relevant, so it was unnecessary to model the entire taxonomy schema in step 1. The Gunnerus dashboard displays the hierarchical data tree and the ship visualisation. The graphic interface can be used to browse ship representation and consists of a 3D visualisation of the asset, accompanied by a tree hierarchy that corresponds to one of the taxonomies. It can display either the design or operational view, and the user can shift between both by selecting the desired alternative from a dropdown list. Thus, the interface is updated automatically with appropriate trees and visualisations. In design mode, the tree displays a SBSB hierarchy that lists the systems contained in the ship representation (Figure 7). In operation mode, nodes on the tree list the ship's components according to the VIS hierarchy (Figure 8). A user can interact with the dashboard to toggle visibility of 3D models and fetch detailed data on components. By toggling tree nodes, a user can filter the visualisation from the entire vessel, down to a system and eventually a specific component. When the mouse pointer hovers over the ship visualisation, the interface displays the component name. If users want to read more information, they can click on a component and a panel opens showing identification and position coordinates. Besides `Vessel.js` and `Three.js`, a few other libraries with specific purposes are adopted to simplify digital twin development. The lightweight widget `Tree` is used to display the chosen taxonomy as a hierarchical list linked to the digital twin visualisation. It provides interactive nodes on which a user can click to select a desired element.

A further development step would be connecting it to sensor logs. This opens the potential to obtain a comprehensive hub for monitoring of ship operation states. Three use case examples are given. The first one is visualisation of logs as raw or aggregated time series. By clicking on tree nodes that correspond to a sensor log, the interface would show the corresponding data channel list and a plot updated with the last received readings. A second use case is providing insights into health or condition of vessel components. The time series data could be filtered or aggregated into meaningful higher-level metrics such

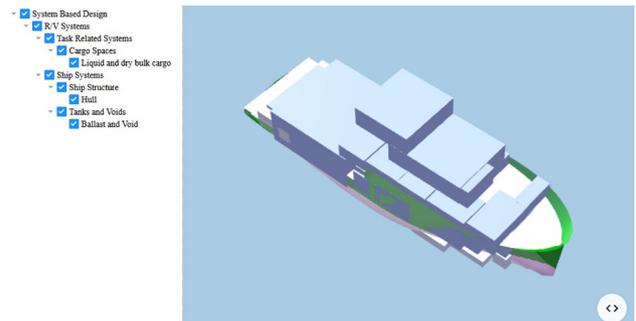


Figure 7. Gunnerus model for early design stage with SBSB taxonomy



Figure 8. Selection of specific components through the browsing tree within the operational Gunnerus model

as rolling averages of generator engines' specific fuel oil consumption and exhaust gas temperatures. Furthermore, the ISO template includes fields specifying the expected range for a measured quantity, defining thresholds for excessively high or low values. This feature could be used to configure automatic trigger warnings in case a system or sensor malfunctions and requires attention. A final use case example is exploring data content, for example in a list of sensors or data channels. This can be achieved by developing functionalities to search and query data through the user interface.

5.4 FINDINGS FROM CASE STUDY 1

5.4 (a) Support to 3D Format Standards

The findings stem mostly from complications which hindered case study development at the level of standard support and data governance. Section 4.3 mentioned that a gITF model of Gunnerus was selected as a starting point for a digital twin. In addition to the format's features themselves, one of the main reasons for this choice was the support provided to that format in `Three.js` and web-based environments. Robinet *et al.* (2018) explain that, to ensure gITF would be successfully adopted, they did not limit their work scope to publishing format specifications; instead, they also developed open source renderers to popular platforms such as `Three.js`. This ensured gITF would be easy to implement by making the codebase necessary to adopt it available beforehand for anyone interested. JT was also considered as a candidate format for development of the case study. JT is a visualisation

format developed by Siemens which is now published as an ISO standard, meaning that it can also be implemented by third parties. However, the authors have not been able to find web-driven JT renderers for Three.js or other libraries. Curiously, the Siemens NX software suite offers a functionality which exports JT models to HTML, allowing their rendering in web pages. When inspecting the source code of the HTML file generated through that functionality, it is possible to confirm it uses excerpts of the Three.js library to render the 3D model. From a legal point of view this reuse is acceptable because Three.js is licensed under an MIT license, which allows commercial use of source code without restrictions if the original authors are acknowledged (GitHub, u.d.). From the perspective of a developer external to Siemens, it would be convenient if this rendering functionality were openly available in Three.js' repository, just as it is for glTF. A secondary goal of developing a web-based Gunnerus digital twin was to assess whether the web approach would suffice its computational demands. The glTF file seen in Figure 4 amounts to 32 MB in size. Considering that it lacks internal components and that it needs to be transmitted through the internet for web use, this size is relatively large. Still, the app worked on a consumer laptop, and the concept of base and derived objects should allow efficient storage of repeated component files. glTF also supports model streaming, allowing selective digital twin loading in case there are connectivity limitations.

