Doctoral theses at NTNU, 2022:345

Ícaro Aragão Fonseca

A Standards-Based Approach to Modelling of Digital Twin Ship Data

Assessment of current standards and methods for future standardisation

Doctoral thesis

Ícaro Aragão Fonseca

ISBN 978-82-326-6759-8 (printed ver.)
ISBN 978-82-326-6569-3 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral theses at NTNU, 2022:345

NTNU
Norwegian University of Science and Technology
Thesis for the degree of Philosophiae Doctor
Faculty of Engineering
Department of Ocean Operations and Civil Engineering

NTNU
Norwegian University of Science and Technology

NTNU

NTNU
Norwegian University of Science and Technology

Ícaro Aragão Fonseca

A Standards-Based Approach to Modelling of Digital Twin Ship Data

Assessment of current standards and methods for future standardisation

Thesis for the degree of Philosophiae Doctor

Ålesund, November, 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Ocean Operations and Civil Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering
Department of Ocean Operations and Civil Engineering

ISBN 978-82-326-6759-8 (printed ver.)
ISBN 978-82-326-6569-3 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral theses at NTNU, 2022:345

PRINTED IN NORWAY
NO - 1598

NORDIC SWAN ECOLABEL
Printed matter
2041 0731

Printed by Skipnes Kommunikasjon AS

Abstract

The popularisation of sensors, internet of things, and data analysis tools is bringing opportunities for several industries to improve their products and processes. One of the concepts making use of such opportunities is the digital twin, a comprehensive simulation of an engineering system that receives updated data from that system to mirror and predict its behaviour. In the ship domain, development of comprehensive digital twins finds obstacles in the lack of interoperability among digital tools and their formats. This thesis investigates use of standards to model digital twin ships with the aim of enabling data exchanges among systems, i.e., a standards-based approach.

The work begins by investigating challenges to digital twin standardisation from the perspective of previous initiatives in the ship industry, comparisons with other sectors, and emerging trends. A few drivers are identified to guide future standardisation attempts: pragmatism in scope, support to heterogeneous systems, openness, and intelligibility. Following from these drivers, the study identifies existing standards covering the domains of ship visualisation, sensor logs, and taxonomies for ship data. They are applied to develop a web-based digital twin of an experiment in a wave basin, taking advantage of the broad support offered to web standards. The case study delimits standardisation gaps in the domains of ship models, with its accompanying metadata, and simulation of ship behaviour in operation. Both of them are taken as motivation for following research stages. The first topic is addressed by extending the open source Vessel.js library, originally developed for conceptual ship design, to handle detailed vessel models suitable for digital twins. This is done with a flexible framework which allows mapping of digital twin data to existing ship taxonomies during both design and operation. The framework is applied to a case study with a research vessel, demonstrating advantages and uncovering obstacles to scaling. The second topic is partially handled through development of functionalities for simulation and visualisation of wave motion response in Vessel.js. The library is used to develop several responsive, real-time applications combining simpler models executed on the browser and results from complex analyses, executed with external software.

The work contributed to digital twin ship interoperability in two manners: by demonstrating the current state of standardisation and introducing novel methods to inform future initiatives. Given the digital twin's ambitious scope, the latter leaves open topics for further research, such as effective simulation-based decision-support during ship operations, accounting for interaction with sensor streams in real-time.

Acknowledgements

Supervisors

Prof. Henrique Gaspar for his role as main supervisor. Prof. Hans Petter Hildre for his role as secondary supervisor.

Co-authors

Felipe Ferrari de Oliveira, Dr. Daniel Prata Vieira, Sergi Escamilla i Miquel, Dr. Pedro de Mello and Humberto Sasaki for co-authoring the papers compiled in this document. Their contributions are described in the corresponding chapters.

Course lecturers

Prof. Stein Ove Erikstad and Prof. Cecilia Haskins for ministering courses which helped to outline this work.

Administrative support

Prof. Eduardo Tannuri for supporting the research stay at Tanque de Provas Numérico or Numerical Offshore Tank (TPN), University of São Paulo. Prof. Rodolfo Gonçalves for supporting a planned research stay at the Ocean Space Planning Laboratory, University of Tokyo, ultimately cancelled due to implications of the coronavirus pandemic. Prof. Karl Henning Halse for his role as administrator of the thesis assessment committee.

Contents

Figures

Tables

Code Listings

Acronyms

API Application Programming Interface. 21–24, 39, 40, 56, 87, 93

BIM Building Information Modelling. 22

CAD Computer-Aided Design. 2, 21, 39, 85, 88–91

CAE Computer-Aided Engineering. 21, 80, 87

CG Center of Gravity. 36, 37, 65, 80

CMM Capability Maturity Model. 12, 13

CSS Cascading Style Sheets. 29, 30, 88

CSV Comma Separated Values. 40, 66

DBB Design Building Block. 63

DNV Det Norske Veritas. 13, 21, 29, 32, 33

DOM Document Object Model. 30

DP Dynamic Positioning. 28, 49, 52–54

Ecma International Formerly, European Computer Manufacturers Association (ECMA). The organisation later changed its name to Ecma International. 31

ESWBS Expanded Ship Work Breakdown Structure. 32, 63

FLNG Floating Liquefied Natural Gas. xii, 72, 73, 76

FMI Functional Mock-up Interface. 89

glTF Graphics Language Transmission Format. 29, 39, 50, 57, 85, 88–91

GUI Graphical User Interface. 28, 67, 77

HTML HyperText Markup Language. 29, 30, 88

IEC International Electrotechnical Commission. 40

IFC Industry Foundation Classes. 21–23, 87

IMO International Maritime Organization. 40

ISO International Organization for Standardization. 21–23, 29, 31–33, 40, 44, 52, 53, 66, 85, 88–90, 93

IVHM Integrated Vehicle Health Management. 1

JSMEA Japan Ship Machinery and Equipment Association. 32

JSMEA-MAC Shipboard machinery and equipment data naming rule by the JSMEA. 32, 33, 42, 63, 88, 90

JSON JavaScript Object Notation. xi, xv, 17, 29, 31, 32, 35, 39, 40, 50, 52–54, 61, 85, 88, 89, 93

JT Jupiter Tessellation. 90

MQTT Message Queuing Telemetry Transport. 66

NASA National Aeronautics and Space Administration. 1, 12

NTNU Norwegian University of Science and Technology, or Norges teknisk-naturvitenskapelige universitet in Norwegian. 61, 62, 65, 66

OCX Open Class 3D Exchange. 91

PDF Portable Document Format. 29

PDM Product Data Management. 13, 94

PLM Product Life Cycle Management. 1, 3, 21, 22

PSV Platform Supply Vessel. xi, xv, 49–53, 55, 70, 71, 76

R/V Research Vessel. 59, 61

RAO Response Amplitude Operator. In this work, it is used in the context of wave motion response. xi, 53–55, 70, 73, 76–78, 92

SBSD System Based Ship Design. xi, xii, 15, 60–64, 68

SFI Ship Research Institute, or *Skipsteknisk Forskningsinstitutt* in Norwegian.. 24, 32, 63, 86, 90

STEP Standard for the Exchange of Product Model Data, an informal name for ISO 10303 Industrial automation systems and integration — Product data representation and exchange. 22, 24, 86

STL 3D format based on tessellated triangles, originally used for stereolithography. 29, 39, 88, 89, 91

SWATH Small Waterplane Area Twin Hull. 36

TPN Tanque de Provas Numérico or Numerical Offshore Tank. v, xi, 47–49, 53, 54, 70, 71

UDP User Datagram Protocol. 55, 89

VIS Vessel Information Structures. Previously called Vessel Information Systems. xi, xii, xv, 29, 32–35, 42, 52, 54, 57, 61–63, 68, 85, 88–90

W3C World Wide Web Consortium. 30

Wamit Wave Analysis Massachusetts Institute of Technology. 53, 73

WebGL Web Graphics Library. 39, 40, 88

WHATWG Web Hypertext Application Technology Working Group. 30

XML Extensible Markup Language. 21, 31, 40

Chapter 1

Preambles and background

1.1 The digital twin concept

Engineers have used simulations for decades to predict behaviours across systems' lifecycles (Boschert and Rosen 2016). Beginning in the 1960s, experts used computers to target specialised engineering problems, and since then, simulations have steadily gained prominence in engineering. Around 1985, they started gaining popularity among practitioners who used tools to address central design questions, such as those in fluid dynamics. Starting in the early 2000s, developments in range of simulation disciplines enabled a systemic approach to engineering problems by accounting for various design concerns (Bertram and Thiart 2005; Andrews and Pawling 2009). During the last decade, in addition to sustained advances to computer power, which can be used to execute increasingly sophisticated simulations, the popularisation of sensor technologies, internet of things, and ubiquitous computing is leading engineers to investigate how such resources can be employed in domestic and industrial concepts, and giving rise to concepts such as the industrial internet of things and digital twin. For instance, in 2002 Grieves and Vickers proposed a tentative concept that links Product Life Cycle Management (PLM) to data collected from the real world, providing information and decision support based on an updated account of an operating system (Grieves and Vickers 2017). In 2010, the term was articulated with its current meaning in a National Aeronautics and Space Administration (NASA) draft report (Shafto et al. 2010):

> A digital twin is an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin. The digital twin is ultra-realistic and may consider one or more important and interdependent vehicle systems, including propulsion/energy storage, avionics, life support, vehicle structure, thermal management/TPS, etc. In addition to the backbone of high-fidelity physical models, the digital twin integrates sensor data from the vehicle's on-board Integrated Vehicle Health Management (IVHM) system, maintenance history, and all avail-

able historical/fleet data obtained using data mining and text mining. The systems on board the digital twin are also capable of mitigating damage or degradation by recommending changes in mission profile to increase both the life span and the probability of mission success.

In summary, the digital twin is a comprehensive simulation of an existing system, based on live data captured during operation. Examples of specialised applications providing similar functionality appear in the literature, but the digital twin is novel due to its multi-disciplinary scope regarding simulating various system aspects using a comprehensive tool.

1.2 Digital twins elements

For succinctness, this work recognises digital twins in terms of core or defining characteristics, as Erikstad (2017) proposes. Thus, a digital twin can be characterised in terms of:

- *Identity* (in the sense of correspondence) of one or more simulation models to a physical asset;
- *Representation* of the asset in terms of digital formats, such as Computer-Aided Design (CAD) models with corresponding metadata;
- The asset's *state* based on quantifiable measurements in nearly real-time;
- The capacity to simulate the system's *behaviours* as a response to stimuli;
- Data or a model of the operating *context* that account for parameters necessary to understand the system during operation.

A few notes are added to clarify the meaning of these characteristics in a maritime context. First, the notion of identity recognizes a digital twin's purpose in aiding a system's operation. However, the four other characteristics relate to digital twin content in terms of data, and thus are the focus when assessing standardisation later. On a second note, given the digital twin's agenda of linking digital models to data collected from a physical system through the internet of things, the states in question are based much more on sensor and instrumentation readings than on condition reports produced manually. A related discussion that derives from this observation is the extent to which a parameter measured by sensors or other equipment yields a meaningful operational state without requiring further analysis. This depends on the variable being measured and the type of decision support the digital twin is intended to provide. In some cases, a single parameter can provide relevant information to a user (for example, a GPS position in a dashboard that shows distance to route destination). In others, it is used as simulation input to infer a desired state (for example, the same position evaluated over time to infer speed over ground so arrival time can be estimated). Modelling the operating context also requires attention. In principle, the digital twin does not prescribe what approach should be adopted to model the operating context. An alternative is obtaining the necessary data logs, such as metocean data, and linking them to the digital twin ad hoc. A more ambitious approach is conceiving a model of the operating context as a shared virtual space, to which several digital twins are connected. This virtual space would ideally link to weather services that provide updated meteorological measurements, and estimations and predictions across operating regions.

1.3 The digital twin's role through the lifecycle

The novelty of the digital twin as a research topic allied to its ambitious aims imply on a lack of consolidation regarding use of associated concepts. In addition, it is meddled with similar terms such as the digital thread or the digital ship product model, the latter of which has been discussed in computer-aided design and engineering literature for decades (Whitfield, Duffy and Meehan 2003). Different authors might advance disparate understandings of these terms, or use them interchangeably. This work adopts definitions published in a glossary from the Defense Acquisition University (i.e., military) to settle confusions (Defense Acquisition University 2022). The most important digital twin characteristic is its ability to link to the operational state of a physical asset. Consequentially, in terms of lifecycle stages the digital twin focuses on phases after construction, such as commissioning and operations. Alternatively, stakeholders of a system might decide to implement a comprehensive data strategy that begins during earlier lifecycle stages, leading later to a functional digital twin. This approach is called a digital thread. During earlier design, stakeholders would create a digital system model that would be maintained during subsequent lifecycle phases to represent the most updated version of a vessel, up to construction. This system model relates closely to product models in traditional PLM disciplines, and to the system representation contained in the digital twin. It might also link to simulations and services thorough the lifecycle, for example: the vessel's analyses developed for use during design and operation, a simulation of construction in a shipyard (Taylor et al. 2020), or software that allows classification society surveyors to review and approve vessel parts during design and commissioning (Astrup and Cabos 2017). Thus, the digital thread represents the data management and governance framework that enable effective use of all digital tools involved in an asset's lifecycle, including the digital twin. It should account for the interplay among data, software, and other digital systems to help decision-makers extract actionable information from digital resources.

1.4 Implications to the maritime sector

It is too early to assess the extent to which the digital twin will influence existing value chains in engineering and industry. This uncertainty is, to a great extent, common among novel technologies, including, and especially, those related to computer machinery. In an interview to ben-Aaron (1985), Weizenbaum claimed that nearly since its beginning, the computer was been a solution looking for a problem. Conversely, Simon (1987) argues that an exploratory approach is required to identify uses for a novel, potentially revolutionary, technology. An illustrative parallel is traced with Google's first months after funding, during which it struggled to find a feasible business model to sustain operations (Zuboff 2019). At the time it already collected all kinds of data and metadata as by-products of user interactions, but the data were not directed to significant ends. As business pressures intensified, the company turned to the previously unused data and started using them to maximise the chances of advertising success for users based on their interactions with Google services and other tracked websites. The success of this business model consolidated the perception of data as a source of value, intensifying trends such as the industrial internet of things and the digital twin. In this sense, the digital twin's current development phase is experiencing many of the uncertainties described in Google's early days. There is now much data collected in ships and other industrial installations, without a specific purpose, or just in case, echoing the tensions between the need to have a feasible business model and the need to explore data to discover what type of value they can yield (Nokkala, Salmela and Toivonen 2019).

Thus, it is necessary to emphasise a few opportunities brought about by digital twin ships. The most immediate is the possibility for naval architects to gather understanding about a vessel's operating condition. The digital twin should allow for consistent follow-ups during operation by aggregating the data and models required to study performance and behaviours of a vessel in its real operating context. For instance, it might be used to identify speed loss and indicate the need for maintenance interventions (Coraddu et al. 2019). Similarly, it is possible to track structural vibration modes for a vessel based on full-scale data as a first step toward accurate assessment of fatigue life (Zijl et al. 2021). As environmental regulations come into force to reduce CO_2 emissions from shipping, digital twins can provide decision-support to adjust a vessel's operational profile (Zaman et al. 2017), such as defining its chartered speed, or to evaluate the efficacy of a retrofit proposal in attaining compliance with future regulatory requirements. In the middle term, the increased access to full-scale data against which simulations can be validated will hopefully lead to an increase in robustness of such simulations, giving ship designers better tools to design future vessels and plan complex operations. Longer-term opportunities are using the digital twin for remote ship operation while accounting, in case of commercial fleets, for synchronization with broader transport chains. In this context, it might act as an informational hub for a crew that is supervising and controlling a vessel from a shore control centre, which might be an early step toward ship autonomy (Ando 2019).

Chapter 2

Research scope

2.1 Motivation

An additional challenge that digital twins experience in the maritime sector is a lack of interoperability among digital systems, hindering, among others, data exchanges among stakeholders. Such preoccupation might appear premature, given the discussion about how digital twins are only starting to be integrated into maritime practice, but it is argued that this is not the case for two reasons. First, Szykman et al. (2001) assert that addressing interoperability problems in anticipation of next-generation systems increases the probability that common solutions are adopted, in comparison to developing them in response to failings of systems developed previously. Second, software and data have been historically fragmented throughout all phases of the ship lifecycle (Gaspar 2018a). This happens for various reasons. Due to the tender-based nature of ship design, each vessel is approached as an individual project also from the digital perspective, whereas mass-produced systems, such as cars and airplanes, rely on a well-consolidated digital documentation in order to be effectively manufactured and supported for years, possibly decades to come. The interplay between a ship's designer and builder might also reduce data traceability compared to industrial sectors where the same company designs and builds (or at least integrates) the system to be delivered. With these reasons held constant, data fragmentation will continue in the future, making concerted standardisation efforts necessary to realise cohesive digital twin ships. The adoption of standards should establish a level playing field among competing software providers. In the long-term, it might prevent actors in the shipping sector from being inadvertently locked into proprietary formats and software suites, and allow market space for those who offer novel digital solutions to compete with established players (Låg and With 2017).

2.2 General objectives

This thesis' general objective is to contribute to a standards-based approach to data modelling and management in digital twin ships. Drawing from the terminology introduced in Chapter 1, digital twins focus primarily on the simulation of assets that have already been built and are operational. If put into a broader perspective, they might be understood as part of an overarching digital thread which starts during early design and later leads to consolidation of a digital twin during construction and operation. Data standardisation in the context of the digital thread is a secondary focus of this thesis. When developing the modelling approach discussed in the following sections, the adoption of existing standards is prioritised to the extent that this is feasible, i.e., a standards-based approach. As the study progresses and identifies limitations of existing alternatives at covering the digital twin's scope, it extended the selection of standards to propose novel data models designed to address identified challenges. These data models are designed to follow some formal consistencies to contribute to the discussion on standardisation, even if they do not qualify as attested standards.

2.3 Specific objectives

The outlining of a standards-based approach to digital twin data, as this thesis' general objective, can be decomposed in three specific ones. The first is to identify the challenges to digital twin standardisation and pinpoint drivers to overcome such challenges. The second is to find the extent to which existing standards covers digital twin content. The third is to devise preliminary methods for standardisation of remaining digital twin contents and evaluate their effectiveness against the identified drivers.

2.4 Research questions

The specific objectives are encapsulated in three pairs of research questions, which are revisited in Chapter 8. Furthermore, after the state of art is put into perspective, Section 3.12 explains the transition from literature research to development of case studies, i.e., from the first pair of research questions to the two following ones.

1. On the challenges to standardisation of digital twin data and paths to future standardisation:
   a. What are the challenges to standardisation of digital twin ship data?
   b. What drivers need to be considered for successful data standardisation in digital twin ships?
2. On the application of existing standards to digital twin ships:
   a. To what extent do existing standards cover digital twin ship data?
   b. Which of the data types contained in the digital twin still lack standardisation?

3. On suitable methods for further digital twin standardisation:
    a. What methods can be proposed to model remaining digital twin content, based on selected standardisation drivers?
    b. What implications do such methods bear?

2.5 Research development

2.5.1 Domain perspective

The research work was published as a collection of papers, including a literature review and case studies developed during the PhD period. Thus, this document is divided in two sections. The first consists of a report summarising the thesis's methods, and results, then discussing how they relate to the overall research agenda. The second is a compilation of papers and one draft, which are listed in Section 2.5.3. The report chapters are arranged according to the different domains of digital twin content (Figure 2.1). Chapter 3 presents challenges and drivers to cohesive modelling of digital twin data, taken from the literature review in Paper I. Chapter 4 does not relate directly to any paper, but provides an overview of the modelling approach used throughout the study. Chapter 5 assesses the extent to which existing standards can be used to develop a digital twin ship using the proposed modelling approach. It applies the approach to a case study with a scale model in a wave basin and relates closely to Paper II. The two following chapters present application and development of standards for specific content domains in digital twins. Chapter 6 focuses more deeply on standardisation of ship representation and data taxonomies to organise them according to requirements across lifecycle stages and is based on draft Paper III. The case study applies the framework to a digital representation of a research vessel in two different stages: early design and operation. Chapter 7 develops how the asset representation can be used to simulate vessel behaviour. It relates directly to Paper IV, and tangentially to Paper V. As a limitation of the case studies developed with this approach, they have not been linked to sensor readings collected from ship operations. Thus, they did not meet one characteristic to qualify as digital twins, and are therefore better described as plain simulations. Chapter 7 is depicted with dashed lines in Figure 2.1 for this reason. Chapter 8 addresses the research questions based on the literature review and case studies. Chapter 9 summarises concluding remarks and suggests future research topics.

2.5.2 Chronological perspective

A chronological perspective of published papers throws further light upon research development and the limitation mentioned (Figure 2.2). The first publications written as part of this thesis work were Papers IV and V. They presented case studies which had been developed as a continuation of previous research into simulation of ship design and operation (Fonseca and Gaspar 2015; Fonseca 2018). The literature review in Paper I evidenced two digital twin aspects which had not been considered by previous papers regarding standardisation: the liking of behavioural models to sensor observations

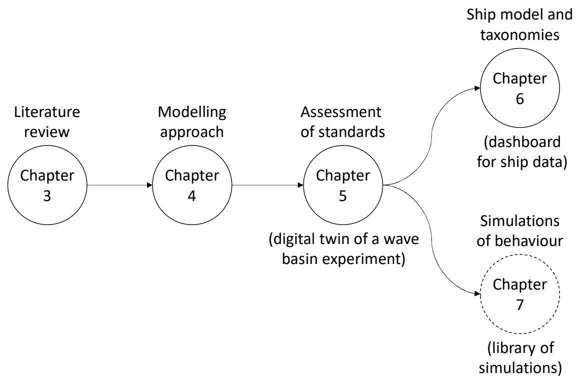


Figure 2.1: Thesis structure in terms of digital twin scope.

and the relevance of building consensus around standardisation alternatives to making them viable. The following publications approached these two questions explicitly, applied to a simplified digital twin case study (Paper II) and to a detailed ship representation suitable for use in the digital twin and digital thread (Paper III). This timeline brings consequences and limitations to the applicability of findings from Chapter 7 to digital twins. From one side, the simulations presented there already offer some of the characteristics useful for digital twins, specifically the ability to compose different mathematical models and to configure update intervals while achieving responsive execution suitable for real-time use. From the other, they lack the integration with sensor data, which should be accounted both from the perspective of network connectivity and of usefulness during ship operations to enable successful digital twins. Chapter 9

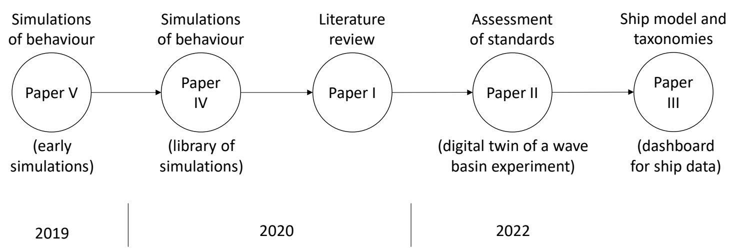


Figure 2.2: Publication sequence of papers appended to thesis compendium.

2.5.3 List of papers

The journals and conference mentioned in this list are indexed at the Norwegian Register for Scientific Journals (NSD).

Published in or submitted to journals:

I Ícaro Aragão Fonseca and Henrique Murilo Gaspar (Sept. 2020). 'Challenges when creating a cohesive digital twin ship: a data modelling perspective'. In: *Ship Technology Research* 68.2, pp. 70–83. DOI: 10.1080/09377255.2020.1815140.

II Ícaro Aragão Fonseca, Henrique Murilo Gaspar, Pedro Cardozo de Mello and Humberto Akira Uehara Sasaki (Jan. 2022). 'A Standards-Based Digital Twin of an Experiment with a Scale Model Ship'. In: *Computer-Aided Design* 145. DOI: 10.1016/j.cad.2021.103191.

III Ícaro A. Fonseca and Henrique M. Gaspar (2022). 'An Open Framework For Data Taxonomies In Digital Twin Ships'. In: *Submitted to International Journal of Maritime Engineering; re-submission being prepared after first round of reviewers' comments.*

IV Sergi Escamilla i Miquel, Ícaro Aragão Fonseca, Henrique Murilo Gaspar and Daniel Prata Vieira (July 2020). 'An open-source library for hydrodynamic simulation of marine structures'. In: *Marine Systems & Ocean Technology* 15.3, pp. 160–174. DOI: 10.1007/s40868-020-00083-3.

Published in conference proceedings:

V Ícaro A. Fonseca, Felipe F. de Oliveira and Henrique M. Gaspar (June 2019). 'Virtual Prototyping and Simulation of Multibody Marine Operations Using Web-Based Technologies'. In: *Proceedings of the 38th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Glasgow, UK: American Society of Mechanical Engineers. DOI: 10.1115/omae2019-96051.

Chapter 3

The digital twin concept in the maritime sector

3.1 Chapter's aim

The literature review elucidates challenges to standardisation of data in digital twin ships, framing it according to three topics. First is application of the digital twin definition to a ship context, second is identifying the core aspects to be modelled in digital twin ships, and third is reviewing standardisation attempts of engineering data to extract drivers that can guide standardisation. This chapter and its accompanying paper address the first pair of research questions.

3.1.1 Paper I

Ícaro Aragão Fonseca and Henrique Murilo Gaspar (Sept. 2020). 'Challenges when creating a cohesive digital twin ship: a data modelling perspective'. In: *Ship Technology Research* 68.2, pp. 70–83. DOI: 10.1080/09377255.2020.1815140.

Abstract: A digital twin is a digital asset that simulates the behaviours of a physical counterpart. Digital twin ship literature identifies that the concept is already being applied to specialised problems, but no clear guide exists for creating broader interdisciplinary digital twins. Relevant dimensions of product data modelling and previous attempts at standardizing ship data elucidate the requirements for effective data modelling in a digital twin context. Such requirements are placed in a broader perspective for digital twin implementation that encompasses challenges and directions for future development of services, networks, and software. Finally, an open standardization for digital twin data is proposed based on lessons extracted from this panorama, proposing its application to a research vessel.

Declaration of co-authorship: I researched the literature and wrote the paper. Henrique Gaspar advised on general development and suggested improvements to various sections.

3.2 Progression of complexity

Since creation of the digital twin concept is recent, it is not yet a mature field of research, specially considering maritime literature. The research literature reflects this state, with various publications, including the current one, proposing implementation approaches and discussing challenges and opportunities to developing digital twins of various systems in general (Fuller et al. 2020; Qi et al. 2021) and maritime digital twins in particular (Bekker et al. 2018). In recent maritime research, existing digital twin implementations are either specialised in scope or in an early stage of development. Still, initial results suggest that digital twins are useful for addressing challenges in personnel training, system integration, and condition monitoring. Thus, it is necessary to work toward piece-wise development of digital twins and digital services in the maritime sector. One notion is gradual development of digital twins toward higher levels of complexity and sophistication. Soon after NASA proposed its ambitious digital twin definition (Section 1.1), researchers questioned the feasibility of achieving such a concept using existing technical and financial resources (West and Blackburn 2017). Such reactions warned of the risk of overestimating the capabilities and degree of fidelity to be expected from a digital twin. An alternative to such a monolithic digital twin system that accounts for every behaviour of interest would be a collection, or hub, of several digital models that account for various behavioural aspects of the system, each attaining a degree of accuracy and detail that is suitable for its designated purpose (West and Blackburn 2018).

Erikstad (2019b) draws from the Capability Maturity Model (CMM) to assess the progression of digital service capability, from lower to higher levels of maturity; Erikstad and Bekker (2021) later elaborate on the same framework. CMM was originally proposed by Humphrey (1988) to capture the degree of software development streamlining using five levels: initial, repeatable, defined, managed, and optimised. As an organisation advances through these capability levels, it is able to establish measurable development processes that can be controlled and improved more systematically. Erikstad adapted CMM to a qualitative index of digital service maturity, describing five levels of functionality that the service might provide — observe, measure, model, predict, and decide. Observe means simple registration and presentation of sensor observations, such as in monitoring dashboards. Measure establishes a comprehensive vessel state through the combination of observations and data analytics. Model links the established vessel state to observations of external conditions to obtain understanding about the vessel's operating behaviour and performance. The following capability level builds upon modelling of vessel behavior to Predict consequences of alternative control actions. The final level provides complete insight into relevant vessel states and behavior in association to its operating context, being able to Decide the quantitative outcome of every relevant operational action taken. Moving up these levels leads to higher value, but it also increases costs, thus requiring a compromise solution from a ship owner or system developer. As technology advances and the cost of hardware components and infrastructure decreases, the balance point to this solution shifts, making profitable services that were previously deemed too expensive.

Figure 3.1 summarises this framework, showing the digital twin elements and suggesting applications to various ship operation domains. The figure illustrates the digital twin as a combination of an asset representation and simulations of behavior connecting to live data streamed from the vessel and its operating environment (Section 1.2). The rows in the central table depict potential digital twin domains, while the columns outline their levels of sophistication. For succinctness, the figure aggregates the first three levels of maturity under a broader "monitor" category. This is justified because they all approach vessel performance reactively, rather than proactively. In addition to providing an overview of potential applications of future digital twins, Figure 3.1 reiterates that future digital twins do not need to take an "everything all at once" approach. From a managerial perspective, these digital twins might attain disparate functionality levels through various domains depending on the ship operator's priorities and on the financial feasibility of existing technical solutions. From a development perspective, they might function as a composition of loosely coupled (i.e., federated) modules, instead of a monolithic system. It is also believed that this arrangement would make the overall digital twin more resilient to failures, as a localised downtime in one of the domain-specific modules would not necessarily interfere with the others.

During 2020, the recommended practice RP-A204[1] by classification society Det Norske Veritas (DNV) expanded and consolidated these ideas across several concepts (DNV 2020). It used CMM in its original form, without adaptations to measure the capacity of organisations that were developing and supporting digital twins. Following the idea of a digital twin as a hub of digital services, the document introduces a "functional element," which corresponds roughly to each service or behavioural model included in a digital twin. The document introduces a second classification of capability across five levels — standalone, descriptive, diagnostic, predictive, prescriptive, and autonomous. For example, a functional element is standalone when it is not yet linked to a data stream from the asset, and an autonomous functional element closes the control loop without dependence on a human user, or at most, relying on human supervision of its performance. The document presents a concept similar to asset representation under the name "asset information model". DNV classifies the model based on the sophistication of its structure, from detached, to basic, integrated, and finally holistic.

3.3 Product data management applied to digital twins

Since it has been argued that a digital twin's asset representation closely resembles a product model, evaluating which Product Data Management (PDM) principles can be adopted for effective management of this data content is warranted. Hamer and Lepoeter (1996) describe core concerns of PDM using five necessary and sufficient dimensions — views, versions, hierarchy, status, and variants — summarised in Figure 3.2. The dimensions of views and hierarchy are considered central to describing the

[1] Namely: Qualification and assurance of digital twins.

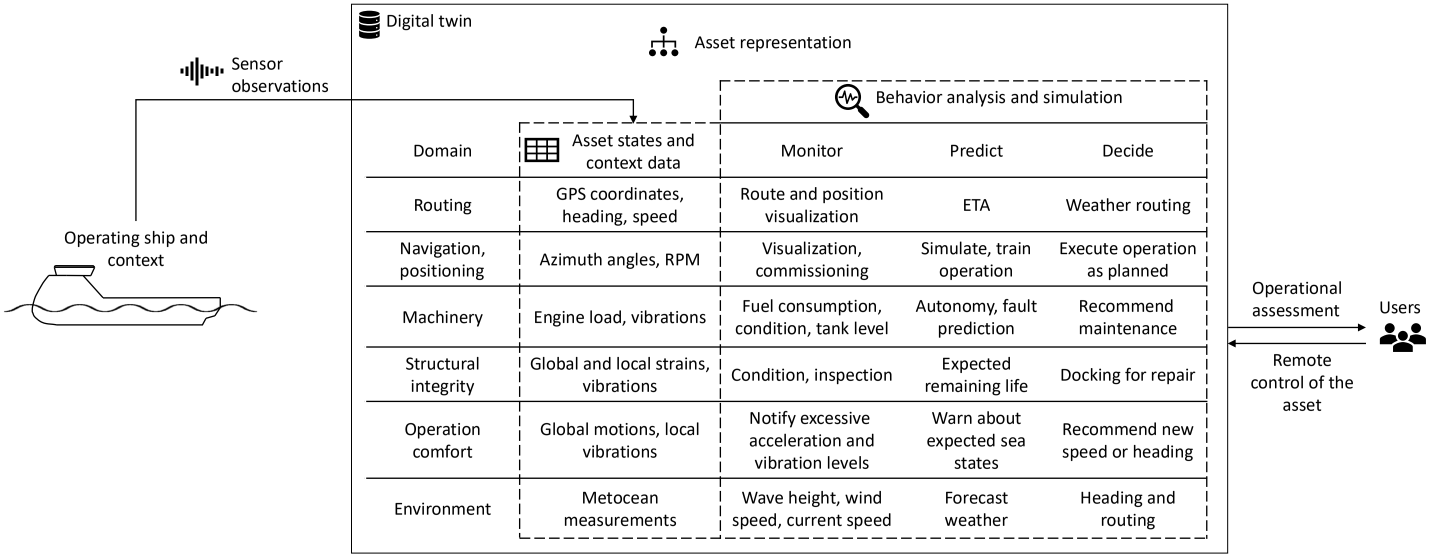


Figure 3.1: Digital twin elements and potential applications at different maturity levels (Fonseca et al. 2022), updated from (Fonseca and Gaspar 2020).

representation of a ship model by a digital twin or other information systems. Ample discussions exist in the literature regarding the role taxonomies play in managing ship data by listing data hierarchically according to a domain's view. For example, it is useful to have a view which organises the vessel in systems that can be mapped to functional roles to ensure it is capable of fulfilling the mission to which it is intended. During early design, this view should have an architectural emphasis on ship geometry and spaces to support assessment of stability and resistance, such as in the System Based Ship

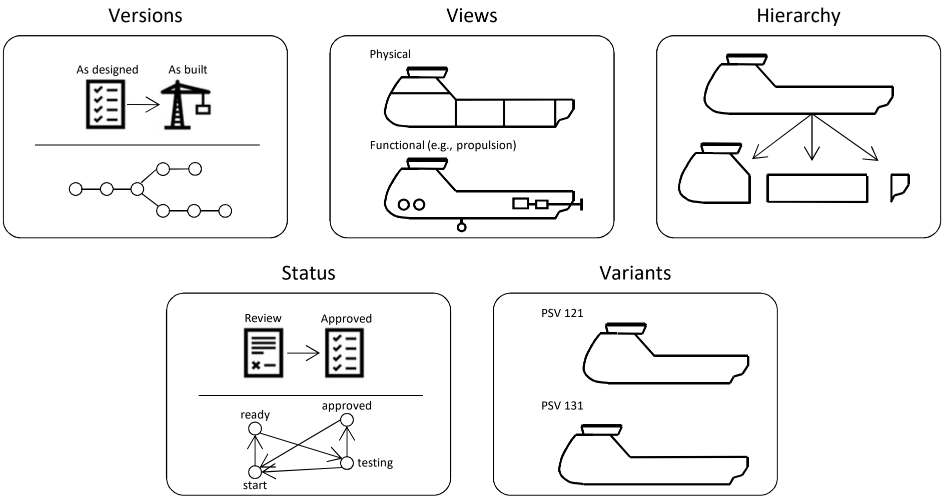


Figure 3.2: Five dimensions of product data management applied to digital ship model. Adapted from (Hamer and Lepoeter 1996).

The dimension of variants might also play a supporting role in the taxonomy's schema. Van den Hamer and Lepoeter characterised this dimension as handling alternative products in a family or manufacturing platform, such as those explained by Erikstad (2019a). This would, in principle, suggest it has limited applicability to the ship domain, since product families are not widespread in the sector. In practice, however, a ship's taxonomy must list entries for contingent alternatives at subsystem and component levels (for example, whether the propulsion system is diesel or diesel-electrical, or whether it has a fixed propeller and rudder, or azimuth thrusters). The dimensions of versions and status are essential to managing model modification and asserting data validity over time in the digital thread. In that context, the versions describe the various iterations that the system embodies spanning from early design, to detailing, construction, operation, and decommissioning. This includes, e.g, design iterations and modifications from the detailing project to the system as built. The dimension of status tracks such versions in terms of class approval, release to production, or any other label that identifies the validity and up-to-dateness of information. In the digital twin, the dimensions of version and status can represent the evolution of the vessel's condition during its operational life span with the goal of compiling information necessary for effective vessel support and ensuring it accurately fulfils its decision aiding role during operation. Thus, if for example a vessel has accumulated significant structural fatigue leading to strength reduction, it might be needed to update the corresponding digital twin model accordingly and reissue it under a new version number.

3.4 Simulation model typologies

The framework in Figure 3.3 organises the digital models of behaviour in relation to their capability of modelling time variation and sensor input. A typical engineering analysis is a static evaluation of ship behaviours, with the purpose of evaluating its performance in a representative load situation. A simulation adds time variation to the behavioural model, thus accounting for a load time series and allowing users to evaluate performance of a ship in a time-dependent context, such as a prospective operation. The digital-twin adds sensor inputs and near real-time synchronisation to the simulation models, allowing its use for decision support during operation.

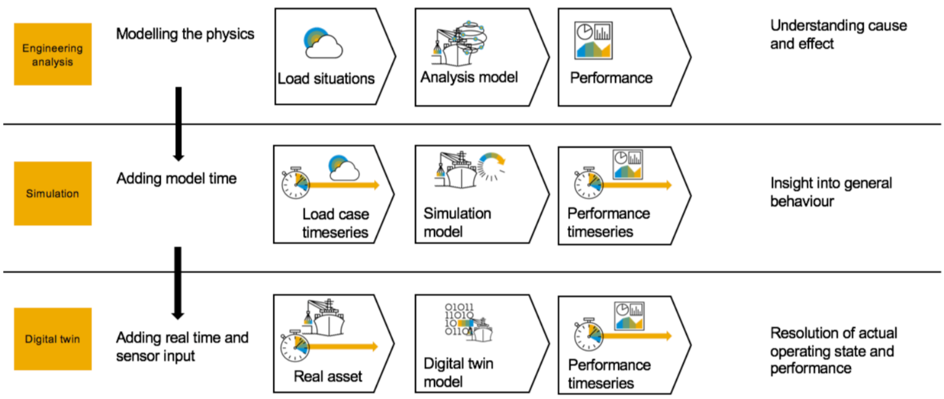


Figure 3.3: Progression from static models of behaviour to digital twins (Erikstad 2017).

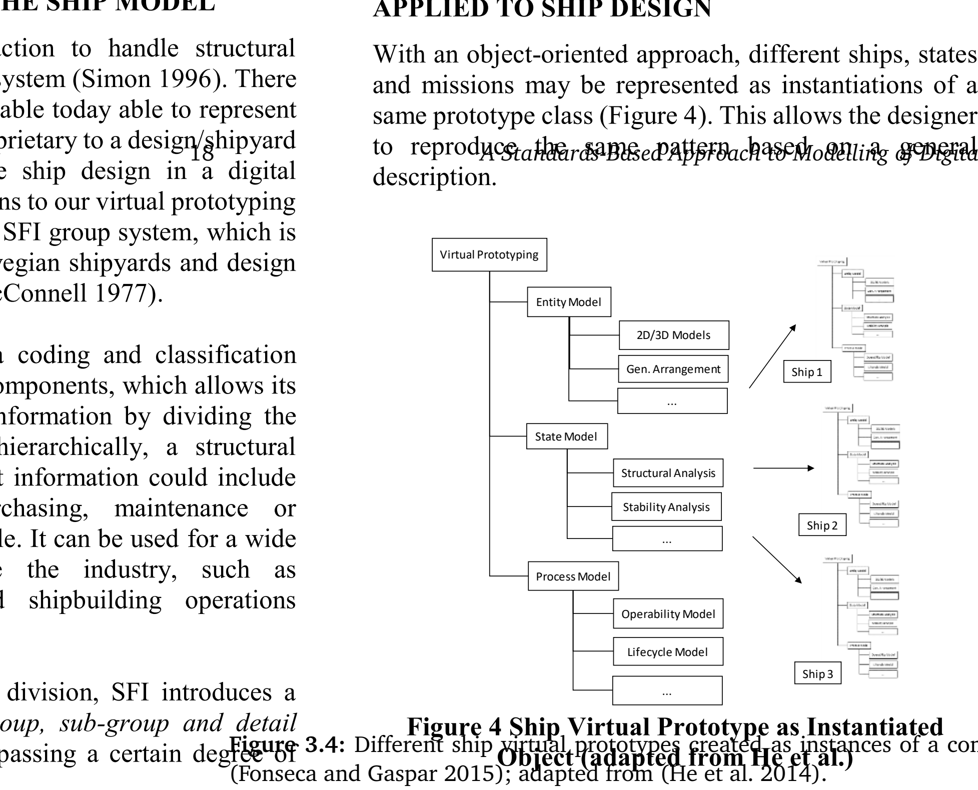
The digital twin mirrors asset operations as responsively as possible, but in practice, it might be necessary to balance latency and update rates of data streams, depending on what is possible and desirable within the existing communication infrastructure. For example, if a ship owner wants to install a digital service to monitor propeller shaft conditions based on vibrations in real-time, it is necessary to create a continuous behavioural model that updates several times per second. A second case is a ship owner who wants to ensure that a vessel is sailing with optimum pitch to minimise fuel consumption. The control module is then triggered only when there is a significant change to the vessel's loading condition (i.e., a discrete event), and then will adjust ballast water to reach the desired buoyancy condition. Even then, it might be unnecessary to retrieve vessel states as frequently as when monitoring vibrations, since changes to draught and pitch occur over a much longer time span. A digital twin should be able to accommodate these various types of behavioural models and time treatments.