5.4 (b) 3D Model Layering

The case study's main limitation is that it included a small amount of internal Gunnerus components in the 3D model. This happened because, when attempting to convert the original splines-based CAD format to a tessellated format usable in a web-based environment, it was found that not only the model's structure did not map to the VIS taxonomy, but it mostly did not correspond to an engineering view of the vessel. Surfaces, solids, and lines were often grouped in arbitrary layers which did not resemble the vessel's internal components. One possible reason for such is that these layers were incidentally convenient for modelling tasks and ended up being saved to the final version of the file. From a data governance perspective, the elements in 3D models should be arranged in layers which can be traced to vessel components. This would simplify conversion of CAD models to visualisation formats and implementation of the framework linking them to the chosen vessel taxonomy.

5.4 (c) Standardisation of Sensor Data in Existing Systems

ISO 19848 provides standard data structures for both sensor logs and accompanying metadata. The latter is relevant because it contextualises data channels regarding their physical meaning and measuring units, thus simplifying interpretation of sensor logs by users. This potential benefit

is directly hindered by the prevalence of disparate metadata formats if they are available at all. Suppose an existing system's data output needs to be standardized according to ISO 19848's scope. To the extent sensor logs tend to follow similar tabular or event-driven structures, conversion of time series data is relatively straightforward. On the other hand, significant work is demanded to prepare the corresponding metadata. For each sensor stream it would be needed to identify meaning, unit, and correspondence to desired taxonomy through external sources before writing them to individual JSON files. Future maritime shipboard systems can avoid this type of re-work by implementing consistent metadata standardisation with identification according to ISO naming rules.

6. CASE STUDY 2: NAVIGATION SIMULATION

6.1 OVERVIEW OF THE NAVIGATION AND MANOEUVREING SIMULATION MODEL

The functional view that early design and operation phases share lends itself to simulation of performance, which is central to digital twins. An example is development of simulators of navigation and manoeuvring. This section presents the enhancement of the operating context in a previously developed navigation simulator by leveraging the standards presented in this work. This enhancement focuses on the objectives of streamlining the approach to modelling of the operating region's terrain and including indication of nearby vessels. Besides its potential as a training tool, the simulator also lays a foundation toward a web-based application for remote navigation monitoring, to be realised by future linking with shipboard sensor logs.

The Gunnerus manoeuvring simulation model was developed in connection with the MARKOM research project within the scope of Advanced data Analytics Framework for Ship Energy Efficiency. It was first presented as part of the second author's master thesis (Oliveira, 2021) and then in an interactive web-based notebook (Oliveira, 2022). The model is based on a linearised formulation of ordinary differential equations by Fossen (2021). It decomposes the manoeuvring simulation into two sub-problems: a predominant forward motion and a sway-yaw motion. The first sub-problem considers a quasi-static, slowly varying forward speed. In that condition, the forward resulting force is the balance between resistance and the propulsive power delivered by the propellers. In the sway-yaw motion, the direction of the stern thrusters is used to calculate the forces leading the vessel to move in the transversal direction and execute curves during navigation. For several reasons, the model's implementation is still in a preliminary form which requires further development and validation before being used for operational purposes. The hydrodynamic resistance and coefficients used in the simulation are calculated with semi-empirical formulas, which tend to not be as accurate as other prediction methods

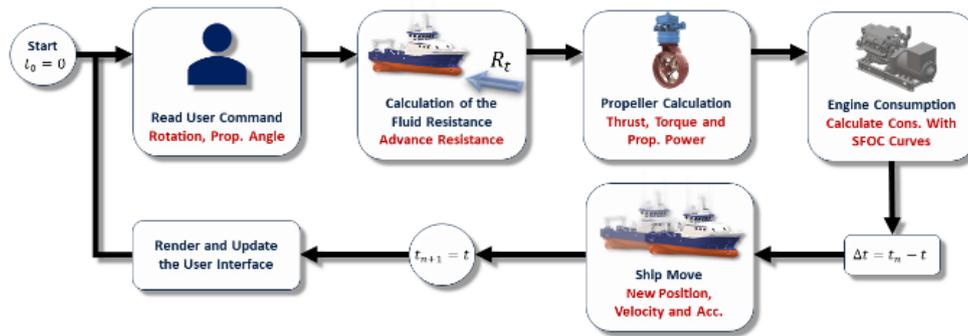


Figure 9. Schematic representation of calculations performed during one time step of the simulation (Oliveira, 2021)

such as model tests, full scale trials, or computational fluid dynamics simulations. The equations do not consider the transversal forces exerted by the bow thrusters, which should make turns and curves considerably sharper, and it does not account for environmental forces such as wind, waves, or currents either. In any case, the inclusion of this model provides a framework for manoeuvring simulations which can have its components in the Vessel.js library gradually improved. Figure 9 illustrates the algorithmic loop executed during every time step of the simulation. Once the simulation starts, it waits for azimuth angle and propeller rotation rate as user commands input through the keyboard. During every loop iteration, the simulation calculates the resultant forces acting on the hull, which is the combination of the force and moment exerted by the propulsion system and the fluid resistance acting on the hull as it moves through water. The resultant forces are input into the equations of motion to calculate the vessel's position, velocity, and acceleration at the end of the current time step. Furthermore, the brake power delivered by the engine is also calculated based on the thruster working condition and used to estimate fuel consumption during the simulation based on the specific fuel oil consumption curve.