3.5 Basic concepts of object-oriented programming

Object-oriented programming is based on a few concepts, among which: encapsulation, inheritance, composition and polymorphism (W3Schools 2022). An object is a data construct that can store properties (i.e., name-value pairs) and methods (i.e., functions) that can manipulate these stored properties and other inputs (Fonseca and Gaspar 2015). This allows grouping of data in a self-contained (i.e., encapsulated) structure in the source code; for example, the JavaScript Object Notation (JSON) to be presented in Section 4.4 allows serialisation and exchange of the properties (but not of the methods) contained in a JavaScript object. The mechanisms of inheritance and composition allow reuse of data structures when creating objects. In the first mechanism, objects inherit properties and methods from a generic "template" that outlines the object's structure with placeholders to be filled with the object's properties. The same blueprint can be used to create several instance objects (Figure 3.4). This process is called instantiation, and is analogous to an "is a" relationship among concepts, such as a bulk carrier being a type of vessel. JavaScript's object-orientation is prototype-based, meaning that objects inherit properties and methods from another object, i.e., the prototype. The composition mechanism gives a flexible tool for building objects without stating an inheritance relationship. It allows assignment of properties and methods to existing objects ad hoc, such as combination of various objects inside an overarching one. In that sense, it is analogous to a "has a" relationship, such as a bulk carrier having a certain number of auxiliary engines. Finally, polymorphism indicates that a certain routine encapsulated in an object can behave differently depending on the object within which it is operating. Thus, two different objects might have methods for calculating fuel consumption based on the distance travelled by a vessel, for example, but this consumption would be calculated differently depending on the number and model of engines in a vessel's propulsion system. Due to its supporting concepts, the object-oriented approach has been popular in the implementation of formal ontologies, i.e., data-models intended to model information about the real world, including representation of engineering products for design, simulation, and archival purposes (for example, as seen in (Coyne et al. 1990)).

3.6 Handling of digital twin states

He et al. (2014) present an object-oriented resource management tool for virtual prototyping in collaborative product design. The candidate has applied the approach to development of maritime simulations in a few studies previous to this thesis. The earliest was a proposal regarding how to adapt the approach to conceptual ship design (Fig-


HE SHIP MODEL

ction to handle structural ystem (Simon 1996). There able today able to represent prietary to a design/shipyard e ship design in a digital ns to our virtual prototyping SFI group system, which is egian shipyards and design cConnell 1977).

coding and classification omponents, which allows its nformation by dividing the hierarchically, a structural t information could include rchasing, maintenance or le. It can be used for a wide the industry, such as d shipbuilding operations

division, SFI introduces a *oup, sub-group and detail* passing a certain degree of

APPLIED TO SHIP DESIGN

With an object-oriented approach, different ships, states and missions may be represented as instantiations of a same prototype class (Figure 4). This allows the designer to reproduce the same pattern based on a general description.

Figure 4 Ship Virtual Prototype as Instantiated Object (adapted from He et al.)

Figure 3.4: Different ship virtual prototypes created as instances of a common object (Fonseca and Gaspar 2015); adapted from (He et al. 2014).

ure 3.5). Later, it was applied to simulations of ship design and maritime operations, with emphasis on propulsion and fuel consumption during transit (Fonseca 2018). The resource management tool organises the simulation into three linked constructs — entity, state, and process models. The entity model (or “Ship” in Figure 3.5) is defined as a product model established according to the design task and, in a mechanical engineering context, it includes 2D and 3D mechanical models, control models, hydraulic models, and so on. It constitutes the foundation of the virtual prototype because it defines the product’s structure and geometry, which are used as basis to execute relevant simulations. In the context of digital twin ships, the entity model is analogous to the asset model and can be taken for a comprehensive 3D model of the vessel, accounting for hull geometry, spatial arrangement, tanks, compartments, structural scantling, machinery, outfitting and so on. The 3D model ought to be complemented with metadata about design specifications, weight distribution, materials, or other relevant characteristics.

The state model evaluates the entity model subject to external, static constraints. He et al. provide as examples the evaluation of a load state in a finite element analysis and of a given work position in a kinematics simulation. In general, the state model is analogous to an “analysis” in Figure 3.3; in the context of ship simulations, it can be exemplified by analyses of static stability, resistance at a given speed, or structural strength under static finite element analysis. Finally, the process model (or “Mission” in Figure 3.5) is analogous to a simulation in Figure 3.3, representing the evolution of a certain product behaviour over time. He et al. posit the process model is obtained through accumulation of successive state models capturing each time instant in a simulation, and vice-versa through decomposition. Here it is argued this correspondence is imprecise; nevertheless, it is still useful to have a concept which captures dynamic

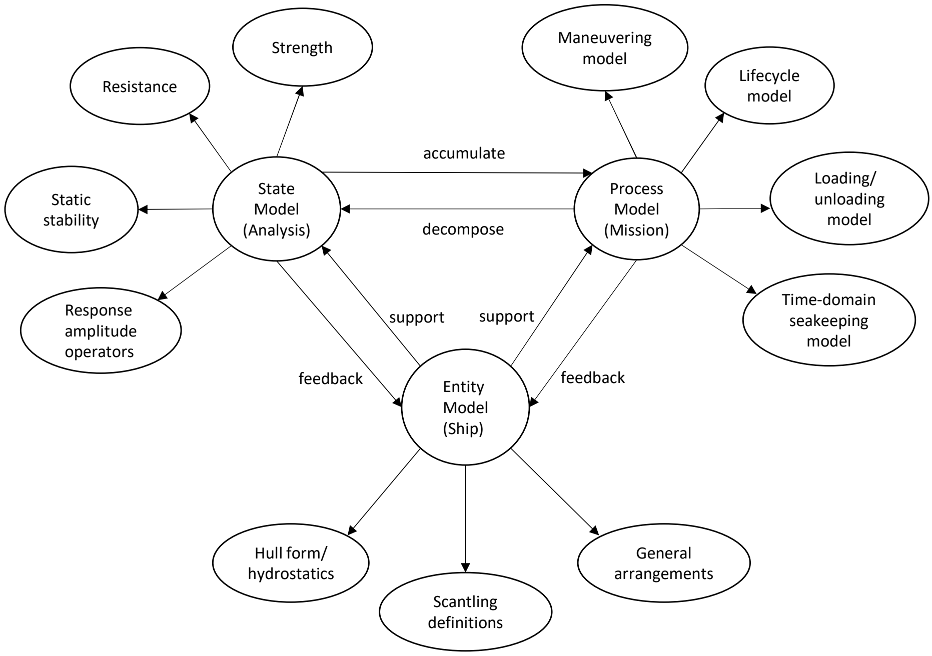


Figure 3.5: Examples of virtual prototyping models in conceptual ship design. Updated from (Fonseca and Gaspar 2015); adapted from (He et al. 2014).

simulations accounting for time progression, as opposed to static analyses. Examples of process models in the ship design context are simulations of navigation, manoeuvring, and time-domain simulations of motion and structural responses. In a virtual prototyping context, results from different state and process models guide system design by forecasting performance of a candidate design in a few expected or representative operational conditions. The entity model is used as support or input to set up the state and process models. Conversely, designers use results from the latter two as feedback to evaluate a candidate design and, if necessary, modify it toward attaining a desired performance according to the simulations' results.

The framework can be compared to requirements expected from a digital twin. It also uses the asset model as an input to modelling of behaviour, a notion that aligns with the vision of a digital twin revolving around a common ship representation. However, having been originally proposed to deal with virtual prototyping, it leaves out some digital twin characteristics and objectives. The characteristic is that in a digital twin, states are not simply forecast based on an underlying mathematical model (i.e., a state model); they are linked to observations measured from an operating vessel and context (for example, sensor readings and metocean data). The objective is that in a digital twin, simulation results are used to guide decisions during operation, not to iterate across design alternatives. The distinction between analysis (i.e., a state model) and simulation (i.e., a process model) is much subtler than a straightforward relationship between decomposition and accumulation. Consider a highly coupled continuous simulation. The system state during a given time step is informative only if taken in context, and it might be meaningless if assessed in isolation. Thus, a simulation with complex, dynamic behaviours does not decompose meaningfully into a temporal sequence of static analyses. For instance, in a time-domain simulation of a maritime operation, it is only possible to effectively evaluate the motion responses when they are assessed over an interval which allows identification of resonances and other effects of interest. Similarly, if one is monitoring vibrations with the objective of assessing structural health (for example, as Soal et al. (2019) show), it is necessary to perform calculations based on the measurements' time histories. In these examples, the decomposition of time-dependent behaviour in instantaneous snapshots is not of great interest.

A distinction is proposed to settle this confusion. Analyses calculate results statically (i.e., they provide output states) based on a vessel's given input state, without dependence on past or future states. Simulations include dependence of states regarding time. They are distinguished across two types. Discrete simulations are conducted across a large time interval, during which transitions between states occur step-wise. The discrete approach might use static analyses to calculate a sequence of snapshots that capture ship states across time instants. Continuous simulations model ship states as a continuous function of time, involving tight coupling between successive states. The progression of states over time might be given in several forms, including a closed-form expression of simple evaluation to differential equations that require numerical solutions during each time step. Figure 3.6 summarises these observations in a flowchart. Not shown in the figure are cases in which various behavioural models might
rel
on

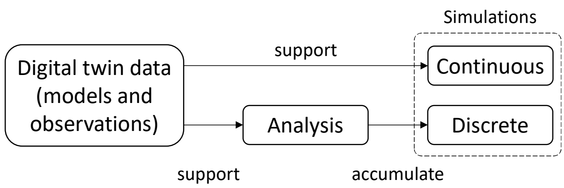


Figure 3.6: Adaptation of the virtual prototyping framework to digital twins.

3.7 Interoperability and data standardisation

Common to many engineering fields, software in maritime and naval architecture is traditionally siloed, with software suites usually not adopting formats and conventions shared among vendors. More than a decade ago (Szykman et al. 2001; Rachuri et al. 2008), it was argued that vendors viewed proprietary formats as competitive advantages to attract and maintain customers. A few years later, Nowacki (2010) discussed competing trends between closed turnkey systems of increasing functional scope and open systems with a modular, partly heterogeneous, approach to system growth. More recently, Gaspar (2019) noted that development trends have consolidated across product design and lifecycle management suites of ship engineering, from one side, and specialised CAD and Computer-Aided Engineering (CAE) software that allow more flexible connections to external suites, from the other. The new drive toward digitisation and digital twins might increase demand for interoperable software systems that allow data to be accessed, parsed, and exchanged easily. A position paper from DNV emphasises this issue, pointing to standardisation to enable digitalisation in the maritime industry (Låg and With 2017). It argues that besides offering technical advantages during data exchanges, standardisation might also offer managerial and engineering improvements to the ship value chain. For example, it might reduce the time required to rework digital models and allow engineers to focus more on acquiring domain-relevant knowledge when learning a certain software suite to undertake design and analysis, rather than memorising the conventions of its digital formats. Software providers appear to notice such demand for openness, with some claiming to support data exchanges in their solutions. However, it is unknown whether these intentions translate to actual formats, libraries, and interfaces that enable interoperability in practice. Consider the Industry Foundation Classes (IFC) approach for building information modelling. A web-loader for their format is developed on a public repository, with examples available in Three.js, a popular open source library for development of web-based 3D visualisations (Industry Foundation Classes (IFC) 2022; *three.s - IFC loader* 2022). It would be desirable to have analogous resources for the maritime domain.

3.8 Levels of standardisation

Discussions about standardisation need to be clear regarding the level of standardisation they address and the extent to which it translates to interoperability gains from a functional perspective. For instance, if two programs are written in the same programming language, their source codes share the same syntactic conventions and compiling tools. However, these commonalities in isolation are insufficient to ensure compatibility at end-user level, as they say nothing about both programs' capability to connect among each other through common data exchange formats and Application Programming Interface (API)s. Rachuri et al. (2008) introduced a typology of PLM standards that range from type zero to type three, defining the meaning of these standard types extensively and unambiguously. This section summarises these type categories; later, Section 4.2.3 maps them to the standards adopted within the scope of this project. In short, Type Zero are implementation languages, which describe the programming languages used to implement the standards computationally. Type One are information-modelling standards. They are languages for definition of information within digital schemas, such as Extensible Markup Language (XML), a popular document-based language, or the EXPRESS language which underlies ISO standards for product data. These languages are generic in the sense that they do not prescribe an application domain. On the other hand, Type Two are content standards that model domains of discourse.

They subdivide into: product information modelling and exchange standards, information exchange standards, product visualisation standards, e-business and value chain support standards, and security standards. The first three sub-types are particularly important to this work. In the context of digital twins, product information modelling standards accounts for product descriptions, operating states, and behaviours. In the context of PLM, they might extend to account for work, risk, and support management. Product visualisation standards address rendering and editing of product geometries. Information exchange standards include protocols, APIs, and other data exchange techniques. This work focuses on standards of Type Zero, One, and Two to the extent they can be used to model the digital twin content outlined earlier. Type Three are architectural framework standards. They set a holistic framework to harmonise use of the other standard types in the value chain, commonly involving various stakeholders. They extrapolate the scope of data standardisation and should be informed more strongly by organisational concerns, returning to the notion of a data thread.

3.9 Previous and current standardisation initiatives

The literature review examines two previous data standardisation initiatives in engineering to identify potential obstacles to standard adoption. The most prominent in the maritime and ship domain is International Organization for Standardization (ISO) 10303 standards for exchange of product model data, known informally as Standard for the Exchange of Product Model Data (STEP). STEP development began in 1984, with the first release about a decade later. Some ship-oriented sections of the standard were under revision and correction as recently as 2008. During an assessment of the state of product data technologies published on the same year, Gielingh (2008) argues that the industrial uptake of STEP and other product standards had been poor, pointing to several causes. One was vendors' lack of business motivation to adopt standards, but he also points to technical limitations of the standards at enabling consistent data exchanges among applications. Since the standard does not enable a common model that is shared by various tools, it limits the scope of the data that can be exchanged to the domain of discourse part which is covered both by software tools and the standard. In face of this challenge, one alternative employed in the literature has been to create a common ship model based on the data required by different software suites, instead of attempting to adopt STEP when exchanging data among them (Whitfield et al. 2011). Other reasons for low adoption include the high costs and technical challenges of implementing STEP. For example, portions of the standard that target ship data modelling are more than a thousand pages long, with the one that covers ship structures nearing two thousand (ISO 2004a).

It is interesting to compare the adoption and challenges experienced by STEP with the case of the IFC standards for Building Information Modelling (BIM), which allow the creation of a shared building model which can be accessed through different software suites. Gielingh (2008) argues that IFC shares technical limitations with STEP, but also notes a few differences between the initiatives. Software vendors had been more involved during development of IFC, possibly to achieve greater market penetration by offering interoperable products. At the time, Gielingh noted the trend of some governments starting to demand use of IFC standards for public tenders, pointing that this could lead to a breakthrough in adoption if replicated internationally. The prediction was correct. Currently, several countries, including Norway, mandate creation of IFC-based digital building models for public contractors. Recent research corroborates the influence such measures have had on adherence to IFC in those countries (Edirisinghe and London 2015). Compared to this experience in the civil engineering sector, the role that business and regulatory scenarios play during adoption of maritime standards is significantly different. Specifically, there is not a similar set of regulatory forces guiding standardisation in maritime, thus leaving the task of ensuring interoperability among digital systems to industrial stakeholders.

3.10 Monolithic and heterogeneous approaches to system development

In face of these challenges, it is more promising to have pragmatic standards that can solve smaller problems than to have comprehensive ones that might be too overwhelming to implement (Cameron, Waaler and Komulainen 2018). It is technically possible to design a comprehensive standard that would, in theory, provide a an absolute degree of standardisation. However, this approach brings at least two major risks. One is being perceived by stakeholders as unable to accommodate desired use cases due to excessive restrictiveness (Låg and With 2017). Another is that of being considered too cumbersome or dispendious to implement, thus failing to account for practical concerns by development teams in the maritime industry. Gaspar (2019) mentions that ship design software represents “a much smaller spectrum of the field compared to civil or mechanical engineering”, noting that it has “even more specific software suits (sic) developed over the time”. Given the relatively niche scope of naval architecture software compared to the two other engineering sectors mentioned, it can be concluded that the amount of resources available to development teams in that sector is in a smaller order of magnitude too. By attempting to propose a completely standardised ship model, a solution might result that is so optimised to some modelling choices that it becomes an unacceptable compromise to those not beginning from these same choices. The possibility of intermediate levels of interoperability might also reduce, generating a dilemma where either the parties interested in exchanging data adhere strictly to a highly detailed standard and are thus able to exchange the data seamlessly, or the standard provides no utility during that exchange. Based on this situation and informed by previous research (Szykman et al. 2001; Gaspar 2018b) and by the design choices (not explicitly stated, but inferred by the author) of ISO 19848, a comparison can be drawn between monolithic software systems and standardisation approaches with heterogeneous ones (Table 3.1). The table’s last two rows describe the trade-offs involved in the different standardisation approaches advanced by each type. An example of heterogeneous system with partial standardisation would be a digital service that provides an API for access through the internet, even if the syntactical conventions for interacting with that API are not standardised in relation to other ship-oriented digital services.

While heterogeneous digital systems might not entirely displace monolithic ones, they are steadily increasing their prevalence among internet-based services.

3.11 Identification of standardisation drivers

In summary, a standard's adoption does not depend solely on its capability of adequately modelling a given domain of discourse. Rather, it is influenced by a complex combination of factors, from stakeholder's (sometimes conflicting) perceptions on the opportunities and threats brought by digital systems' interoperability to the regulatory frameworks enforcing standard adoption, as in the case of the civil engineering industry. Still, the literature review identified promising drivers to guide future standardisation attempts in ship industry. This was done mainly by examining past experiences with the SFI and STEP initiatives, and then looking into comments and qualitative trends pointed by other authors. The first driver is pragmatic scope. This means that digital twin standardisation should not be guided by a single comprehensive and extensive standard, but rather by a combination of smaller, more specialised ones. This last point brings the discussion to the second driver, which is the necessity to support heterogeneous systems. Since the digital twin will be based upon a collection of standards, it is needed to ensure they provide a baseline level of compatibility among each other through APIs, data exchange protocols, or other mechanisms. Achieving this in practice requires balancing a trade-off between increased modelling strictness and out-of-the-box interoperability, from one side, and increased customisability, from the other.[2] Together, the first and second drivers aim to reduce barriers to adoption and to simplify implementation (though hopefully not at the expense of oversimplifying the engineering problem to be simulated). This simplification is expected also because, by providing stakeholders with an assortment of standards covering different domains of discourse in the digital twin, they have the opportunity to select and implement the minimum set which is necessary to cover their intended use cases. The final drivers are openness and intelligibility, which are closely intertwined. Openness means that the standard should avoid, as far as possible, proprietary solutions for the reasons discussed earlier (i.e., establish a level playing field and avoiding lock-in risks). Intelligibility stems from the need for users to adequately comprehend the data and models they are consuming. This has already been put forth as a requirement for naval architecture design software (Andrews 2018) and remains true for future digital twin ships.

3.12 Gaps and remaining research questions

This chapter presented a literature review with the aim of answering the first pair of research questions "on the challenges to standardisation of digital twin data and

[2] Alternatively, Datta (2017) argues for automated mechanisms through which systems "understand *what needs to be understood*", italics his own, and are able to obtain the "glue" to facilitate interoperability. The candidate recognises this is a desirable capability, but fails to see how to realise this vision with existing digital technologies.

Table 3.1: Comparison between monolithic and heterogeneous digital systems, with corresponding standardisation approaches (Fonseca and Gaspar 2022).

	Monolithic digital systems	Heterogeneous digital systems
Consumption and exchange of digital content	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	Complete specification of ship models and data.	Flexible model specifications, allowing for customized implementations and use cases.
Standardisation initiatives' aim	Full compatibility of a given model across systems.	Partial compatibility of model across systems, plus consistency respective to the chosen implementation.

paths to future standardisation”. The current state of digital twin ships and challenges to standardisation were placed in perspective. Given the novelty of the digital twin concept and the absence of digital twin standardisation approaches departing from a naval architecture perspective, it was needed to identify principles to guide the modelling approach in the remaining of the thesis, which were encapsulated in four drivers. The following chapters aim to outline a standards-based approach to modelling of digital twin ships by answering the second and third pairs of research questions: "on the application of existing standards to digital twin ships”, and "on suitable methods for further digital twin standardisation”, respectively. Given the ambitious digital twin scope and the relatively niche market of naval architecture software, it is judged wise to leverage existing standards which can cover parts of the digital twin scope, even if they have not been developed specifically for the ship industry. Thus, to answer the second pair of research questions, it will be needed to map the scope of existing standards to digital twin ships, demonstrate their functionality, and evaluate to what extent they cover digital twin content. Once the modelling scope of existing standards is understood, the work turns to the digital twin content types (or domains of discourse) which still lack standardisation, with the aim of contributing to address these gaps in light of the drivers identified earlier. Thus, to answer the third and last pair of research questions, it is needed to propose standardisation methods for the most relevant digital twin domains which are not yet covered by existing standards, demonstrate their functionality, evaluate their adequateness to simulation and their broader implications to the ship value chain.

Chapter 4

Modelling approach

4.1 Chapter aim

This chapter does not relate directly to one of the appended papers. Instead, it explains the modelling principles that underlie the following case studies, which should be detailed for two reasons. The first is to familiarise uninitiated readers with the adopted formats and data structures, allowing adequate understanding of the research. Given the page limitation of published articles, it was impossible to detail these topics in the accompanying publications. Second is to offer an overview of modelling conventions that guided development of the case studies. After reading the appended papers, readers would have scattered glimpses of the modelling approaches across content domains and at different stages of maturity. This chapter unites these partial views into a cohesive framework.