6.2 INTEGRATION WITH OPERATING CONTEXT DATA

The existing simulator was improved in two aspects related to modelling of the vessel's operating context. The first aspect is the improved generation of terrain models for simulations in coastal areas. Oliveira, Fonseca and Gaspar (2022) attempted an approach which stored a complete 3D model of the digital twin's operating region, created with the Unity game engine for virtual reality applications, and then loaded it directly into the digital twin interface. This approach required obtaining individual 3D models for different operating regions, thus being susceptible to data availability for the desired region and requiring considerable storage space (e.g., a virtual reality model of the Trondheim city centre provided by the municipality occupies 878 MB, requiring considerable down sampling and compression to be made light enough for a web application). By using the approach described in 3.4 (a), it is possible to obtain digital elevation models for any

region within Norwegian territory and then generate 3D visualisations based on them. This process was carried to visualise the terrain in Ålesund and nearby regions. Two DEM files corresponding to that region were downloaded from the mapping authority. Each file covers a square Mercator projection of 50 km sides with a resolution of 10 meters. The files include only elevation above sea level, which is sufficient for the simulation purposes. On that note, the Norwegian Mapping authority also provides seabed depth datasets, though the authors did not go as far as assessing the possibility of using them to generate 3D visualisations. The areas contained in both files were merged and then cropped into a single square area of 50 km sides approximately centred in Ålesund. The original file was down sampled to a wireframe grid with sides of 200 units and used to generate the visualisation shown in Figure 10.

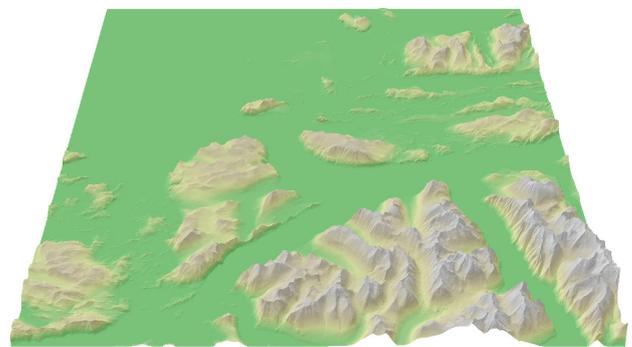


Figure 10. Ålesund visualisation generated from DEMs

The second improvement is the inclusion of surrounding vessels based on AIS data. A Python script was written to request information about the vessels in a radius of 6 nautical miles around a point of interest used as reference (taken here as the easternmost tip of the Hessa island in Ålesund). The queried information included both the vessel's states (e.g., latitude, longitude, heading, speed) and characteristics (e.g., length, beam, maximum draught). The commercial service used (namely, Datalastic) requires different queries to obtain these two groups of data. When the query for vessel states was executed, it found 113 vessels transiting in the region. A second batch of queries for was executed to obtain these vessels' characteristics and the complete data was saved to a JSON file. In the simulator's

web application, a shape comprising a combination of a cube and a pyramid geometry was created as a generic vessel visualisation. The combined shape had each of its three dimensions measuring 1 meter, so they can be scaled to visualise a given vessel by multiplying the corresponding dimension by the vessel's actual characteristic. The JSON data was retrieved in the web application and used to place and scale the generic shapes along the map, so they stand for the vessel visualisations. Since the map's area is relatively small compared to the Earth's curvature, the latitudes and longitudes stored in AIS data can be converted to cartesian positions in the 3D visualisation with a simpler equirectangular projection centred around the reference point. Figure 11 shows a close-up of Ålesund's city centre including vessel visualisations, some of which docked in proximity to each other. At this stage, the web application shows the static AIS data stored once in the JSON file. Different approaches could be taken to display vessel positions and movements in an up-to-date manner, one of which will be outlined later.

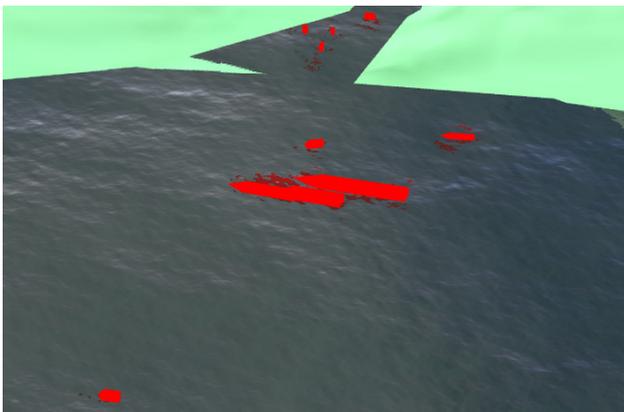


Figure 11. Visualisation of vessels docked and transiting in central Ålesund

6.3 NAVIGATION SIMULATOR

Figure 10 is a screenshot of the manoeuvring simulator user interface and shows Gunnerus navigating in the operating context modelled previously. The user can navigate the vessel by using the WASD keys on the keyboard, which control the propeller angle and rotation on the simulation. The simulation includes three checkpoint marks which are triggered when the vessel passes through them. When all three checkpoints are triggered, the simulation round concludes successfully, providing a “gamification” element to the user. The pane on the right-side displays OpenBridge interface elements, which inform the users with current simulation states and are aggregated in four clusters: Propulsion, Navigation, Others, and Checkpoints. The Propulsion cluster indicates engine load utilisation, azimuths' angle, and rotations per min. The first two parameters are displayed both as text labels and as a rounded indicator on the top-right corner which moves to represent these parameters. The Navigation cluster displays a compass, speedometer, and the vessel's cartesian positions

in relation to the map's centre. The side pane can be rolled out to display the two other clusters, containing parameters of secondary importance. The Others cluster lists current fuel consumption rate and total amount of fuel consumed during the simulation round; the Checkpoints cluster states the number of remaining checkpoints to be reached. The bottom-bar allows the user to change visualisation's camera perspective to third-person, orbital (which can be controlled with the mouse), aerial fixed position, and two different placings inside the bridge.

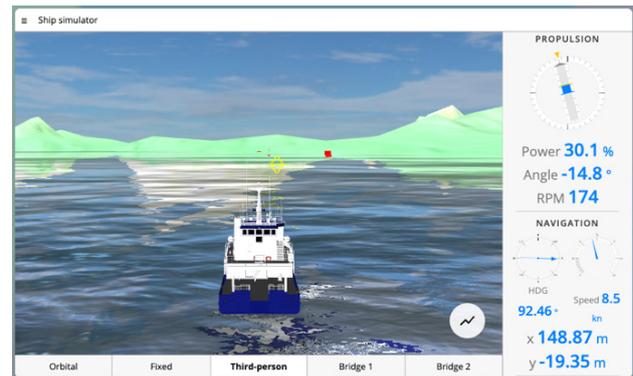


Figure 12. Manoeuvring simulation with OpenBridge indicators and yellow checkpoint mark

As commented, once the simulator's manoeuvring model has been validated, it could be used for crew training. In that regard, Oliveira, Fonseca and Gaspar (2022) illustrate the possibility of using operational logs to validate simulation accuracy by comparing its results with empirical data. When using the simulator as a training tool, the crew needs to be aware that the surrounding vessels will not behave as they would if Gunnerus was sailing near them, since the simulator plots their positions based on received AIS data instead of modelling their expected behaviour if Gunnerus was navigating in proximity. One alternative to overcome this limitation would be to have specific training setups where stretches of a real Gunnerus voyage are recorded, including positions of surrounding vessels, then the crew would train by following approximately the same path as the real vessel did. Another potential use case for the simulator is providing a dashboard for remote monitoring of Gunnerus navigation. In that case, the simulation would be linked to Gunnerus sensor logs streams to display the vessel's current states and expected trajectory. There already exists a streaming setup in which a server collects sensor logs from the vessel and passes them to clients over the Message Queuing Telemetry Transport (MQTT) internet of things protocol. When linking sensor data to the digital twin web app, web browsers do not support MQTT natively; however, open source broker alternatives exist that can convert messaging packets to WebSocket before sending them to a web client, such as the Eclipse Mosquitto. This integration would be the last step toward achieving a digital twin of Gunnerus navigation.

6.4 FINDINGS FROM CASE STUDY 2

6.4 (a) Digital Elevation Models' Suitability to Digital Twin Visualisations

Compared to the previous approach to visualisation of topography, the technique applied here reduces storage and memory needs by using lighter 2D images to generate 3D visualisations when the digital twin is executed. In web applications, where network bandwidth often acts as a constraint to data transfer speeds, this size reduction reduced the application loading time. In terms of visualisation, the new approach displays better textures and colours than the previous one, but it lacks buildings and other man-made structures, since they are not stored in the DEMs. During early stage of case study development, it was intended to obtain a level of integration where, once the current coordinates of a vessel along the Norwegian coast was obtained, it would be possible to render the surrounding region's topography. Unfortunately, the process was found to be more complicated than anticipated because the original DEM files have significant size (around 150 MB for the original model covering a region of 50 km sides) and require a fair number of conversions and pre-processes to generate the lighter image files (less than 2 MB for the same area and resolution) containing elevation profiles and textures used for visualisation. Automatic generation of 3D models is still technically feasible but requires further work. An approach would be carrying all required conversions beforehand, storing pre-built visualisation files, and retrieving them when the visualisation of a certain area is needed. This seems better than storing the original files and then converting them on demand because it is possible to avoid the complications and extra processing power needed to perform these conversions while using a little less than 10% of the original files' size in terms of storage.