4.2 Standardisation framework

4.2.1 Overview

Figure 4.1 shows the framework that guided this work's approach to digital twin modelling throughout the study. When accounting for operations with multiple vessels, the framework could be expanded with data from various ships. The first group of data includes the ship model and sensor observations. The ship model represents the ship as a physical asset, and the sensor observations support monitoring and calculation of ship states. Modelling operational context is analogous; the application should include a static digital model of the operating region, which might be augmented with metocean data to represent weather states during operation. Simulation modules might access the ship and environment data to account for various behavioural aspects during operation. The adoption of data standards is expected to simplify reuse of digital twin content due to shared modelling conventions. However, it is necessary to consider the scope of such standardisation more precisely.

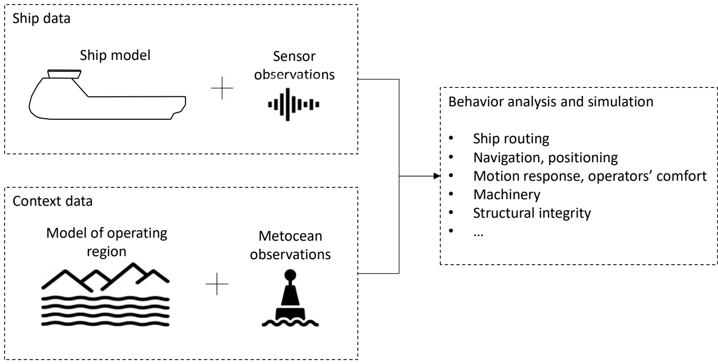


Figure 4.1: General digital twin data framework.

4.2.2 Web-driven standardisation

A web-driven approach to digital twin modelling is used. The web has long associated with transmission of lightweight documents and services over the internet, and advancements and dissemination of computer power, and network infrastructure, make it possible to create sophisticated web-based applications. These apps are supported by increasingly robust architectures for back-end integration with servers, databases, and devices connected to the internet of things. The reasons for this choice relate to compatibility, support, ease of use, and availability of open source libraries, as Gaspar (2017) outlined. The web represents an example of standards that enable software compatibility among systems. In practice, web pages can be accessed and executed by any device running a modern web browser, including personal computers and mobile devices, regardless of operating system. That web apps are hosted on the internet make them convenient to distribute. When rolling out an update, a developer might simply ship the new source code to the host server, and all users connected to the internet are able to access the latest version immediately. Web-based applications allow creation of a Graphical User Interface (GUI) for effective interaction with apps and simulations. GUI development is supported by diverse resources, some embedded into cornerstone web standards (Section 4.3) and others available through libraries and frameworks that third parties publish . A particularly strong example in the development environment is the assortment of libraries for user interface, including libraries for data visualisations, plots, and charts, such as D3.js (Bostock, Ogievetsky and Heer 2011), and for 3D graphics, such as Three.js (Section 4.6.3). Web-based development is supported by a strong, open source culture, with many libraries for several purposes under development. Although not overwhelming, an increasing number of applications and libraries target the maritime sector. For example, the OpenBridge design system publishes a collection of indicator components that can be combined to create web-based interfaces for ship navigation and manoeuvring, including operation of Dynamic Positioning (DP) systems (Nordby, Gernez and Mallam 2019).

4.2.3 Selection of standards

The selection of standards in this work would ideally be limited to options that are both open and vendor-neutral. In practice, these conditions are likely too restrictive, and compromises might be necessary to allow for any options. For example, if "open" is taken to mean that a standard should be available to any interested parties, then the several ISO standards which are sold as copyrighted Portable Document Format (PDF) files are not open (Smith 2006). In other cases, a standard might be open to use but have its development strongly influenced by a single vendor, thus compromising its neutrality in relation to others. This might occur if, for example, a company decides to reclaim property over a technology that it previously made open for others to use (Villani, Bonnet, Rondepierre et al. 2018), or if it adds proprietary features to an existing standard to undermine competitors, such as the embrace, extend, and extinguish tactic. Table 4.1 summarises the selected standards. JavaScript and HTML correspond to the implementation language, and JSON is used for information modelling. These standards are content-agnostic, so they are complemented by others that target aspects of the digital twin (Type Two). Type One and Two standards were selected to be compatible with a web-based approach, but this does not mean that digital twin data can be used only in web environments; standards such as STL and glTF are supported by several software for 3D modelling and computer-aided design. ISO 19848[1] and the Vessel.js ship model specifications can be modelled in JSON, which, in practice, is a text file written using syntactic conventions, thus supported by any text-viewing software.

Table 4.1: Main standards adopted in this study. Updated from (Fonseca et al. 2022). *Early outline of specification proposed in this work, not qualifying as a published standard.

Type	Scope	Adopted
0	Implementation languages	JavaScript, HTML
1	Information modelling	JSON
2	Content standards (domains of discourse)	glTF, STL, ISO 19848, DNV VIS, Ship model spec,* Simulation module template*

4.3 Implementation languages

4.3.1 HyperText Markup Language

Web pages and applications are based on three implementation languages — HyperText Markup Language (HTML), JavaScript, and Cascading Style Sheets (CSS) — with the first two being particularly important to development of the topic discussed in this thesis. HTML is used to write and format digital documents displayed when a user opens a web page (Gaspar 2017). The word "markup" in the initialism means that

[1] Ships and marine technology—Standard data for shipboard machinery and equipment.

the document annotation is visually distinguishable from the document content. More specifically, document content might be, for example, text, images, vector graphics, 3D graphics, or videos. They are annotated with tag pairs that define how content is displayed, embedded in or submitted from the page. For example, element tags can be used to structure text in terms of headings, paragraphs, lists, or links. Element tags such as “img” and “video” allow placing of external image and video files on the web page. Others, such as “input,” “range,” or “button,” allow a user to interact with forms and sliders, and submit content. Tags associated with each element can be used to adjust style, such as formatting, fonts, colours, and size. Placement of tags also defines the corresponding Document Object Model (DOM), allowing manipulation of document content based on its tree structure. Since the element concept in HTML is versatile, this list is not exhaustive. The diversity of features found in current web pages serves as an example of functionalities provided by the language. HTML and DOM are maintained by the Web Hypertext Application Technology Working Group (WHATWG) (WHATWG 2022). The role played by CSS is limited to describing visual presentation of documents written in HTML, in case developers want to separate a description from the document itself. They might specify all style characteristics, such as colours, fonts, and layouts, using a succinct style sheet syntax. Each style sheet stores a set of pairs with a property and value that have corresponding units, if necessary. Together, they specify a desired characteristic (for example, margin: 0.5 cm). A style sheet links to one or more selectors, allowing them to be applied to any content in a document, from the entire body to a set of elements, or a specific element. A cascading mechanism propagates the values defined on a style sheet to every element displayed in the document, resolving eventual assignment conflicts. The language is maintained by the World Wide Web Consortium (W3C) (World Wide Web Consortium (W3C) 2022).

4.3.2 JavaScript

JavaScript is the major programming language for the web, in fact the only one readily supported by web browsers (Gaspar 2017). It was created to manipulate and interact with content in HTML documents on the client-side, but it has since expanded its uses to other domains, such as back-end environments and standalone applications. Among JavaScript’s primary characteristics are its high level, just-in-time compilation, and dynamic typing. High level means that it hides basic machine procedures, such as memory management, from the source code. JavaScript is also highly expressive, offering sophisticated functions for variable manipulation.[2] Just-in-time compilation means that instead of being compiled into a separate file, it is compiled during execution (i.e., at run time) by a compiler such as the V8 open source engine, developed by Google and used in the Chrome browser (V8 project authors 2022).[3] Benchmarks suggest that JavaScript achieves performance comparable, and in some cases superior, to

[2]The concept of high expressive power is sometimes conflated with the general notion of a high-level programming language (Gamelab Conference 2018). Recent programming languages, such as Rust, are splitting this conflation by offering high expressiveness with access to low-level hardware interaction (Jung 2020).

[3]These standards evolved and gained complexity with publication of new versions, and thus the development and maintenance of web engines on which browsers are based becomes increasingly resource intensive. For example, after spending years building the Edge browser, based on its own proprietary engines for HTML

other interpreted and just-in-time compiled languages, while maintaining syntactical succinctness of its source code (Gaspar 2017). Dynamic typing (i.e., a dynamically typed language) means the programming language infers variable and construct types during run time, as opposed to static typing, in which variables types are assigned during compilation, usually based on explicit indications provided by the programmer. In addition to being dynamically typed, JavaScript is very flexible when handling variable types. It is possible to assign the same variable different data types during execution, so a variable which during the beginning of execution contained a string might later be reassigned with a Boolean, number, and so on. JavaScript supports object-oriented programming, among other paradigms. JavaScript's implementation is based on the ECMAScript standard, published by non-profit organisation Ecma International (Ecma International 2021).

4.4 Information modelling: JavaScript Object Notation

JavaScript Object Notation (JSON) is a text-based format for storage and transmission of data objects. Despite being developed based on JavaScript's object syntax, it has since garnered support in the software industry, so that today, there exist libraries for handling JSON in various programming languages. The simplicity of its syntax means that JSON is human readable, allowing data interpretation by stakeholders, without dependency on proprietary software suites. A JSON file stores either an object or array, with the object structure being the most common throughout this study. Figure 4.2 depicts a graphical scheme of the JSON syntax; Listing 4.1 illustrates how a trivial JSON object based on that syntax can be created to store string, number, object, string, and Boolean values, in order. In comparison to XML, JSON offers simpler syntax. Several tools support data conversion between formats. JSON is standardised through various entities, including ISO and Ecma International (Ecma International 2017).

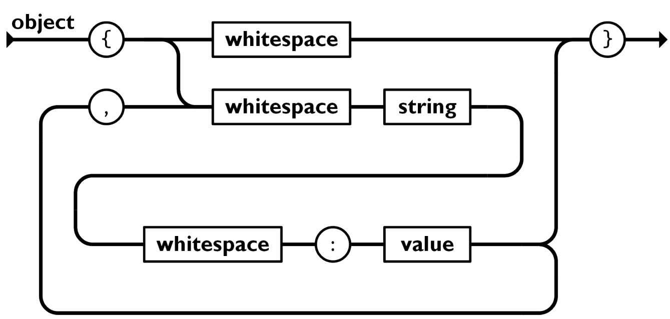


Figure 4.2: A syntax diagram of an object in JSON (Ecma International 2017).

rendering and JavaScript execution, Microsoft abandoned the project and rebuilt it based on Google's dominant Chromium code base (Warren 2019). This factor might require special assessment to allow long-term sustainability of the web, but it does not bring hindrances to the scope of functionalities implemented in this study.

Code listing 4.1: A dummy JSON file.

```
{
    "string_field": "text",
    "number_field": 30,
    "object_field": {
        "property_field": "text"
    },
    "array_field": [1, 2, 3],
    "boolean_field": "true"
}
```

4.5 Ship taxonomies

4.5.1 Candidate alternatives

To allow organised inclusion of ship data (i.e., product model and states) in a digital twin, it is necessary to rely on a taxonomy or schema that sorts the content into structures. They arrange data hierarchically according to a view of the ship that is suitable to a purpose during the lifecycle. To be adequate to a comprehensive digital twin, a taxonomy should adopt a functional view that supports ship operations and provides a high degree of hierarchical detailing that traces even minor components. In practice, it is likely that various organisations and stakeholders will settle for disparate taxonomies that suit their strategies and operational methods. When searching the literature for taxonomies that could be used in this study, few candidates were found, for example, those listed in reference (Pal 2015). Most of them were later disconsidered for different reasons, for instance, the Expanded Ship Work Breakdown Structure (ESWBS) is scoped toward warships, which are not the main focus of this research. Among the candidate taxonomies that support civilian vessels, the SFI Group System (Xantic 2001) was initially considered.[4] However, it was dismissed because it is licensed commercially, requiring payment for access to a copy, and because its status in terms of support and updates is unclear due to the lack of publications presenting its development or maintenance in the last decade. The most promising alternatives are the two naming rules for sensor data advanced in the annexes to the ISO 19848 standard (Section 4.7): Vessel Information Structures (VIS) by DNV and JSMEA-MAC by the Japan Ship Machinery and Equipment Association (JSMEA). Both are openly published and documented on the internet, allowing users to download Excel tables containing the entire naming rules (DNV 2022; Japan Ship Machinery and Equipment Association (JSMEA) 2022). Due to the candidate's location in Norway, VIS was adopted as this work's primary taxonomy. Nevertheless, Chapter 6 explains how object-oriented structures might be used to link digital twin data to a stakeholder's taxonomy of choice.

[4] Ship Research Institute, or *Skipsteknisk Forskningsinstitutt* in Norwegian. Later merged with SINTEF.

4.5.2 Vessel Information Structures

In a conference paper published in 2008, DNV presents as a data model developed for use as a basis for classification throughout a vessel's entire lifecycle (Vindøy 2008). In 2018, with DNV's push for digital standardisation in maritime, VIS was annexed to ISO 19848 as Vessel Information Systems. A 2021 paper presents a revision of VIS, now under the name of Vessel Information Structures, with primary focus on standardising sensor identification based on reference libraries (Låg, Vindøy and Ramsrud 2021).[5] The paper explains plans by (DNV 2022) to publish main VIS releases twice a year, thus providing continuous support. VIS was based on principles outlined in the ISO 15926 standard (ISO 2004b),[6] and is primarily organised around a functional hierarchy of a vessel based on names and numerical tags. Figure 4.3 shows the highest level of the hierarchy. VIS provides various terms with which to organise entries, depending on the type of functional relation they fulfil (Figure 4.4). In VIS, a basic functional unit is called a function leaf, mutually exclusive instances of a same role are function selections, functions that together constitute an encompassing parent are compositions, and functions sorted together simply by convenience are called groups. A function leaf can be assigned to a corresponding physical component, and a separate equipment list details all components with their respective identification tags. Components might, in turn, be further detailed in a hierarchy of sub-functions. Finally, sub-function leaves are mapped to a component or component selection at the terminal nodes of the hierarchy tree. The number of functions and components amount to more than 2,500 and 750, respectively. Given VIS's scope as support for ship classification, it covers a larger scope than the vessel data model, including codes for documentation about work tasks, schedules, and materials. The schema is commonly extensive in sections that cover ship functions, suggesting that it might fulfil the detailing of requirements discussed earlier.

Admittedly, the labels provided by either VIS or its alternative JSMEA-MAC cover only a portion of the digital twin content — ship representation — while leaving aside simulations of ship behaviour and environmental factors for which the digital twin accounts. At a higher level, the latter concern can be addressed by aligning to the list of relevant metocean parameters to be included in databases for offshore structure design and operation according to the ISO 19901-1 standard (ISO 2015)[7]: waves, currents, winds, ice, water levels, and others. On the other hand, it is argued that defining a classification of ship behaviour models contained in the digital twin is undesirable. Section 1.4 argued that development of digital twins currently undergoes an exploratory stage during which promising use cases are identified and tested out. Thus, it is premature to prescribe a list of ship behaviours which could or should be modelled by the digital twin. In fact, it is also unnecessary, as existing software repositories allow identification of a given package according to multiple customised tags, avoiding the problem

[5]This revision was published after the research work for this thesis had been developed, thus it was not used in case studies. It focuses on addressing inconsistencies when constructing descriptive sensor tags, an issue discussed in Paper II, Section 5.2.

[6]Namely: Industrial automation systems and integration—Integration of lifecycle data for process plants, including oil and gas production facilities.

[7]Namely: Petroleum and natural gas industries — Specific requirements for offshore structures — Part 1: Metocean design and operating considerations

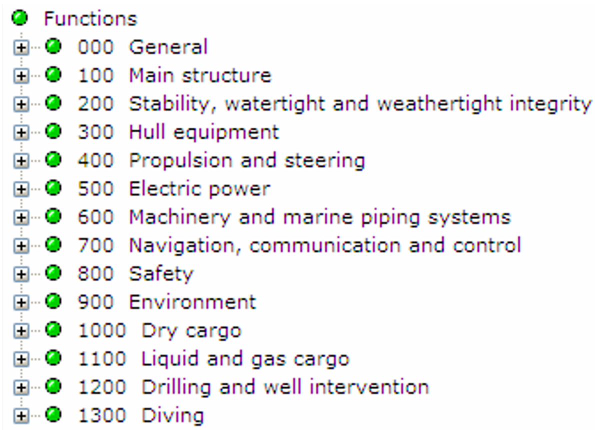


Figure 4.3: Top level VIS functions (Vindøy 2008).

Type	Code	Common name	Function name	Type	Code	Name
Function group	**521**	**Emergency electric power generation**	**Emergency electric power generation**			
Function composition	521i	Emergency electric power generation arrangement	Emergency electric power generation arrangement			
Function leaf	521.1	Emergency generator driver; [C101] Emergency generator engine; [C102] Emergency generator steam turbine; .... [E202] Emergency generator electric power unit	Emergency electric power generation - generator driver	Component selection	CS1	Driver
Function leaf	521.2	Emergency generator driver to generator shaft	Emergency electric power - generator driver to generator power transfer	Component	C221	Shaft
Function leaf	521.3	Emergency generator	Emergency electric power generation - generator	Component	E32	Generator
Function leaf	521.4	Emergency generator supporting structures	Emergency electric power generation - generator arrangement supporting	Component	H601	Supporting structure

Figure 4.4: Excerpt of the generic product model (DNV 2022).

of fixing a set of categories altogether (for example, (npm 2022)). This allows a simulation algorithm to be tagged according to its engineering domain, such as structural resistance, to its mathematical model, such as neural networks, and so on.

4.6 Ship representation

4.6.1 A first attempt: custom ship model

Following from Section 3.11's argument, the modelling of ship representation is chosen to follow simplicity, heterogeneous support, openness, and intelligibility as standardisation drivers. It is judged more adequate to aim for a simple, yet extensible approach rather than modelling a ship by creating a specific data structure for each possible component and part that might be installed inside it. These principles can be observed, for example, in the provision of tools for creation of ship models according to various conventions, especially regarding choice of taxonomy (one of which is VIS) and the degree of detailing in 3D visualisations. In the first attempt, the ship model is based on a combination of JSON metadata and 3D files. During an initial implementation, the ship representation consisted of a unique ship specification, a metadata specification for each ship component, and 3D model files. The metadata contained identification tags, coordinate positions that defined where the corresponding 3D model would be placed inside of the ship, and, if necessary, weight and centre of gravity. Metadata specifications for physically identical components might reference the same 3D model file, allowing it to be placed in two or more different positions inside the vessel. This approach was used in the case study in Chapter 5, where listings in the chapter represent examples of how this was done in practice.

4.6.2 A second attempt: adoption of Vessel.js library

Motivation

As the project evolved, consistency was achieved between the requirements and solutions identified during the case studies, and ones advanced by the Vessel.js library (Gaspar 2018b). Thus, during the remaining case studies (Chapters 6 and 7), the ship model was implemented using Vessel.js. The reasoning behind the transition will become clearer when the case studies are presented later. They are mostly related to the advantages of including more developed functionalities for efficient loading of repeated components, including their 3D models, and calculation of relevant simulation parameters. The following paragraphs describe the constituent elements of a ship specification in the Vessel.js library. Each element contains a property named “affiliations”, listing one or more tags which can later be used to map it to the desired taxonomies.

Hull and structure

A ship specification lists fields for simplified modelling of structural arrangements with hulls, decks, and bulkheads. A hull is based on a set of main dimensions and an accompanying table of offsets, which is used to generate the 3D geometry automatically. Taking as example the simplified barge presented in (Fonseca 2018), a prismatic cuboid hull is defined as illustrated by Listing 4.2.

Code listing 4.2: Excerpt of an object defining a prismatic hull with main dimensions and table of offsets. Updated from (Fonseca 2018).

```
"hull": {
    "attributes": {
        "LOA": 22.5,
        "BOA": 10,
        "Depth": 2.5,
        "FPP": 22.5,
        "APP": 0
    },
    "halfBreadths": {
        "waterlines": [0, 0, 1],
        "stations": [0, 1],
        "table": [[0, 0], [1, 1], [1, 1]]
    },
    affiliations: {
        "SBSD": "Hull",
        ...
    }
}
```

The 3D hull generator is currently limited to conventional, single-hull forms, so it is unable to render catamaran and Small Waterplane Area Twin Hull (SWATH) shapes, for example. The decks and bulkheads allow a user to specify desired elements, with positions, thicknesses, material densities, and, in the case of decks, longitudinal spans, as exemplified by the deck specification in Listing 4.3. The structural elements are visualised in the 3D model using flat surfaces generated automatically.

Code listing 4.3: Excerpt of an object defining a deck on top of a ballast tank. Updated from (Fonseca 2018).

```
"decks": {
    "BallastTop": {
        "zFloor": 0.4,
        "thickness": 0.01,
        "xAft": 0,
        "xFwd": 22.5,
        "yCentre": 0,
        "breadth": 10,
        "density": 7850
    },
    affiliations: {...}
}
```

The weights of hull and structural elements are calculated differently. The hull weight and CG are estimated with semi-empirical formulas which can be corrected for different vessel types (Parsons 2004). The weight of each structural element is calculated as its volume multiplied by the specified material density (in the previous example, steel), and the element's CG is placed at its geometric center.

Base and derived objects

Vessel.js advances the notion of “base” and “derived” objects in the ship model (Listings 4.4, and 4.5, respectively). A base object is a generic construct that represented a component or tank; a derived object allows replication of a base object across locations inside of a vessel by specifying the desired coordinates. A base object includes weight, Center of Gravity (CG) data, and a path to corresponding visualisation files. A user might choose to load a detailed 3D model contained in an external file (as explained in the following section) or simply specify linear dimensions so that the loader automatically creates a box in the visualisation, which is convenient to represent tanks. A base object’s CG can be defined in three ways. If the object’s CG is left unspecified and the base object has been defined based on three linear dimensions, the algorithm will automatically place it in the center of the corresponding bounding box. Alternatively, the CG can be defined explicitly, as Listing 4.4 exemplifies. Finally, it is possible to map various CG positions depending on a tank’s filling level. In that case, in addition to the object’s lightweight in kilograms, it is necessary to define its volume capacity in cubic metres, cargo content density, and a list of CG positions for different tank filling ratios.[8]

Code listing 4.4: Example of base object.

```
{
  "id": "MainEngine",
  "affiliations": {},
  "file3D": "MainEngine.stl",
  "weightInformation": {
    "lightweight": 6480,
    "cg": [0, 0, 1.35]
  }
}
```

Code listing 4.5: Example of derived object.

```
{
  "id": "MainEngineSB",
  "baseObject": "MainEngine",
  "affiliations": {
    "VIS": "411.1 Propulsion driver, SB",
    "Deck": "Tween deck"
  },
  "referenceState": {
    "xCentre": 6,
    "yCentre": 4,
    "zBase": 5.3
  }
}
```

Figures 4.5 and 4.6 show the visualisation of a barge created with Vessel.js. It is composed of a hull, a deck, a bulkhead, two cargo tanks, and two bottom tanks defined

[8]It is a current limitation of the library that it defines cargo density as a base object’s (i.e., a tank’s) property, thus complicating modification of the cargo characteristics. This starts from the assumption that each tank will always carry the same type of cargo with a same density, which often does not hold true.

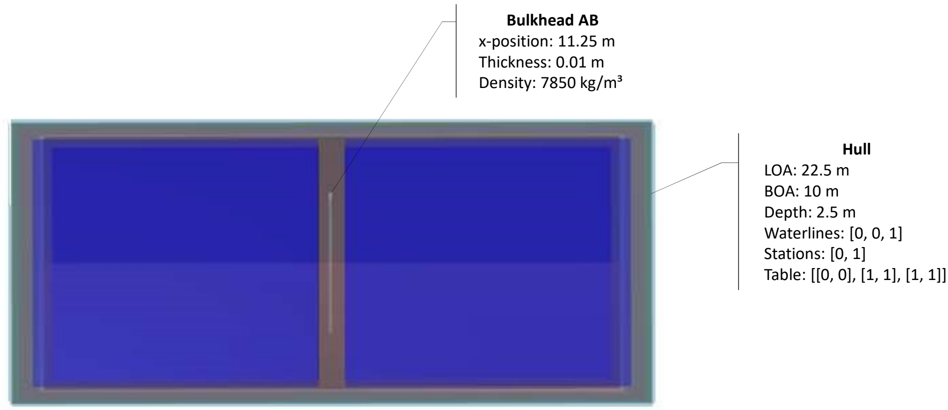


Figure 4.5: Barge's top view, showing relevant hull and bulkhead properties (Fonseca 2018).

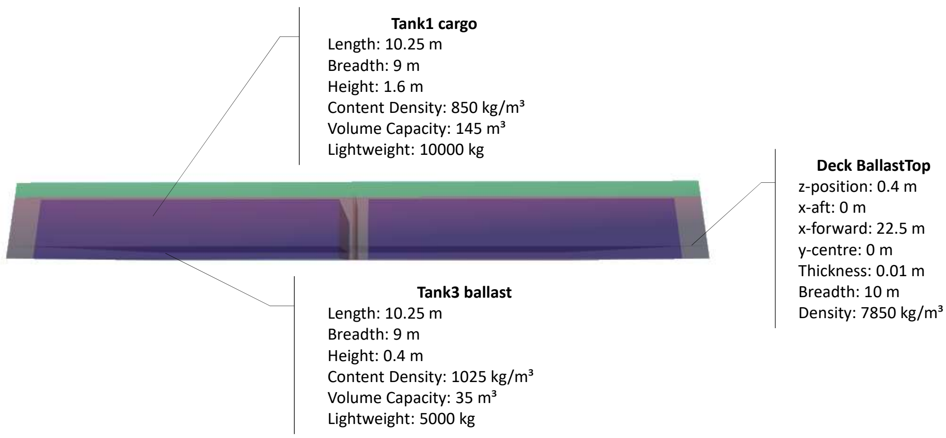


Figure 4.6: Barge's side view with deck and tank properties (Fonseca 2018).

Other relevant parameters

Each ship specification holds an object named `calculationParameters` which aggregates additional coefficients needed to perform any of the desired simulations. Examples of such parameters are the design characteristics and semi empirical parameters to estimate ship weight. This object's structure is admittedly somewhat arbitrary in that it aggregates dissimilar parameters for convenience, without including an explanation of their meaning or physical units. This is one area where the previous attempt of ship representation with a custom ship model fared better and could inform future development of Vessel.js, as shown by Listing 5.1.

4.6.3 3D visualisation models

While the Vessel.js library is able to create 3D models for hull, structure and cuboid base objects automatically, it has been also necessary to load more detailed 3D models into the ship visualisations developed for the case studies. This was done with polygon-based formats rather than splines-based CAD formats, ensuring it was possible to render and manipulate them during real-time simulations. A concern for simplicity influenced the choice of models and 3D formats. It is insufficient for a format to be documented standard, as this leaves the workload of implementing the specifications to the format adopters. On the contrary, the standardisation entity should provide adopters with tools to load files in the proposed format to existing software systems (i.e., loaders) and to convert them to other consolidated standards (i.e., converters), thus ensuring broad broader compatibility (Robinet et al. 2018). The case studies used two formats — STL and glTF — and all visualisations were created using the Three.js library.

STL

STL was developed by 3D Systems for stereolithography 3D printing. The format describes geometries that comprise triangulated surfaces. Each facet is defined by a unit normal and its three vertices, ordered by the right-hand rule. An STL file can be stored as either text or binary.

Graphics Language Transmission Format

The Graphics Language Transmission Format (glTF) is a standard for 3D scenes, which can include geometries, colours, textures, assemblies, and animations. The word “transmission” applies in this context because glTF was designed to minimise the size of assets and the processing required to render them during run time, as opposed to simply storing scenes in an exchangeable file format (Robinet et al. 2018). It can be stored either as a self-contained binary file or a JSON file, possibly referencing external textures and binary resources. The Khronos Group develops and maintains glTF (Khronos Group 2022).

Three.js visualisation library

Three.js is an open source JavaScript library for creating and rendering 3D graphics on the web (Three.js authors 2022). It provides tools to create and control geometries, animations, materials, shaders, lights, and cameras. It offers loaders for both adopted formats and others. Graphics are displayed using Web Graphics Library (WebGL) API.

API renders graphics on web browsers without using proprietary extensions, and allows acceleration of 3D scenes using specialised graphical processing units. The Khronos Group develops and maintains WebGL.

4.7 Ship states and sensor readings

Ship states in a digital twin relate closely to sensor readings collected during operation. An alternative is the 61162 series of standards by the International Electrotechnical Commission (IEC), which allows exchanges of data from navigational equipment. ISO 19848, namely "Ships and marine technology — Standard data for shipboard machinery and equipment" (ISO 2018b), is used. It is a more recent document with a more versatile scope, thus providing higher value for research. It was designed to allow data exchanges on-board ships, though ISO recognises that in the future, it might also be used when linking shipboard servers to the internet, and, it is possible to infer, to a digital twin. The standard can be implemented in JSON, XML, or CSV formats, with the last recommended for large data series. The standard defines two types of structures to organise data and provides diagrams to illustrate their respective logical structures (Figures 4.7 and 4.8). The first is the "data channel list", which contains metadata regarding a list of data channels (i.e., a collection of transmission channels that sends one or more measured variables). A data channel package is structured in a header and a main body. The header contains ship and author identification, for example, the IMO number of the vessel to which the data log belongs and the name of the person or instrumentation system which wrote the data, respectively. Each channel has metadata fields for data channel identification, according to desired taxonomies and properties, such as type (for example, instantaneous, average, calculated, alert, and set point), descriptive names, physical units, and expected value ranges. The standard document exemplifies application of this structure to the data logged by an air cooling system with four data channels: outlet temperature and pressure of the cooling water, and two alerts specifying the status of the measured values. The temperature is measured as an instantaneous value with an update cycle of 1 second, while the pressure is calculated as the average value in the last hour, updated at every minute.

The second is "time series data" (Figure 4.8), which contains the data measurements themselves. The package follows a similar structure, with an introductory header and main data body. The time series can be either tabular or event data, where tabular is used for values that are normally updated at regular intervals, and event is used for values that occur at irregular intervals, such as alerts and manual inputs. Back to the air cooling example, in case the measured temperature in Celsius degrees rises above the threshold specified by the corresponding ranges, it will trigger an alert in the respective data channel pointing to this anomaly. The alert and the measurements which triggered them will be stored in the time series and event and tabular data, respectively.

This arrangement encapsulates sensor reading and metadata in modular packages, easing data interpretation during exchanges. As long as users have access to the corresponding data channel list, they can interpret the physical and operational meaning of a time series. The document also offers flexibility for users to decide how they

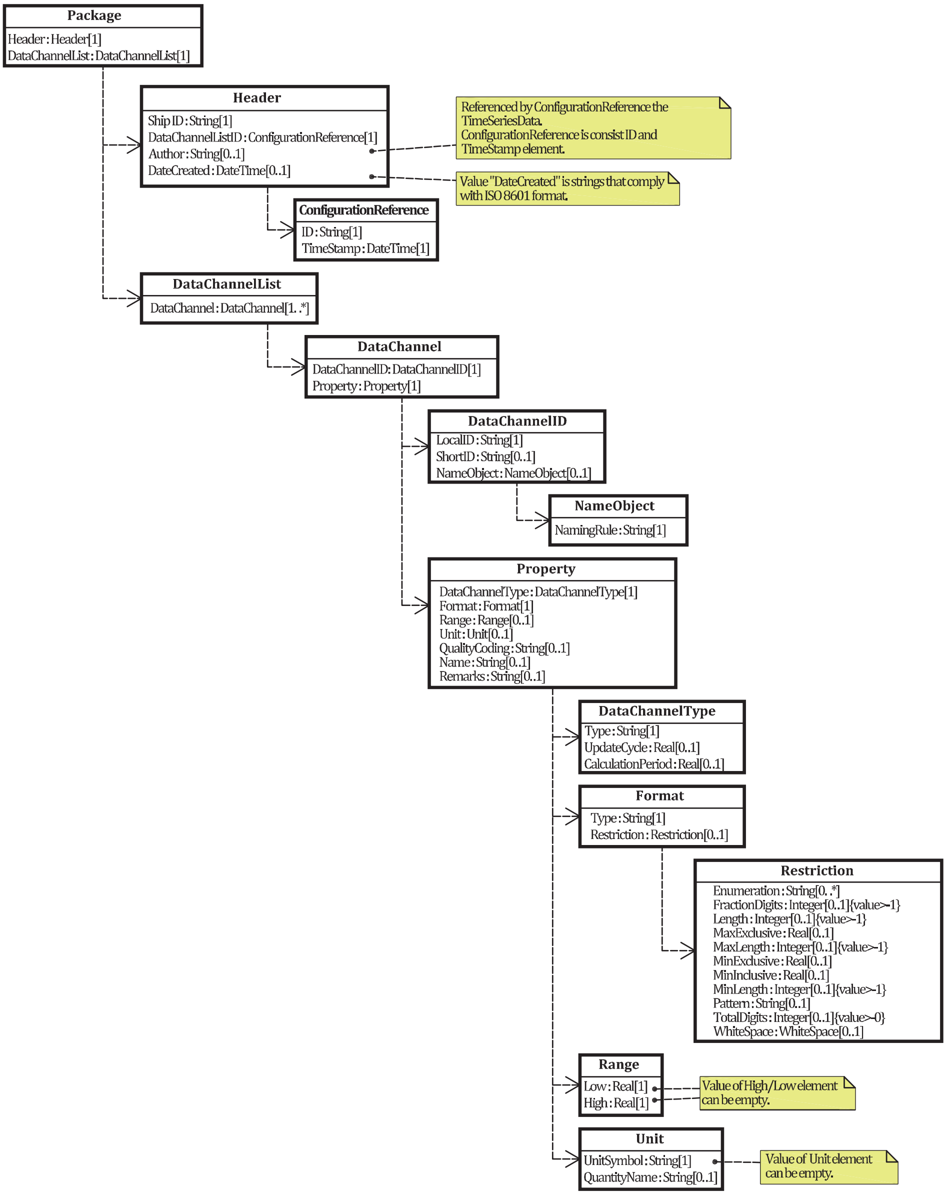


Figure 4.7: Logical structure of a data channel list package; reproduced from (ISO 2018b). The numbers between brackets indicates the number of instances contained in the data structure: [1] is one mandatory instance, [0..1] is one optional instance, and [1..*] is at least one mandatory instance.

Figure 2 — Logical structure of Data Channel List

Details of each element are described below.

a) Package structure

Name	Data type	Note	Mandatory/ Option	Max count
Header	b) Header	See b).	Mandatory	1
DataChannelList	d) DataChannelList	See d).	Mandatory	1

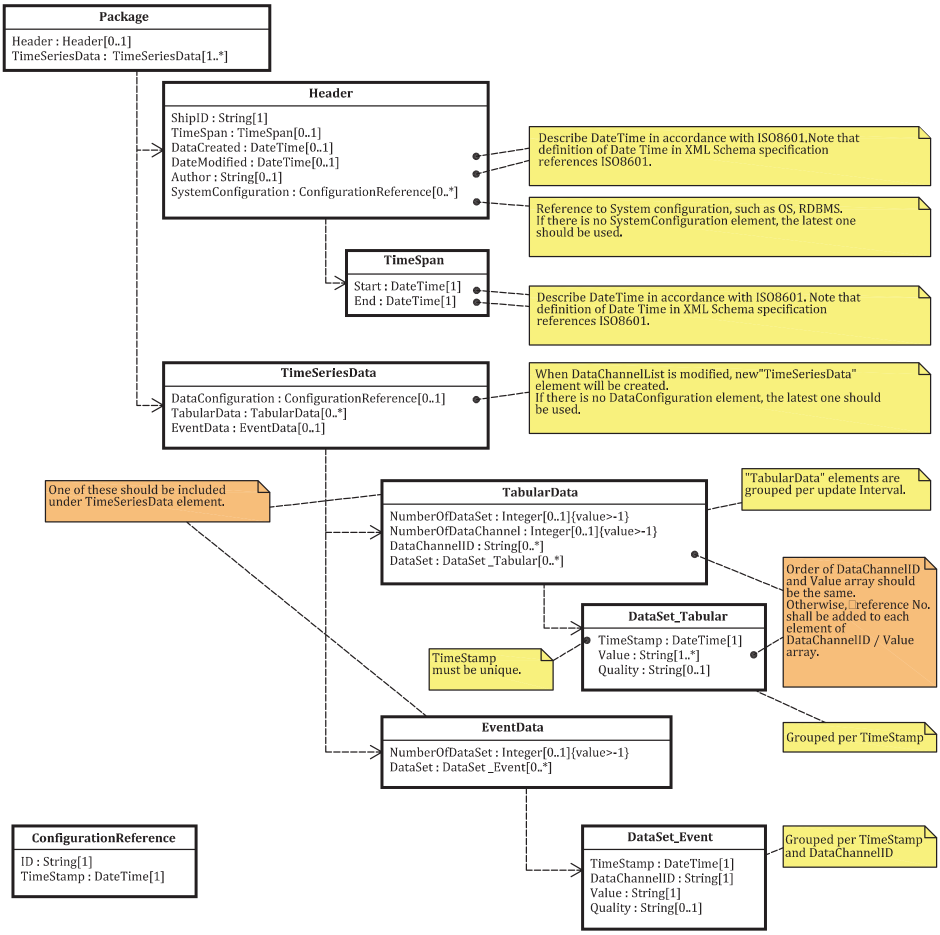


Figure 4.8: Logical structure of time series data, containing tabular and event data; reproduced from (ISO 2018b).

will store and manage data. It does not prescribe an overarching structure for data packages when storing them, avoiding imposing unnecessary restrictions on database design. Designated identification fields contained in the packages can be used to name the data according to a desired taxonomy. Annex B explains how to identify packages according to the JSMEA-MAC rule, and Annex C does the same for VIS.

4.8 Operating context

Challenges related to data on the operating context are different from those experienced with ship data. Standardisation of geographic and meteorological data is more

Figure 4 — Logical structure of Time Series Data

Details of each element are as follows.

a) Package structure

Name	Data type	Note	Mandatory/ Optional	Max count
Header	b) Header	See b). For periodic data exchange, this element can be omitted to reduce data size.	Optional	
TimeSeriesData	e) TimeSeriesData	See e). TimeSeriesData is grouped by DataChannelList editions.	Mandatory	*

disseminated and mature than that of ship data, which occurs because, among other factors, data collected from a geographic region might concern various public and private stakeholders, such as research entities, governments, and maritime companies, requiring development and adoption of standards to enable consistent exchanges among parties. However, various challenges associate with data access. The most immediate is that ship operators do not, in principle, own weather-monitoring infrastructure. Assuming that desired data are available, it is necessary to gather them from external providers, possibly several. Based on this situation, this study does not prioritise modelling of operating contexts. There is a greater amount of standardisation to be conducted inside of domains under direct control of ship stakeholders, and therefore it was felt to be more advantageous to focus on these aspects. The development of case studies relied on an ocean model intended to display current wave states. It was based on the generic visualisation in Figure 4.9, which was adapted from extant research to display regular waves based on inputs — direction, period, height, and phase difference in relation to simulation time (Chaves and Gaspar 2016). Wavelength was inferred from dispersion in relation to deep waters. In summary, the approach makes it possible to gather parameters from a measured or simulated wave state, and then use them to reproduce a corresponding visualisation. The ocean model can be modified in size or to add surfaces and textures that represent the sea bottom, terrain elevations, and city buildings when simulating regions near shore. As a consequence of supporting only regular wave states, Vessel.js does not simulate seakeeping in stochastic sea states, i.e., those states encompassing distributions of wave energy among different frequencies and described by corresponding spectra. This is certainly a relevant limitation which could be addressed in future work, both from the perspective of how to efficiently render and visualise stochastic states interactively and how to accurately simulate seakeeping with the digital twin in these circumstances.

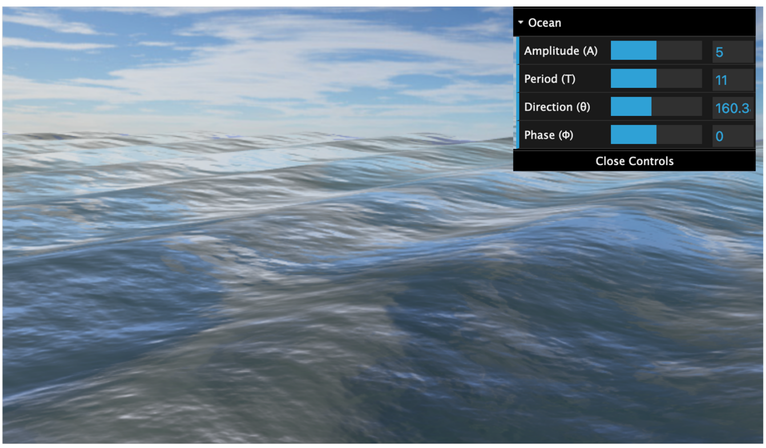


Figure 4.9: Ocean visualisation with regular wave.

4.9 Behavioural models

4.9.1 Vessel.js' general simulation template

A digital twin must use several behavioural models that account for simulation domains and types of states. This work presented two implementation approaches, which can be understood in light of Section 2.5.2's contextualisation. Chapter 5 presents a digital twin prototype which handles an experiment's states in real-time and stores them according to ISO 19848. Though qualifying as a digital twin prototype and making use shipboard data standards, it followed an ad hoc approach to modelling of behaviour and therefore does not bring contributions to that domain. On the other hand, Chapter 7 explains development and usage of Vessel.js to execute web-based simulations. This chapter presents a much more streamlined approach regarding simulation modelling; however, the examples presented in it do not qualify as digital twins due to the missing link with an operational asset and its states. This section outlines the latter approach. During Vessel.js' implementation, each behavioural model was developed as a JavaScript object following a similar structure. Listing 4.6 exemplifies the simulation template with an excerpt of a wave motion response model. It receives as arguments a ship and state objects, environmental parameters (i.e., wave characteristics and water density), and model-specific parameters (i.e., longitudinal positions on the vessel where motion should be evaluated, and a damping parameter to account for viscous effects). It then verifies whether the necessary ship states have been calculated, stores the input parameters, and defines methods to calculate simulation outputs, such as vertical motions (i.e., pitch and heave) and roll motion.