6.4 (b) AIS Data Suitability to Real-Time Applications

The study used AIS data to generate a 3D visualisation of a remote operating context. The approach was found to be technically feasible, though it faces some hindrances in terms of financial costs and comprehensiveness of data. In terms of costs, the assessed service offers three subscription plans, ranging from 20,000 API requests on the cheapest one, to 80,000 requests on the intermediate one, and unlimited requests on the most expensive one. To visualise a given vessel in the simulator, it is necessary to execute two requests: one to retrieve its states and another to retrieve its characteristics. This means that to update the positions of the 113 vessels mentioned in Section 6.2 every 20 minutes for 30 days, it would be needed to execute 244,080 requests, thus requiring the most expensive plan. In terms of data comprehensiveness, AIS data covers only some of the relevant operating context aspects: it does not account for all small boats (such as recreational or fishing), buoys, or other maritime infrastructure. Even vessels which

are fitted with AIS can turn their transponders off, making them undetectable through that mean. Thus, in a remote monitoring centre, AIS data should be complemented with other sources of data such as radars, video streams, and so on. The developed simulator shows only static data stored in a JSON file. Other than incurring on the financial costs mentioned previously, visualisation of up-to-date vessel positions and movements is not technically difficult. AIS data is updated at a rate which is satisfactory for that purpose: the Class A AIS stipulated for SOLAS vessels has a transmission rate of 6 seconds for vessels sailing a steady course between 14 and 23 knots, and even higher rate for faster vessels or vessels changing course. Thus, it would be possible to set a server which requests data about a given operating region at a desired frequency and sends this data over to the web simulator through a protocol such as WebSocket. Then, the client would update the simulator's visualisation with the received data. In the intervals between receipt of updated vessel states, the client could interpolate their positions based on the most recent speed and course data.

6.4 (c) OpenBridge's Role in Simulator Usability

The inclusion of OpenBridge UI indicators, especially the rounded Propulsion element seen in the top right corner of Figure 10, greatly improved simulator's usability. In the previous version of the web application, relevant simulation parameters were displayed in the UI as text labels. This required a greater effort from the user to interpret the numerical values in physical terms. E.g., if the user read that the azimuth angle is 45°, they would need to recall position of the 0° reference line and the measuring convention (clockwise or anticlockwise) to determine the thrust direction. This turned out to be confusing, to the point the authors found it hard to cross all the checkpoints as proposed in the simulator's minigame. These problems were solved with the inclusion of graphic UI indicators, which make interpretation of simulation parameters in physical terms more immediate.

7. DISCUSSION

7.1 FRAMEWORK CONCEPT

This study shows a feasible path toward use of existing standards to classify ship data according to the desired taxonomy or naming rule. Its contribution is in proposing a framework to apply existing standards and taxonomies, while generating data structures suitable for modern database technologies. In this sense, the data model provides a standardised, yet flexible way to handle taxonomies in digital twins, making it possible to store and distribute a ship's representation and sensor logs while maintaining independence from alternative schemas. This separation makes exchange and reuse of such taxonomies more manageable. For example, if a user decides to avoid VIS and map Gunnerus to an alternative taxonomy such as

JSMEA or SFI, this could be done by importing the desired schema as a JSON file, adding necessary tags to each base and derived object in the ship model and then creating a hash map to link both data structures. Although not demonstrated in this study, this process could be simplified further by converting the entire taxonomy (e.g., the VIS schema in Excel) to JSON and then automating generation of the hash map. The framework is also extensible for allowing organisation of the same ship model according to multiple taxonomies. This means it could be applied to identify a single ship model according to weight groups, building zones, and so on. On that note, it is well known that the addition of PDM structures with the intent of making digital engineering tools more comprehensive might risk overwhelming end-users with complexity (van den Hamer and Lepoeter, 1996). Thus, framework extension needs to be guided by comparison of the added data management value brought to the ship value chain against required implementation effort and increased end-user complexity.