Code listing 4.6: Excerpt of the script for simulation of motion response with closed-form expressions.

```
function WaveMotion(ship, states, waveCreator, g = 9.81, rho = 1025, position = 0,
        dampPercentage = 20) {
        StateModule.call(this, ship, states);

        // if ship does not have a floating condition state
        if (typeof this.states.discrete.FloatingCondition === "undefined") {
                // calculate it
                this.setDraft();
        }

        // if ship does not have a speed state
        if (typeof this.states.discrete.Speed === "undefined") {
                // use its design speed
                this.setSpeed();
        }

        // if ship does not have a heading state
        if (typeof this.states.discrete.Heading === "undefined") {
                // point it to North
                this.setHeading();
        }
```

```
        // store input parameters as properties
        this.speedState = this.states.discrete.Speed.state;
        ...

        // specify desired outputs from simulations
        this.output = ["verticalMotion"];
}

// define methods for calculation of outputs
...
```

4.9.2 Handling of states during simulations

The simulation module handles states at runtime by interacting with the specialised object given as argument to it. This object organises states in a hierarchy of groups, first according to time frequency (i.e., discrete or continuous) and then to simulation domain (for example, buoyancy condition, stability, and motions). A given object can be provided as argument to several simulation modules, allowing them to share the same set of states. As the methods in the simulation module need to be explicitly invoked to execute calculations, it is possible to customise the update rate of different simulation states by adjusting the frequency their corresponding models are called, for example, at a certain number of time steps, at a certain time period, after a trigger condition is reached, and so on. When a method is activated, it will access the state object to collect inputs and then to update them with output results after calculations are performed. The change propagation of through stored states is managed through a cache system that indicates when a value has been updated, thus requiring recalculation of downstream (i.e., dependent) results (Fonseca 2018). Finally, for case studies which include more than one vessel, it is possible to describe each with a corresponding ship model and state object; their encapsulation features will ensure the states are assigned to the correct ship at runtime.

Chapter 5

Assessment of standards

5.1 Chapter's aim

The first case study is a digital twin of a wave basin experiment at the Numerical Offshore Tank (TPN, from the acronym in Portuguese) at the University of São Paulo. The study was planned as a first assessment of selected standards, with the purpose of identifying aspects in which standards were deemed satisfactory, and limitations that could be addressed during later stages of the project. Development followed a bottom-up approach, in which existing instrumentation and control systems at the wave basin were integrated into a functional digital twin. This chapter and its accompanying paper address the second pair of research questions.

5.1.1 Paper II

Ícaro Aragão Fonseca et al. (Jan. 2022). 'A Standards-Based Digital Twin of an Experiment with a Scale Model Ship'. In: *Computer-Aided Design* 145. DOI: 10.1016/j.cad.2021.103191.

Abstract: Digital twin ships support the operation of a complex asset using a comprehensive simulation model fed with live data collected from the asset and its context. The ambitious scope of a digital twin project can be addressed better using gradual development that recognises the digital services' levels of maturity, including simple observation, prediction of behaviour, and decisions based on simulation results. Several challenges to development of digital twin ships are evident, especially interoperability of models and data generated across stakeholders who use disparate software systems. We propose use of existing standards and web technologies to asset representation and sensor readings in digital twin ships is proposed, providing a framework that can be linked to services such as visualisations and simulations of vessel behaviours. A case study applies the standards to an experiment that involved a scale model ship equipped with a dynamic positioning system in waves. The digital twin prototype illustrated the

capability of mirroring and controlling the model's position in real-time, and predicting motion responses across wave conditions. Thus, it closes the loop between test and design in the lifecycle by allowing validation of results in comparison to empirical data during operation. Future research should extend such standardisation to full-scale experiments that use greater numbers of ship components and sensor logs.

Declaration of co-authorship: I wrote the paper, collected and prepared data, and developed the digital twin. Henrique Gaspar advised general development, suggested improvements and inclusion of sections and figures, and arranged collaboration with TPN. Pedro de Mello provided access to the experimental data, hardware interfaces, and wave basin setup, and operations. Humberto Sasaki prepared the external control system setup for integration with the digital twin, and helped with conducting the experiments.

5.2 Physical setup

5.2.1 Wave basin

The wave basin (Figure 5.1) measures 14 meters on each side and is 4.1 meters deep. The tank boundary contains flaps that can generate regular and irregular sea states. The flaps also operate as absorbers to minimise wave reflections, avoiding interference with specified wave characteristics. The tank is equipped with a commercial installation that measures water elevation using a few probes installed in the tank. A few simplifications were used during experiments to simplify digital twin development. The wave condition was always regular and taken from the same known direction, which avoided having to identify wave directions with the simulation. When calculating wave characteristics from water elevation, the inputs from a single probe floating close to the model ship were sufficient to estimate desired parameters.

Figure 5.1: Wave basin at TPN (Mello 2012).

5.2.2 Model ship

The model ship is a Platform Supply Vessel (PSV) at 1:70 scale, measuring 1.24 meters in length, 0.345 of beam, and 0.082 of draught (Figure 5.2). It is actuated with a Dynamic Positioning (DP) system that comprises one bow tunnel thruster and two stern thrusters capable of rotating in azimuth, relative to the model hull. The DP system can be controlled remotely over radio frequency. During experiments, the azimuth angles of both stern thrusters were locked to neutral position for simplification of the mathematical control problem to be solved (Ianagui 2019). A commercial installation based on stereoscopic cameras reads the PSV model motions on waves and sends the results to a DP control software developed by TPN. Based on the received motion readings, the software controls thrusters' rotations to achieve and maintain a desired positioning set point, which is defined in terms of planar (i.e., x, y) coordinates and yaw (i.e., z) rotation.

Figure 5.2: PSV scale model used during the experiments (Ianagui 2019).

5.3 Objectives

The digital twin provided an interface to monitor and operate the PSV model during experiments. In terms of monitoring, it displayed an updated 3D visualisation of the experiment with wave conditions, ship positions, and thruster rotations. In terms of operation, it allowed a user to define the set points that should then be sent to the control module so that the PSV could be manoeuvred to a desired position. In addition to the main objectives, some complementary functionalities were explored, such as validation of motion responses in comparison to experimental measurements and minimisation of motion response.

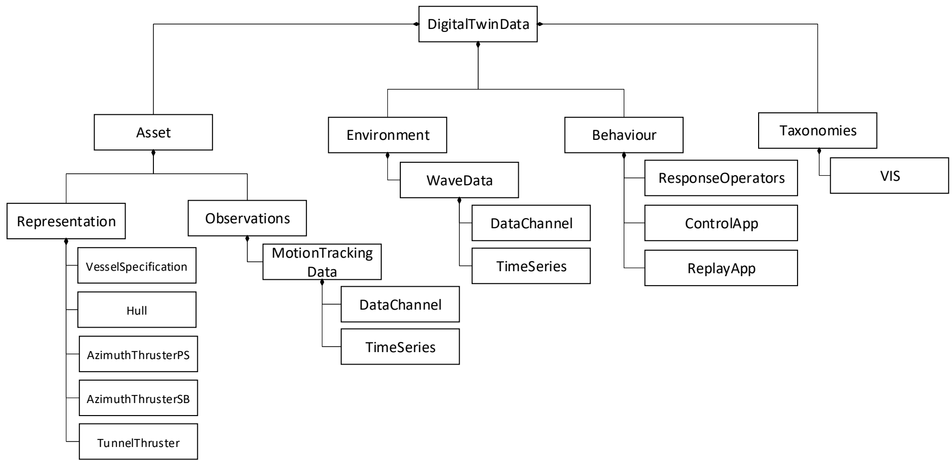


Figure 5.3: Data diagram of the digital twin prepared for the case study (Fonseca et al. 2022).

5.4 Modelling

5.4.1 Outline

Figure 5.3 shows the digital twin's content. It organises the data and applications required to execute the digital twin according to the categories discussed earlier. The following sections follow the same overall structure, presenting the modelling approximately in the order it appears in the figure, from left to right: ship representation in Section 5.4.2, observations of ship states in Section 5.4.3, wave basin model and states (in this case, analogous to operation environment) in Section 5.4.4, ship station keeping behaviour in Section 5.4.5, data taxonomies in Section 5.4.6, and digital twin data integration in Section 5.4.7.

5.4.2 Ship representation

The ship representation included four components — a hull, one bow thruster, and two azimuth thrusters (Figure 5.4). Since the azimuth thrusters were identical, the ship model required three 3D model files, with one of them being used twice. The glTF standard allowed creation of azimuth thrusters as assemblies, including rotation axes for both the azimuth and propeller. Metadata were prepared in five JSON packages. The first is a ship specification with general PSV data, such as main dimensions, weight distribution, and centre of gravity:

Code listing 5.1: PSV specification.

```
{
  "Package": {
    "Remarks": "An initial draft a ship specification standard.",
```

```
    "Author": "Ícaro Fonseca",
    "ShipID": "PSVH",
    "MainDimensions": {
      "LOA": 1.24,
      "B": 0.3450,
      "DesignDraft": 0.0820,
      "UnitSymbol": "m"
    },
    "Weight": {
      "Lightweight": 22.80,
      "UnitSymbol": "kg"
    },
    "CG": {
      "XCG": 0.0136,
      "YCG": 0.0000,
      "ZCG": 0.0826,
      "UnitSymbol": "m",
      "Remarks": "Referece from station 5, bottom tangent, central longtd. plane."
    },
    "AssetID": {
      "LocalID": "/dnv_vis/071",
      "ShortID": "VesselSpecification.json",
      "NameObject": {
        "NamingRule": "dnv_vis",
        "vis:FunctionLeaf": "071 Vessel specification"
      }
    }
  }
}
```


Besides one package for vessel specification, the four others were one for each PSV component. For example, the package for the starboard azimuth thruster reads:

Code listing 5.2: Metadata about the starboard azimuthal.


```
{
  "Package": {
    "Remarks": "Azimuth thruster starboard.",
    "Author": "TPN employee",
    "Topology": {
      "Visualization": {
        "GLB": "AZ.glb"
      },
      "Position": {
        "x": -600,
        "y": -60,
        "z": 65
      },
      "UnitSymbol": "mm"
    },
    "AssetID": {
      "LocalID": "/dnv_vis/433.1(mm)",
      "ShortID": "AzimuthThrusterSB.json",
      "NameObject": {
```

```
        "NamingRule": "dnv_vis",
        "vis:FunctionLeaf": "433.1 Propulsion thruster, starboard",
        "vis:Component": "[C322] Propulsion thruster, azimuth"
      }
    }
  }
}
```

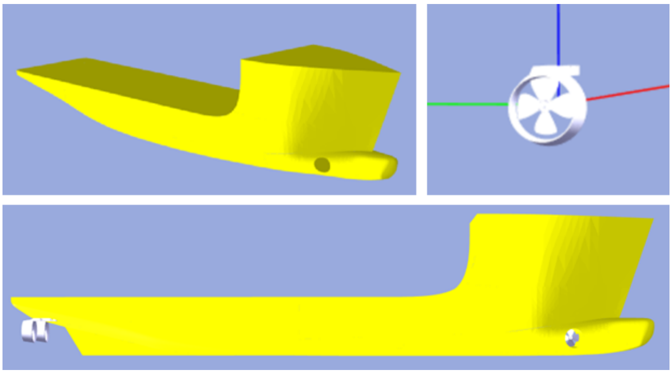


Figure 5.4: PSV visualisation based on hull, two azimuth thrusters, and a bow tunnel thruster. The local axis shows the propeller rotation centre on the thruster assembly.

5.4.3 Ship states

Both metadata templates (i.e., data channel list and time series data) were prepared according to ISO 19848 to store sensor data generated during simulations (ISO 2018b). All vessel states were exchanged with the DP module over the same connection, so they were organised in a single data channel list, encompassing: instantaneous position at 6 degrees of freedom, instantaneous propeller rotation rate per minute, and set point commands sent to the DP system at 3 degrees of freedom (i.e., surge, sway, and yaw). A template for time series data was created to store sensor logs. After the experiments concluded, the sensor logs were collected and converted to the standardised template. In the same manner as in Listings 5.1 and 5.2, sensor logs were identified using VIS according to tags inserted in the JSON metadata. Although the digital twin measured and displayed propeller rotations during experiments, it did not store them permanently, and so it was not possible to rewind them later.

5.4.4 Wave basin model and states

The operating context was based on the ocean model discussed in Section 4.8. Its development can be described in three stages. During the first, the visualisation was scaled to the length and breadth of the wave basin, in proportion to the PSV model. This produced a static model of the wave basin, but it was still necessary to develop it further to display the wave state that was occurring in the wave basin in near real-time. During the second stage, an algorithm to receive readings of wave elevations as they were measured by a buoy floating near the PSV model was written. The algorithm followed a rudimentary approach, during which it kept a record of all crests and valleys found in the water elevation signal the instant they were encountered, and then updated the wave parameters once significant changes were detected. During the third stage, the waves' heights, periods, and phases in the basin visualisation were linked to the monitoring algorithm, so that the digital twin could display the current wave state as soon as it was identified. The digital twin stored wave data with an approach similar to the one used for ship states (i.e., describing the data channel and time series data using ISO templates). Since wave parameters were not converted from the raw water elevation log that was collected from the tank, but from post-processed results obtained from the monitoring algorithm, the data channel used as a calculated value. Despite ISO 19848 not having been designed to store wave data, it was sufficient to model waves during the experiment. Incorporating new standards for environmental data would have added unnecessary complexity to the study, while being unable to evaluate their effectiveness at handling real metocean data, including factors such as weather condition variations between operating regions.

5.4.5 Behavioural models

The DP control experiment was conducted using proprietary software suite Matlab R2013b (MathWorks 2013), so it was unnecessary to develop behavioural models for the digital twin. However, the digital twin required access to expected motion responses during operation to provide two functionalities. The first one is validation, whereby the digital twin compares the motion response results from the computational analyses with the ones measured during experiments in real-time. The second one is the manoeuvring decision support, whereby the digital twin automatically manoeuvres the vessel toward the heading angle which minimises response in a desired motion mode. This is accomplished with a rudimentary technique where the algorithm searches the heading angle which minimises the response and then manoeuvres the PSV model to the selected position. Paper II describes both functionalities and their corresponding algorithms in detail. The results from motion response analysis executed previously at TPN were collected, using the Wave Analysis Massachusetts Institute of Technology (Wamit) software based on the boundary element method (Wamit 2022). Results were converted to a JSON file, sorting Response Amplitude Operators (RAOs) and phases according to motion modes, wave periods, and incidence angles in degrees (Figure 5.5). Metadata included documentation that allowed interpretation of listed results. Two web applications were developed for the digital twin. The first was a dashboard to monitor and control the PSV during experiments, which included a 3D visualisation, 2D charts, a text list of measured values, and slider inputs for DP control. The second was a replay application that displayed a video of the experiment, in addition to the digital twin visualisation and motion response charts, updated synchronously.

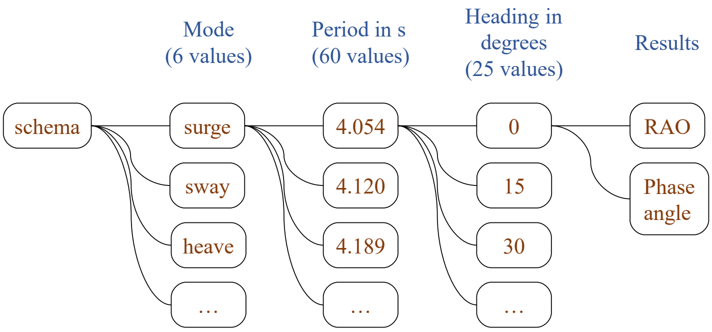


Figure 5.5: Diagram of the JSON structure storing RAOs.

5.4.6 Data taxonomy

A VIS schema was created to map the digital twin data listed in its scope (Listing 5.3). In practice, this means the ship representation in Section 5.4.2 and sensor logs in Section 5.4.3). Despite the variety of data being small in comparison to an extensive digital twin, and that the schema itself had little practical utility during this study, this allowed a first experience at applying VIS and anticipation of some of its scaling limitations for future research (see Section 5.6).

Code listing 5.3: Excerpt from the VIS schema written for the digital twin data.

```
{
  "ID": "433",
  "CommonName": "Propulsion thruster arrangement",
  "Children": {
    "433.1": {
      "ID": "433.1",
      "CommonName": "Propulsion thruster",
      "Component": "[C322] Propulsion thruster, azimuth",
      "Path": {
        "Asset": [
          "AzimuthThrusterPS.json",
          "AzimuthThrusterSB.json"
        ],
        "MeasuredStates": []
      }
    }
  }
}
```

5.4.7 Integration

Figure 5.6 shows the integration of digital twin systems during the experiments. The DP control system (element 1) and system that measured water elevation (2) were already installed and operational at TPN. The three remaining elements were developed for the case study. The web application (4) shows the graphical user interface and includes a few scripts for retrieving RAOs according to current wave states, accessing digital twin data from a local folder (5). The communication bridge in Python (3) establishes connections between the web application and the instrumentation systems, receiving messages from the measuring systems in the User Datagram Protocol (UDP) and passing them to the web application through WebSocket, and vice versa. The bridge converts the water elevation signal to regular wave parameters to avoid overloading the connection with the web client. The architecture can be greatly simplified during experiment

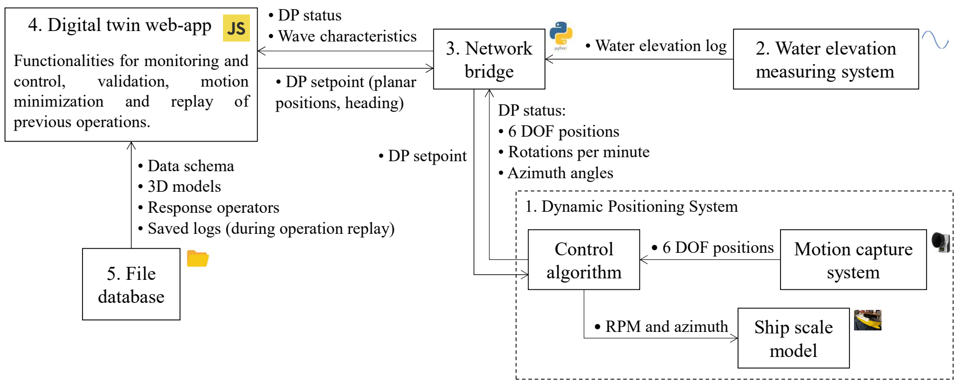


Figure 5.6: Architecture for experiment streaming and control (Fonseca et al. 2022).

5.5 Results

The functionalities of remote monitoring and control offered a few advantages during the experiments (Figure 5.7). It provided a central terminal for PSV operation with visual clues for identifying if it was well-positioned or drifting inside the tank, and allowed verification of propulsion thrusters during the experiments, even if they were submerged and visible. The additional features allow comparisons of measured responses against expected responses from simulations, and minimisation of amplitude responses for an uncoupled motion mode regarding wave incidence angles. These additional functionalities are still in a preliminary phase, leaving room for development. For example, instead of displaying motions as comparisons of time series charts, the digital twin could present them as a denser, more informative visualisation, showing difference bars for all six motion modes simultaneously. Similarly, the minimisation algorithm could be improved to account for coupled motions, so that it would be of greater assistance during operation.

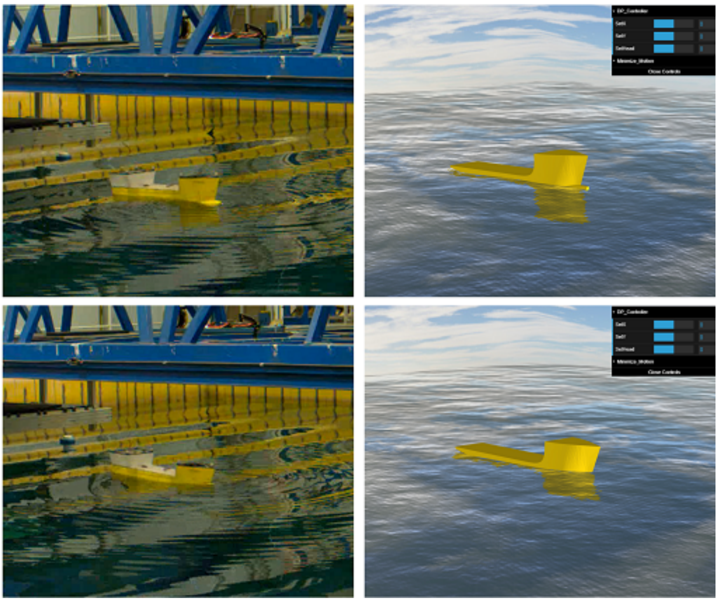


Figure 5.7: Monitoring functionality during the experiment (Fonseca et al. 2022).