7.2 STANDARDS-BASED IMPLEMENTATION

The case study demonstrated the framework with a standards-based, web-driven implementation. It relied on intelligible standards with the objective of making files self-explanatory and allowing development of software based on shared assumptions regarding data formats and content. Section 4.4 examined obstacles to framework scaling in terms of data governance, support to 3D formats, and to sensor logs. While none of them is impeditive per se, they require concerted effort from different parties to ensure smooth framework adoption and implementation. This paper mostly discussed organisation of existing ship data according to the desired taxonomies. A stronger degree of standardisation would be to specify a product ontology describing all useful aspects which a digital twin could represent such as, e.g., geometry, weight distribution, and structural strength. This is because, in a digital twin, the ship model functions not only as a visualisation and monitoring hub, but a central element of behaviour simulations. E.g., data regarding hull geometry and weight distribution contained in the ship model are used to calculate the floating state of a loading condition, yielding corresponding hydrostatic and stability parameters. While there is much work to obtain a standardized vessel model under these terms, this work gives a preliminary contribution in this direction. It presents an approach to reconciling ship models suitable for concept design and operation phases by building upon previous work on the open source Vessel.js library. The work did not investigate how stakeholders would transition from the early stage to detailed perspective as the lifecycle progresses. Approaching this issue requires features that support evolution of the ship model, which brings up how versions can be managed in a digital thread. Solutions from several domains might provide insights regarding how to achieve this task, including, version control during software development, cloud-based file storage, and platforms

for collaborative document editing. Whatever measures are implemented, they must be supported by clear data governance guidelines streamlining data management through a ship's lifecycle.

8. CONCLUSION AND FURTHER WORK

This study's primary contribution is outlining the use of standards to enable management and exchange of ship data in digital twins, specially mapping of ship models and data to different taxonomies. This is admittedly restrained scope compared to the holistic ambitions of the digital twin and digital thread. The restraint is deliberate and stems from two main motivations. First, that the heightened interest on these research topics can be used as an opportunity to reengineer the foundations of data exchange in the ship industry toward more interoperability. Second, that this interest yields results which improve the state of naval architecture practice in the short to medium term. The selection of standards gravitating around a few principles: they tended to be intelligible and simple, allowing humans to read and interpret data, and they were also flexible at accommodating taxonomies to reduce prescriptions, and thus limitations, imposed on users. The paper discussed a few apps that show how standardisation helps create a streamlined ship model that serves as both conceptual design and operation. They serve as a proof of concept for the proposed standards-based approach to modelling of ship digital twin data. The approach provides a central interface to access and monitor ship representation while maintaining a holistic view of the ship models used during ship design and operation. These tools can be used to develop simulations with the purpose of informing decisions during operation. They are also a step toward creating a data loop in the industry, in which models from a previous build can be adapted to a subsequent one.

A few research initiatives exist for digital twin technology applied to Gunnerus. In the Department of Ocean Operations and Civil Engineering, where the current authors are based, initiatives are developing and applying a digital twin to various research topics, including operation monitoring, dynamic co-simulation, and nautical training (Figure 11). In that context, three tasks for future research are suggested: two immediate ones and a tentative, longer-term one. The first task is to correct and validate the simulation and control model in Case 2 based on empirical data. This would by extension provide a low-cost simulator for navigation training in an easily accessible web environment. The second task is implementing digital twin design patterns with the proposed framework. Erikstad and Bekker (2021) describe several use case templates in which digital twins could be useful (i.e., "design patterns"). It would be a significant research contribution to demonstrate how standardised data and communication protocols can be composed to develop a handful digital twin cases covering different design patterns.



Figure 13. Digital twin Gunneurs as a simulator for nautical training (District News Møre og Romsdal, NRK)

A third (and somewhat tentative) research task is using the digital twins as an enabler for a feedback loop between design and operation with the objective of refining future vessel designs. This is a tentative research topic for a few reasons. First, the time spans of design and operation are quite discrepant, so that consequences of a decision taken during design will unfold over years or decades of operations. Second, as Andrews (2018) points, the key issue in early stage design of complex vessels is the choice of style. This could be paraphrased as the choice of a core set of drivers to guide the vessel's design, based on the stakeholders' intentions for that vessel, and in detriment of all other style alternatives. Thus, while the digital twin might be a useful tool for quantifying and validating specific behaviours of vessels which have already been built, it is still uncertain how it might contribute to a broader understanding of the trade-offs and choices between different design alternatives.

9. ACKNOWLEDGEMENTS

Professors Anriette Bekker and David Andrews provided feedback on an early version of this manuscript which was included as part of the first author's doctoral thesis (Fonseca, 2022).

10. SOURCE CODE

Web applications, data and source code can be accessed on <https://vesseljs.org/>.