5.6 Discussion

Paper II includes an extended discussion on results and the open source approach. This section adds a few comments, starting with the digital twin's potential for scaling. The bottom-up approach to digital twin development was sufficient for this case study as a proof of concept, but it occasionally revealed challenges to use during the experiments. Since the digital twin was constituted as a distributed system, it must propagate states concurrently through the disparate components seen in Figure 5.6. This arrangement requires users to understand how each system interacts to operate the digital twin properly. This complicates troubleshooting of undesired outcomes, since errors must be traced to one of various subsystems, or to connections among them. A first step toward minimising this issue might be to approach the digital twin infrastructure as a design problem to be solved, with the goal of limiting complexity when addressing digital twin requirements with new software or hardware. This measure is not expected to entirely solve the problem, because the digital twin is fundamentally a complex system distributed through onshore and offshore locations. Thus, a second row of measures might be designing its hardware and software components for robust and traceable connections. They should be developed with interfaces for remote operation (for example, APIs) since early design, rather than offering *ad hoc* interfaces built on top of an existing system. They should also provide features to remotely monitor relevant statuses (for example, “turned on and active,” “busy”, and “not responding”), so that an operator can easily identify when a component is out of service.[1] The limitations of the adopted standards are taken as motivation for the following chapter. Handling 3D models as Listing 5.2 proposes was largely adequate to digital twin modelling, but it lacks refinement. More importantly, it should provide a mechanism to reuse 3D models, rather than the current approach, during which the digital twin must access and load the file every time it appears in the ship representation. This led to the adoption of Vessel.js during the case studies to take advantage of its base/derived object systems, as Section 4.6 discusses. The migration required two minor adjustments to Vessel.js — linking it to the glTF loader and including fields to specify physical units in the ship specification. The case study represented the first attempt to apply VIS to a digital twin in practice. VIS is well-documented on the web (DNV 2022) and is compatible with the proposed tag system, but it lacks a formal template or data structure that can be readily linked to a digital twin. Thus, a standardised method of handling taxonomies for ship and sensor data in digital twins was considered to be necessary.

[1] As part of its efforts toward standardising sensor names, a revision of VIS published after this research had been developed defines "state" labels which can be used toward this propose, though only in binary form (for example, on/off, running/not.running) (Låg, Vindøy and Ramsrud 2021).

Chapter 6

Ship representation and taxonomies

6.1 Chapter's aim

This chapter discusses linking ship data to data taxonomies, with a focus on ship representation. A proposed framework reconciles a few central issues during data management in the ship digital twin and digital thread — the necessity of modelling the ship using multiple perspectives, depending on lifecycle phase, stakeholders' ability to link ship data to chosen taxonomies, and provision of data structures to support data reuse and exchanges within this scope. The case study uses the framework with previously available data from NTNU's Research Vessel (R/V) Gunnerus. This chapter and its accompanying paper partially address the third pair of research questions.

6.1.1 Paper III

Ícaro A. Fonseca and Henrique M. Gaspar (2022). 'An Open Framework For Data Taxonomies In Digital Twin Ships'. In: *Submitted to International Journal of Maritime Engineering; re-submission being prepared after first round of reviewers' comments.*

Abstract: 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 software suites. This study proposes an open framework for organising ship and sensor data in digital twins. The framework is based on principles of product data management, allowing use of taxonomies suitable across stages of the lifecycle up to operation, when the digital twin is deployed. The work implements the framework using a web-based approach that focuses on early design and operation phases, then apply it to a case study with a research vessel. The case study outlines browsing of ship model,

examples of behaviour simulations, and future monitoring of sensor measurements. The discussion explains how the framework might be expanded and the challenges when applying it to existing engineering practice.

Declaration of co-authorship: I wrote the paper, proposed the framework, and developed the web applications to illustrate it. When developing the apps, I reused Gunnerus 3D models developed previously by others. Henrique Gaspar advised general development, and suggested improvements and inclusion of sections and figures.

6.2 Taxonomies of ship data

6.2.1 Taxonomies as enablers of a digital thread

Management of ship data (i.e., ship representation and sensor logs) throughout the lifecycle relates closely to the degree of hierarchical detailing and the domain view chosen to model a vessel (Hamer and Lepoeter 1996; Pal 2015).

Figure 6.1 illustrates some of these key views, organising lifecycle stages in early stage design — which comprises feasibility, contract and concept design —, detail design, construction, commission, operation, and decommission. During early stage design, traditional cargo vessels are commonly divided into lightweight and deadweight groups. These groups might be further decomposed into subgroups. Lightweights might, for example, be further divided into hull, machinery, deckhouse, and outfitting subgroups. However, design methods regarding complex vessels commonly focus on space utilisation by the systems required to fulfil a ship's purpose. This is the case with the Design Building Block approach, which has been applied consistently to design of complex civilian and combatant vessels (Andrews and Dicks 1997; Andrews and Pawling 2008). Andrews (2018) gives its most updated and comprehensive exposition in the context of early stage ship design. The System Based Ship Design (SBSD) is another example of a design method that uses this view. The approach was originally derived for passenger ships Levander 1991, though over time, it has been applied across several types of vessels, including cargo tankers and offshore support vessels (Levander 2012). In the following detail design phase, the vessel's components shall be defined with sufficient specificity for effective construction. This detailing is most often accompanied by a deepening in the hierarchical schema used to model the vessel, reflected on an increase in the number of its constituent levels. Later, a shipyard will choose a view that supports efficient ship construction, which might be a taxonomy of the project in terms of block zones, groups of components that share production process, and identification of suppliers for procurement purposes. During the following two stages, the vessel and digital twin are tested and operated, bringing the focus to the vessel's systems functionalities. These stages relate to taxonomies discussed in Chapter 4. During decommissioning, it is useful to identify materials contained inside of a vessel for handling and recycling (Andrade, Monteiro and Gaspar 2015).

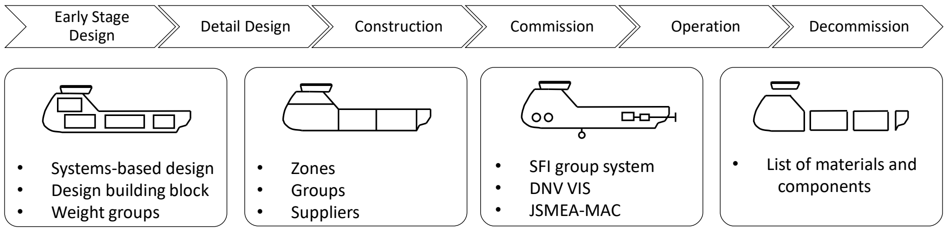


Figure 6.1: Example taxonomies appropriate to various lifecycle stages (Fonseca and Gaspar 2022).

6.2.2 Standardisation framework

Considerable variety exists among taxonomies that are useful when identifying and managing ship data, which is true even if the focus is limited to a particular stage of the lifecycle, with options catering to multiple engineering processes and geographic regions. This points to a need to provide stakeholders with flexible tools that describe a ship's data according to their taxonomies. In practice, stakeholders should be able to map ship data to hierarchical schema that might be useful. This problem is addressed by separating managed data from the schema used to organise them. In this framework, each component in the ship model and each sensor log collected from the vessel is attached to one identification hashtag for each taxonomy to which it should be mapped. Figure 6.2 exemplifies how it can be applied to two situations. The top section shows how, during early stage design, a fuel tank in a high level ship model can be linked to the appropriate field in a SBSD schema. Conversely, the bottom section shows how a reduction gear might be identified according to VIS during detailing stage, when the SBSD taxonomy does not provide an adequate level of granularity to describe individual vessel components. In the same manner, the VIS Equipment library could be used to further divide the gear box in terms of its internal parts and thus identify them in monitoring dashboards or when managing maintenance activities. Corresponding taxonomies are stored as JSON files, listing all entries in hierarchical form. A hash map that links each entry in the taxonomy to a corresponding element in the digital twin is written out, allowing quick transition from schema to content. Both taxonomies exemplified adopt a functional view of the ship because, as Section 4.5.1 discusses, this is the most suitable choice for digital twins. The same principles could be applied to incorporate, for example, weight or zone-driven views of the vessel. More generally, this mechanism provides flexibility while allowing taxonomies to be maintained and distributed independently from project-specific data.

6.3 Physical setup

The physical setup for the case study was Norwegian University of Science and Technology (NTNU)'s R/V Gunnerus (Figure 6.3). The vessel is used for scientific research across disciplines, with tasks such as collecting and analysing biological and geological

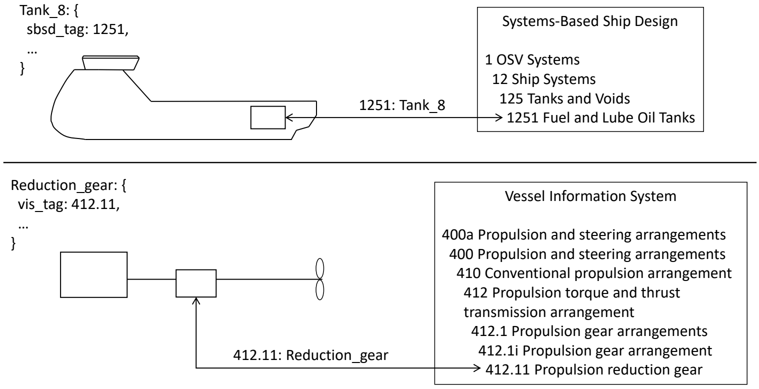


Figure 6.2: A framework that links elements in a ship model to two functional taxonomies: SBSD and VIS (Fonseca and Gaspar 2022).

samples. As part of digitalisation research at NTNU, Gunnerus data, such as detailed 3D models, document archives, and sample sensor logs, are collected in a common file folder. Gunnerus is also outfitted with a comprehensive instrumentation setup that registers logs from several of its systems, streaming them wirelessly to a central server.

Figure 6.3: Starboard view of R/V Gunnerus. Photo by Fredrik Skoglund.

6.4 Objectives

This case study creates a digital model of Gunnerus based on the proposed framework. It combines methods and standards to illustrate how a ship representation can be used to support various stages of the lifecycle, focusing specifically on early design and operation. This chapter gives a step further compared to the previous one by incorporating a full scale — instead of a scale model — ship into the case study. The case study models only ship representation, but it outlines how the same method can be linked to sensor logs and provide a unified monitoring hub for a vessel, setting a basis for further development of a web-based, digital-twin Gunnerus.

6.5 Modelling

6.5.1 Outline

The ship representation includes two views of the same Gunnerus vessel. One offers an architectural model of Gunnerus, thus being suitable for use with design methods such as SBSD or the DBB approach. The second is a detailed model of physical constitution that is suitable for operation. The case study does not yet link the Gunnerus representation to sensor streams, but it is discussed how they can be integrated to the proposed framework using existing standards and transmission infrastructures.

6.5.2 Data taxonomies

Two taxonomies are used to accomplish the objectives — SBSD and VIS for design and operation, respectively. Since VIS includes a comprehensive list of vessel functions and components. It 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. VIS' scope, as well as that of JSMEA-MAC and SFI, is limited to civilian vessels. For an example of taxonomy suitable to military vessels, the reader might refer to the Expanded Ship Work Breakdown Structure (ESWBS), used by the United States Navy (Garzke, Cimino and Yoder 2014). SBSD required adjustment before being applied to a research vessel. SBSD divides a vessel into spaces assigned to various systems, classified into two groups — task-related systems and ship systems. The first relates directly to execution of a mission, and the second accounts for the internal infrastructure that supports vessel operation. These notions are sufficiently general to allow application to any vessel design. While there exists literature about design of research vessels (Washio et al. 1994), the candidate did not identify a case study conceived specifically through the SBSD method. This led to the necessity of consulting existing Gunnerus specifications and arrangements to identify constituent systems in a fashion that allowed mapping to the SBSD taxonomy, as Figure 6.4 summarises. Task-related systems are those used for facilities that support scientific research (for example, offices, analysis rooms, and a remotely operated vehicle and its deployment A-frame) and cargo handling systems (for example, cargo hold, deck area, and crane). Ship systems are similar to those found in other vessels of similar size — ship structure, outfitting, accommodation, tanks, and voids. The next step was preparing the taxonomies for use as digital twin schemas, conducted by converting them to a JSON data structure, as Listing 6.1 exemplifies. The schema was prepared manually for use with the visualisation app; the data structure could be organised slightly differently depending on the objective, for example, exchange with stakeholders or use as database schema. Since only a limited excerpt of VIS was used during the case study, it was not required to convert the entire schema to JSON. In the future, it would be possible to create a script to parse the Excel table in Figure 4.4 and convert it to a hierarchical JSON structure automatically.

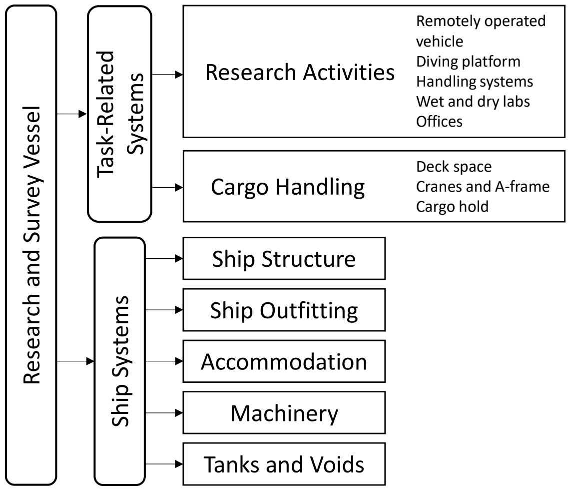


Figure 6.4: SBSD function structure applied to R/V Gunnerus (Fonseca and Gaspar 2022).

Code listing 6.1: Excerpt from the VIS schema in JSON prepared for use with the Gunnerus visualisation app.

```
let VIS_schema = [
  {
    "id": "110", "text": "110 Ship structure", "children": [
      {
        "id": "111", "text": "111 Ship hull structure", "children": [
          { "id": "111.1", "text": "111.1 Decks" },
          { "id": "111.2", "text": "111.2 Transverse bulkheads" },
          { "id": "111.41", "text": "111.41 Single skin sides" },
          { "id": "111.71", "text": "111.71 Single bottom" }]
      }
    ]
  },
  ...
];
```

6.5.3 Ship representation

The Gunnerus representation was built using the Vessel.js library (Oliveira 2022), allowing the advantage of using the base and derived objects mechanism in Listings 4.4 and 4.5), instead of using the templates discussed in the previous chapter (Listings 5.1, 5.2). Figure 6.5[1] shows the design-oriented model of Gunnerus, emphasising vessel spatial arrangements. It comprised compartmentation, a simplified parametric model of a structural arrangement, and a hull visualisation drawn from a table of offsets. The hull shape, based on a table of offsets model, allows direct reuse of analyses and simulations in the Vessel.js library (for example, calculation of ship buoyancy conditions based on a table of offsets), discussed in the following chapter. This approach was simpler than adapting simulations in the library to work with a detailed hull geometry at the pretext of gaining a small precision margin. Vessel.js usually calculates vessel lightweight by adding weights assigned to each derived object and structural component to an estimation of hull weight based on semi-empirical formulas. However, since Gunnerus had already been built and tested, it is simpler and more accurate to assign the total lightweight and CG to the entire model, instead of relying on bottom-up estimations of systems' weights.

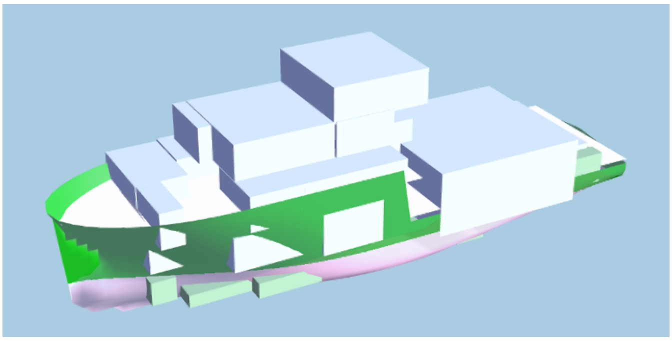


Figure 6.5: Compartmentation model of Gunnerus. Extracted from (Oliveira 2022).

The operational ship model is based on a glTF visualisation, which was available at NTNU (Figure 6.6). The file is adequate for a digital twin because it shows the vessel's external appearance with detail. However, it needed to be complemented with internal parts. The digital twin would ideally include 3D models of all vessel components. In practice, many components in Gunnerus are not linked to relevant instrumentation or digital twin states, and do not relate directly to conventional simulations of ship operation (for example, furniture in accommodations and offices, computers, and research equipment.[2] Thus, they were not included in the Gunnerus model for simplification of the modelling and to focus instead on including machinery and cargo equipment.

[1]Both the conceptual and operational 3D models of R/V Gunnerus were developed by different authors prior to the development of this case study. When the case study was being conceived, it was relatively straightforward to leverage these resources to test the framework, compared to developing news vessel models specifically for that purpose.

[2]Such accommodation and office arrangements can be used to simulate less-conventional operational aspects, such as space utilisation and work ergonomics. This study sets a foundation for inclusion of such components in the future, if necessary.

Figure 6.6: Detailed model of Gunerus. Credit: Department of Ocean Operations and Civil Engineering, NTNU.

6.5.4 Assessment of the potential to link the digital twin to sensor logs

ISO 19848 includes fields for identification tags according to a desired naming rule as an explicit part of its design (ISO 2018b). When the standard is adopted, this feature allows extension of the framework used to map ship components to taxonomies (Figure 6.2) to include sensor streams and logs. Conversely, in cases when it is not, the feasibility of implementing the framework becomes much more erratic, as it depends on the ability of the templates provided by each different vendor to accommodate the necessary identification tags. This is the case with Gunnerus, where collected sensor logs are transmitted and stored in custom formats, defined by the instrumentation manufacturer. Several logs are stored in a custom CSV template, in which the only indication of meaning are the variable names themselves, thus lacking metadata providing contextualisation in terms, for example, of physical meaning and units. It is possible to create ISO 19848 schemas for existing data channels inside the vessel, which is already sufficient to allow interpretation of sensor readings based on the metadata associated with them. As a step toward greater standardisation, logs stored in files and databases could be converted to tabular data packages as ISO defines. Infrastructural challenges associated with linking sensor streams to a web-based monitoring hub also need to be considered. A setup exists to stream sensor logs from Gunnerus to a central server, which is able to repass those streams to clients over the Message Queuing Telemetry Transport (MQTT) protocol. MQTT is a bi-directional protocol that allows network communications among devices that are connected to the internet of things (ISO/IEC 2016). Web browsers do not support this protocol, but open source brokers are able to convert between MQTT and WebSocket in both directions, thus linking to a web-based digital twin's front-end.

6.5.5 Integration

The framework was used to develop an interface that links the Gunnerus representation, with design and operational models, and their respective taxonomies to a browsing interface that allows users to browse and select parts of the ship model. Tags stored in the asset metadata are used to map them to desired taxonomies. Listing 6.2 shows an excerpt of a hash map that links the operational model of Gunnerus to VIS. The map links entries in the taxonomy to components in the ship model, and to the label displayed to a user in the GUI. The same data structure could be expanded to link data channels that contain sensor logs. It is possible to adjust granularity (i.e., hierarchical detailing) during implementation, depending on a stakeholders' needs. In this example, some entries in VIS linked to various derived objects that belong to the same group (for example, three bulkheads are mapped to the same node). An alternative is to split them into individual entries for greater granularity.

Code listing 6.2: Excerpt of the hash map that links numerical VIS tags to visualisation labels and names of components in the 3D model.

```
let hash_map = {
  "111.1": ["decks", "Deck"],
  "111.2": ["bulkheads", "AB", "B23", "FB"],
  "111.41": ["hull side", "Hull_0", "Hull_1", "Hull_3"],
  "111.71": ["hull bottom", "Hull_2"],
  "112": ["superstructure", "Sup_Struct"],
  "331.11": ["frame", "SH_GROUP_A_FRAME_ROV_HANGAR"],
  "411.1": ["engines", "NOGVA_Scania_bk_l", "NOGVA_Scania_bk_c",
    "NOGVA_Scania_bk_r"],
  "413.1": ["propeller", "PTS_Propeller", "PTS_PropellerFrame",
    "STB_PropellerFrame", "STB_Propeller"],
  "433.1": ["thruster", "Thruster_Front"]
};
```

6.6 Results

The Gunnerus monitoring dashboard allows users to browse and access both Gunnerus views (Figures 6.7 and 6.8). A hierarchical tree on the left-hand side allows users to filter systems according to corresponding taxonomies. The dashboard is intended to set the basis for gradual content additions and eventual linking to sensor streams for ship monitoring.

6.7 Discussion

Paper III (Fonseca and Gaspar 2022) discusses the importance of proposing methods to allow systematic progression between views used in the lifecycle. This requirement translates to accompanying processes used to model the vessel, especially since the principles discussed here are proposed as a path to standardisation. To illustrate this,

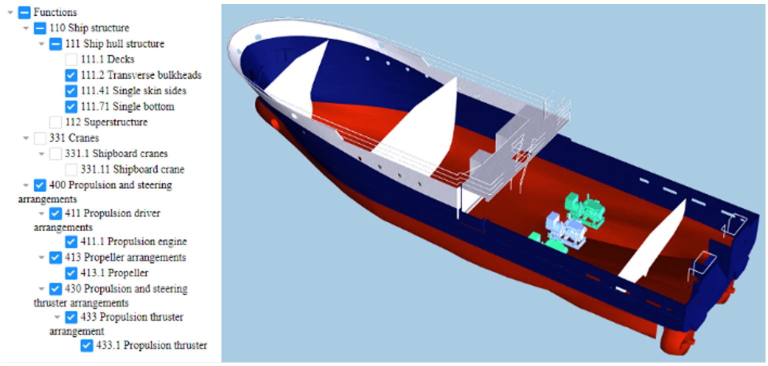


Figure 6.7: A screenshot of the operational Gunnerus dashboard with the VIS taxonomy (Fonseca and Gaspar 2022).