11. REFERENCES

1. ANDO, H. (2019) *Digitalization in the maritime industry*. ClassNK Technical Journal, No. 1.
2. ANDRADE, S. L., MONTEIRO T. G. & GASPAR, H. M. (2015) *Product life-cycle management in ship design: From concept to decommission in a virtual environment*. European Conference on Modelling and Simulation, Albena (Varna), Bulgaria, pp. 178–184. DOI: 10.7148/2015-0178.
3. ANDREWS, D. (2018) *The Sophistication of Early Stage Design for Complex Vessel*. International Journal of Maritime Engineering, Vol. 160, Special Edition 2018. DOI: 10.3940/rina.ijme.2018.SE.472.
4. CORADDU, A., ONETO, L., BALDI, F., CIPOLLINI, F., ATLAR, M., & SAVIO, S. (2019). *Data-driven ship digital twin for estimating the speed loss caused by the marine fouling*. Ocean Engineering, Vol. 186. DOI: 10.1016/j.oceaneng.2019.05.045.
5. DEFENSE ACQUISITION UNIVERSITY. (u.d.) *Glossary of Defense Acquisition Acronyms and Terms*. <https://www.dau.edu/glossary/Pages/Glossary.aspx> (Accessed 15th March 2022).
6. DNV. (u.d.) *DNV sensor naming resource page*. <https://vista.dnv.com/docs> (Accessed 15th March 2022).
7. EDIRISINGHE, R. & LONDON K. (2015) *Comparative analysis of international and national level BIM standardization efforts and BIM adoption*. 32nd Conference on Applications of IT in the Architecture, Engineering and Construction Industry, Eindhoven, Netherlands, pp. 148–158.
8. ERIKSTAD, S. O. (2017) *Merging physics, Big data analytics and simulation for the next-generation digital twins*. 11th Symposium on High-Performance Marine Vehicles, Naples, Italy.
9. ERIKSTAD, S. O. & BEKKER, A. (2021) *Design patterns for intelligent services based on digital twins*. 20th COMPIT, Mülheim, Germany, pp. 235-245.
10. ERIKSTAD, S. O. & LEVANDER, K. (2012) *System based design of offshore support vessels*. 11th International Marine Design Conference, Glasgow, United Kingdom.
11. FONSECA, Í. A. (2022) *A Standards-Based Approach to Modelling of Digital Twin Ship Data — Assessment of current standards and methods for future standardisation*. Doctoral thesis. NTNU. DOI: 11250/3035546.
12. FONSECA, Í. A. & GASPAR, H. M. (2020) *Challenges when creating a cohesive digital twin ship: a data modelling perspective*. Ship Technology Research, Vol. 9, No. 2, pp. 1–14. DOI: 10.1080/09377255.2020.1815140.
13. FONSECA, Í. A., GASPAR, H. M., DE MELLO, P. C. & SASAKI, H. A. U. (2022) *A standards-based digital twin of an experiment with a scale model ship*. Computer-Aided Design, Vol. 154. DOI: 10.1016/j.cad.2021.103191.
14. FOSSEN, T. I. (2011) *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons. DOI: 10.1002/9781119994138.
15. GASPAR, H. M. (2017) *JavaScript applied to maritime design and engineering*. 16th Computer Applications and Information Technologies in the Maritime Industry, Cardiff, Wales, pp. 253–269.
16. GASPAR, H. M. (2018) *Vessel.js: An open and collaborative ship design object-oriented library*.

- 13th International Maritime Design Conference, Espoo, Finland. DOI: 10.1201/9780429440533.
17. GASPARD, H. M. (2019) *A perspective on the past, present and future of computer-aided ship design*. 18th COMPIT, Tullamore, Ireland, pp. 485–499.
18. GIERING, J. E., & DYCK, A. (2021). *Maritime Digital Twin architecture*. *Automatisierungstechnik*, Vol 69, Issue 12, 1081–1095. DOI: 10.1515/auto-2021-0082.
19. GitHub (u.d.) Choose an open source license. URL: <https://choosealicense.com/>. (Accessed 1st January 2023).
20. HATLEDAL, L. I., CHU, Y., STYVE, A. & ZHANG, H. (2021) Vico: An entity-component-system based co-simulation framework. *Simulation Modelling Practice and Theory*, Vol. 108. DOI: 10.1016/j.simpat.2020.102243.
21. IEC. (2016) *61162-1 Maritime navigation and radiocommunication equipment and systems — Digital interfaces — Part 1: Single talker and multiple listeners*. International Electrotechnical Commission, Geneva, Switzerland.
22. ISO. (2018a) *16739-1 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*. International Organization for Standardization, Geneva, Switzerland.
23. ISO. (2018b) *19847 Ships and marine technology—Shipboard data servers to share field data at sea*. International Organization for Standardization, Geneva, Switzerland.
24. ISO. (2018c) *19848 Ships and marine technology—Standard data for shipboard machinery and equipment*. International Organization for Standardization, Geneva, Switzerland.
25. JSMEA. (u.d.) Description of Local ID definitions — jsmea-codebook. https://www.jsmea.or.jp/ssap/topics/jsmea_codebook.html (Accessed 15th March 2022).
26. NIELSEN, U. D., MOUNET, R. E., & BRODTKORB, A. H. (2021). *Tuning of transfer functions for analysis of wave-ship interactions*. *Marine Structures*, Vol. 79. DOI: 10.1016/j.marstruc.2021.103029.
27. NORDBY, K., MALLAM, S. C. & LÜTZHÖFT, M. (2019) *Open user interface architecture for digital multivendor ship bridge systems*. *WMU J Marit Affairs* 18, pp. 297–318. DOI: 10.1007/s13437-019-00168-w.
28. LEE, J. H., NAM, Y. S., KIM, Y., LIU, Y., LEE, J., & YANG, H. (2022). *Real-time digital twin for ship operation in waves*. *Ocean Engineering*, Vol. 266, Part 2. DOI: 10.1016/j.oceaneng.2022.112867.
29. LEVANDER, K. (1991) *System-Based Passenger Ship Design*. 4th International Marine Systems Design Conference (IMSDC'91). Vol. 2. Discussion about Invited Lecture (3). Kobe, Japan.
30. LIU, Y., & REN, H. (2022). *Acquisition method of evaluation stress for the digital twin model of ship monitoring structure*. *Applied Ocean Research*, Vol. 129, 103368. DOI: 10.1016/j.apor.2022.103368.
31. LÅG, S., VARTDAL, B. J. & KNUTSEN, E. K. (2015) *Position Paper 04-2015 — Ship Connectivity*. DNV GL Strategic Research & Innovation, Høvik, Norway.
32. LÅG, S., VINDØY, V. & RAMSRUD, K. (2021) *A Standardized Sensor Naming Method to Support Digital Twins and Enabling New Data Driven Applications in the Maritime Industry*. 13th Symposium on High-Performance Marine Vehicles, Tullamore, Ireland.
33. LÅG, S. & WITH, S. B. (2017) *Standardisation as an Enabler of Digitalization in the Maritime Industry*. Det Norske Veritas, Høvik, Norway.
34. NOKKALA, T., SALMELA, H. & TOIVONEN, J. (2019) *Data governance in digital platforms*. Americas' Conference on Information Systems, Cancún, Mexico.
35. NOWACKI, H. (2010) *Five decades of computer-aided ship design*. *Computer-Aided Design*, Vol. 42, No. 11, pp. 956–969. DOI: 10.1016/j.cad.2009.07.006.
36. OCX. (u.d.) *The Open Class 3D Exchange Format*. URL: <https://3docx.org/> (Accessed 15th March 2022).
37. OLIVEIRA, F. F. (2021) *An Open Web Platform Aimed at Ship Design, Simulation and Digital Twin*. Master thesis, Appendices C1 and C2. NTNU, pp. 141–149. DOI: 11250/2824236.
38. OLIVEIRA, F. F. (2022) *From Concept to Simulation — This collection is a vessel.js tutorial to teach how to create a maritime web based simulation from sketch using Vessel.js library*. URL: <https://observablehq.com/collection/@ferrari212/from-hull-to-simulation> (Accessed 1st January 2023).
39. OLIVEIRA, F. F. DE, FONSECA, Í. A. & GASPARD, H. M. (2022) *Applying Open Web Architectures Towards Collaborative Maritime Design and Simulation*. 41st International Conference on Offshore Mechanics and Arctic Engineering, Hamburg, Germany. Accepted for publication.
40. PAL, M. (2015) *Ship work breakdown structures through different ship lifecycle stages*. ICCAS, Bremen, Germany.
41. ROBINET, F., ARNAUD, R., PARISI, T. & COZZI, P. (2018) *glTF: Designing an Open-Standard Runtime Asset Format*. GPU Pro 360 Guide to 3D Engine Design. A K Peters/CRC Press, Boca Raton, Florida, pp. 243–260.

42. SANDVIK, B. (2013) *Converting terrain data to a WebGL-friendly format*. URL: <https://blog.mastermaps.com/2013/10/terrain-building-with-threejs-part-1.html> (Accessed 22nd February 2023).
43. SHAFTO, M. *et al.* (2010) *Draft modeling, simulation, information technology & processing roadmap*. Technology Area, Vol. 11, pp. 18–19.
44. SPECTEC. (2001) *SFI group system—A system for classification of technical and economic ship information*. Technical Product Description.
45. TAYLOR, N., HUMAN, C., KRUGER, K., BEKKER, A., & BASSON, A. (2020) *Comparison of Digital Twin Development in Manufacturing and Maritime Domains*. International Workshop on Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future, Valencia, Spain, pp. 158-170. DOI: 10.1007/978-3-030-27477-1_12.
46. VAN DEN HAMER, P. & LÉPOETER, K. (1996) *Managing design data: The five dimensions of CAD frameworks, configuration management, and product data management*. Proceedings of the IEEE, Vol. 84, No. 1, pp. 42–56. DOI: 10.1109/5.476025.
47. WEI, Q., LIU, Y., DONG, Y., LI, T., LI, W. (2023). *A digital twin framework for real-time ship routing considering decarbonization regulatory compliance*. Ocean Engineering.