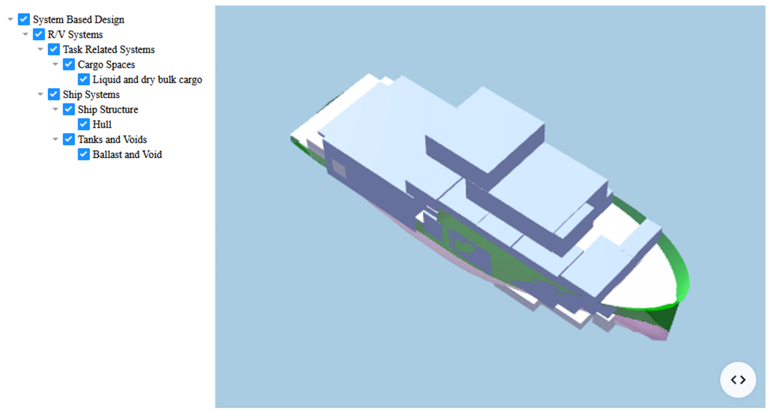


Figure 6.8: A screenshot of the design Gunnerus dashboard with SBSD (Fonseca and Gaspar 2022).

consider the problem of quantifying a ship's weight based on existing digital models. Markers should be defined to indicate when the lightweight is an estimation, or an accurate value measured from the ship as built. As the model evolves from an architectural to a detailed view, there should be clear practices that specify whether weights were assigned to the system blocks, an individual component, or the ship as a whole, with a single value representing the total lightweight, as was done during this case study. This decision might vary depending on stakeholder conveniences, but it is important to plan them to avoid errors due to ambiguities or poor data quality. A final issue is the necessity of planning a systematic way to use the ship representation as an input to the behavioural models in a digital twin system, a topic addressed in the following chapter.

Chapter 7

Simulations of behaviour

7.1 Chapter's aim

The last digital twin aspect that this study assesses is simulation of behaviour, with the purpose of contributing to a flexible approach in which modules that account for domains of behaviour can be linked to the digital twin data (i.e., ship representation, environment, and states) and then used as input to perform desired simulations. This approach enables flexibility during creation of digital twins by providing customisation with select simulation modules and gradual addition of behavioural domains as a user desires. It can be contrasted with closed solutions that prescribe a range of available features from the beginning, thus precluding alternative configurations. This chapter and both its accompanying papers continue to address the third pair of research questions.

7.1.1 Paper IV

Sergi Escamilla i Miquel et al. (July 2020). 'An open-source library for hydrodynamic simulation of marine structures'. In: *Marine Systems & Ocean Technology* 15.3, pp. 160–174. DOI: `10.1007/s40868-020-00083-3`.

Abstract: The work focuses on an open and collaborative approach for hydrodynamic simulations of multibody operations. It builds on Vessel.js, an existing web-based ship design library, by modelling the interaction between entities and creating multibody models able to output different responses. To develop the cases here studied, the simulations are decomposed into single elements to understand their behaviour separately before making them interact with other elements to create a multibody simulation. In the process, different hydrodynamic models are used to analyse the bodies according to the requirements of the simulations and the needed level of complexity. The simulations are coded in JavaScript and visualised in a web environment, with the option of using external hydrodynamic analyses, which in this work were exemplified using a commercial software that adopts the linear potential wave theory. The paper concludes

with a discussion about future applications of methods and simulations.

Declaration of co-authorship: Section 5.1 in the paper gives an overview of the presented simulations, sorting them into three groups: closed-form expressions, differential equations solved in real-time, and RAO imported from external software. The first group of simulations had been previously developed by me, as mentioned in Paper V below. The new simulations were developed by Sergi i Miquel as part of his Master's thesis. Henrique Gaspar was the thesis's main supervisor, and I was co-supervisor. I prepared the paper for submission by reusing parts of the thesis after editing, and writing new text when needed. I carried most of the work toward meeting reviewers' requests. Daniel Vieira assisted me by executing new computational analyses, preparing figures, and writing text for that purpose. Henrique advised general development, defined the paper's overall structure, guided me with responding to reviewers, and arranged the collaboration with TPN.

7.1.2 Paper V

Ícaro A. Fonseca, Felipe F. de Oliveira and Henrique M. Gaspar (June 2019). 'Virtual Prototyping and Simulation of Multibody Marine Operations Using Web-Based Technologies'. In: *Proceedings of the 38th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Glasgow, UK: American Society of Mechanical Engineers. DOI: 10.1115/omae2019-96051.

Abstract: This paper focuses on virtual prototyping and simulation of marine operations based on web technologies. The ship is represented as a digital object, which can be used to perform different types of analyses and simulations. The presented simulations are: motion of a single hull and of multiple hulls in regular waves calculated with closed-form expressions, induced pendulum motion response to a lifted load, and motion of a barge with initial movements in still water calculated with equations of motion.

The simulations are developed as web applications in JavaScript and HTML, with graphical user interfaces and 3D renders of the operations. Relevant parameters of the simulations such as wave characteristics and design dimensions are linked to interactive dashboards, allowing the user to modify them and visualise the results in real-time. The applications are straightforward enough to be executed locally in the web browser of most modern devices.

The work employs an open source approach, relying most notably on the Vessel.js library. This aims to foster reuse of models and collaboration with external contributors.

Declaration of co-authorship: I developed most of the simulations presented (i.e., single hull motion responses, multiple hulls motion responses, and pendulum motion of lifted loads), which required creating a framework to handle simulation states, and writing mathematical formulas and linking them to the PSV model and ocean visualisation. The PSV model and ocean visualisation were developed by other authors, for example, Gaspar (2018b), and already available when the study began, as Section 4.8 explains. I wrote most of the text. Felipe Ferrari developed the barge motion simulation and wrote the corresponding text sections. Henrique Gaspar supervised general development and suggested improvements, such as inclusion of figures and diagrams.

7.2 Objectives

This study develops a library of web-based simulation models that can be executed based on selected data standards. Simulation algorithms followed some drivers, so they contribute to digital twin development. They should handle the various types of states discussed in Section 4.9.2, from discrete to continuous. Following a path toward standardisation, they should follow a common template that can be replicated when developing simulations. Execution should be lightweight to provide prompt (i.e., near real-time) assistance during ship operations.

7.3 Modelling

7.3.1 Outline

The case studies focus on modelling ship motions during operations, including interactions with external systems, such as lifted loads and mooring lines. This topic was chosen for two reasons. One is the number of open source applications that reference development of motion models, tracing to Chaves and Gaspar (2016). The other is TPN's mutual interest in the topic as a research partner, thus influencing the choice of simulation domain. When developing the library, the approach was to provide various simulation models for motion response so that users could choose and combine the ones that most suited their needs. Figure 7.1 shows the process of using the simulation library, which aligns with this chapter's structure. During the first stage, a user defines a floating system model, ship, or platform using the library. The floating system can then be linked to a chain of analyses and simulations — vessel loading conditions, buoyancy conditions, and motion response. Finally, mathematical results are linked to a visualisation so that the user can inspect the motion of an ocean with regular waves (Section 4.8). A single simulation might include various systems that rely on multiple motion models.

7.3.2 Ship representations

Platform supply vessel

The simulations used models across floating systems. The first was a platform supply vessel based on a commercial ship specification (Figure 7.2), developed as one of the first ship models in Vessel.js with the goal of testing and illustrating the library's potential (Gaspar 2018b). It has been used in several behavioural models, including motion

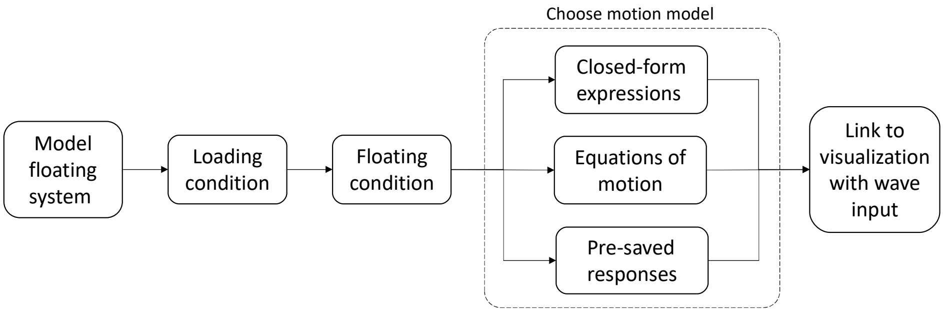


Figure 7.1: The process of setting up a simulation using the library.

responses of single and multiple hulls with closed-form expressions and a lifted load pendulum.

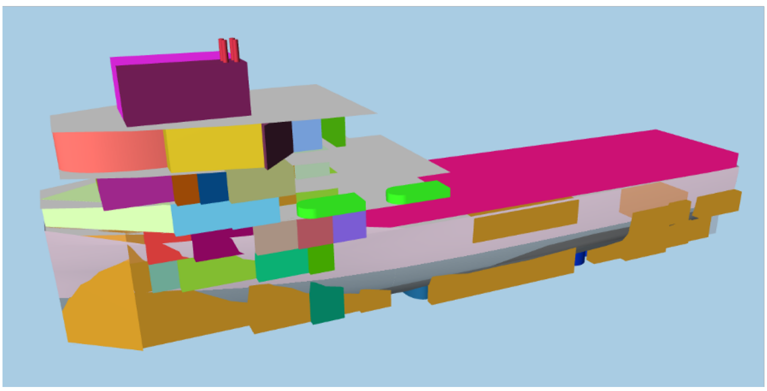


Figure 7.2: Platform supply vessel visualisation, introduced by Gaspar (2018b).

Mississippi barge

The second ship model was based on a typical Mississippi barge (Figure 7.3), created due to a need for a box-shaped hull that would simplify calculation of motion responses based on differential equations of motion (Oliveira 2019). The same barge was also used to explore simulations of coupled motions that involved mooring lines. Each of the barge's four corners was linked to a catenary line, and total motion was evaluated by considering the system as a whole.

Other floating systems

Two other models were developed for simulation of subsea and side-by-side offloading operations. The first was a Floating Liquefied Natural Gas (FLNG) platform and the

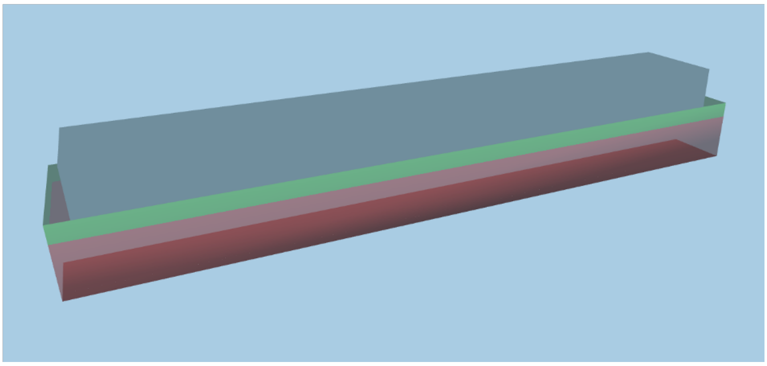


Figure 7.3: Mississippi barge visualisation, introduced by Oliveira (2019).

second a Suezmax tanker (Figure 7.4). The simulations in which they were used evaluated motions based on the corresponding RAOs, calculated using external software. Hull geometries were not used to yield mathematical results when displaying motion to the user, since results were calculated and stored previously. For that reason, the FLNG platform and Suezmax tankers were developed as placeholder visualisations, with box-like hull forms. The hull in Figure 7.4 is being used only for visualisation purposes and not as input for calculations; the motion results for that vessel are imported from Wamit analyses. For this reason, it sufficed to use a simplified geometry as observed in the figure.

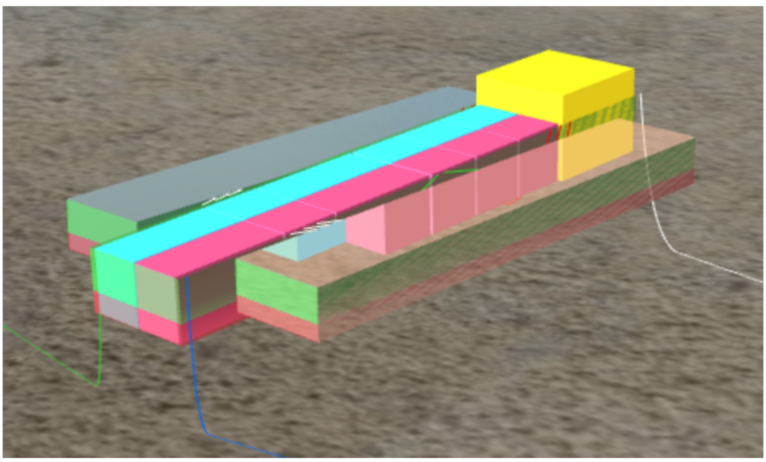


Figure 7.4: Placeholder visualisations for FLNG platform and Suezmax vessels (Miquel et al. 2020).

7.3.3 Behavioural models

Buoyancy condition

The Vessel.js library includes functions to calculate hydrostatic and stability parameters based on a vessel's weight distribution, which is given by the vessel lightweight plus cargo, stores, and appropriately loaded condition of each tank (Figure 7.5). The library finds the balance between the submerged hull volume and the total displacement by integrating the hull form iteratively across draughts. Once the correct draught is found, the library evaluates water plane dimensions and coefficients, form coefficients, position of metacentres, and trim for small angles. The library is not yet capable

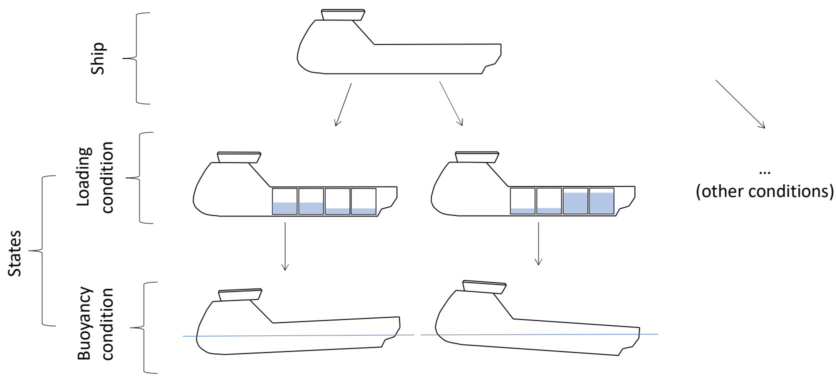


Figure 7.5: Calculation of a buoyancy condition based on tank loading (Fonseca, Oliveira and Gaspar 2019).

Closed-form expressions of motion response

Ship motion responses to regular waves can be calculated using closed-form expressions by Jensen, Mansour and Olsen (2004) (Figure 7.6). The method is intended for use during early design stages, modelling the hull form as a composition of box-shapes when deriving expressions. The formulas account for heave, pitch, and roll motion modes. The wave is defined by angular frequency, amplitude, and heading direction in relation to the ship. The simulation of motions is executed in two stages. In the first one, Vessel.js calculates amplitude responses based on the ship states and wave characteristics. Later, when it is needed to display the motion in a time-continuous visualisation, it interpolated amplitudes into a time-series calculated with a sinusoidal function.

Lifted load pendulum

As a first attempt to include continuous states calculated from differential equations into a simulation, a pendulum model of a lifted load hanging from an A-frame and in-

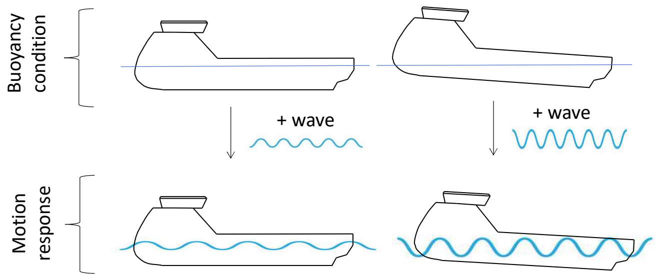


Figure 7.6: Calculation of ship motion based on closed-form expressions (Fonseca, Oliveira and Gaspar 2019).

stalled on the stern was implemented (Figure 7.7). The pendulum was modelled using a Lagrangian formulation for spherical pendulum with a moving pivot (Myhre 2022). The spherical assumption considers that there is no slack to the rope through which the load is suspended, so the load effectively moves on a spherical surface. Resulting formulas were solved using an implementation of the Dormand–Prince method in the open source Numeric.js library for numerical analysis in JavaScript (Loisel 2022). The method solves ordinary differential equations using an adaptive time step. At each new frame of the visualisation, it is invoked to calculate an updated pendulum position.

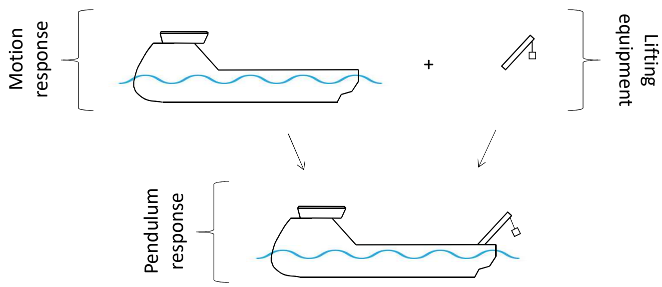


Figure 7.7: Calculation of lifted load by combining ship motion, based on closed-form expressions, and pendulum motion, based on differential equations (Fonseca, Oliveira and Gaspar 2019).

Simplified equations of motion

Motion equations were derived for a simplified example using the box-shaped barge. The simulation assumed that the barge was floating in equilibrium until the user introduced a disturbance in heave, roll, or pitch, triggering a motion response (Figure 7.8). The simulation assumed small displacements in relation to the barge's neutral position to simplify calculations of hydrodynamic coefficients during solutions to the equations (Oliveira 2019). A second simulation was developed in which the moving barge was connected to four mooring lines that interacted with its motion response. Line geometries and forces were solved using a quasi-static, iterative approach for each

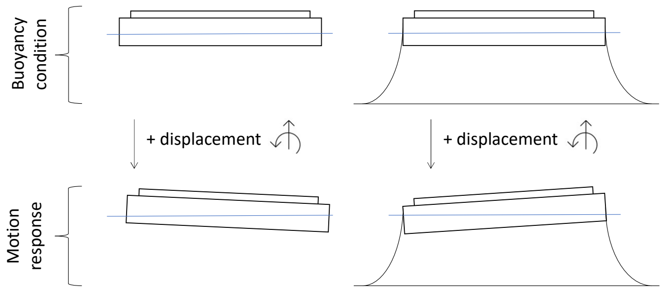


Figure 7.8: Evaluation of barge responses to initial displacements imposed by a user.

Response amplitude operators imported from external software

A viable approach to including accurate motion responses in a web-based simulation is to perform motion response analyses in advance using specialised software and then importing RAOs into the simulation as a text file, as Section 5.4 explains. Motion equations are not solved in real-time, but responses can be fetched quickly from a results database and then interpolated into a sinusoidal time-series to display a desired visualisation to a user. This allows simulation of complex operations, including hydrodynamic coupling among multiple bodies (Newman 2001) and the shadow effect, which accounts for the influence of the Floating Liquefied Natural Gas (FLNG) over the environmental forces acting upon nearby ships (Vieira et al. 2011). In practice this allows simulation of complex operations such as side-by-side offloading with two or more vessels and interactions with mooring lines and hawsers (i.e., cables linking ship and Floating Liquefied Natural Gas (FLNG) platform).

7.3.4 Integration

Methods can be combined to create simulations that reconcile multiple approaches. Table 7.1 summarises the examples, mapping them to the three methods of handling states — discrete states, continuous states based on extrapolation of static results (i.e., RAOs calculated by closed-form expressions or external software), and continuous states based on ordinary differential equations. When assembling these models in an app, continuous states linked directly with the 3D visualisation and refreshed at the same rate. Taking as example the application simulating PSV motions with closed-form expressions running on a regular consumer laptop, the refresh rate revolves around 60 Hz when simulating one hull and around 24 Hz when simulating 8 hulls. Figure 7.9

shows how this interaction works, taking as an example the lifted load app, since it combines the three methods. The three sets of states were calculated using a discrete approach — the loading condition, buoyancy condition, and RAOs. When visualising the simulation, the ship is placed with the correct coordinate positions, heading, and draught based on the first two sets of states. The simulation uses ship and wave characteristics to calculate RAOs based on closed-form expressions, and then saves them. This is done while the visualisation is initialising. Once it starts executing, continuous

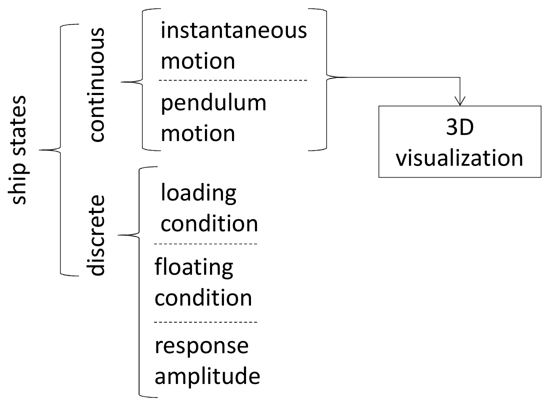


Figure 7.9: Example of data structure storing ship states during simulation, during which continuous states link to a 3D visualisation that is updated in real-time (Fonseca, Oliveira and Gaspar 2019). Although not depicted, linking also applies to wave characteristics.

7.4 Results

Several simulations were developed as open source web apps. Papers IV and V give a broader overview of the app assortment, with Figure 7.10 exemplifying four. The interfaces contain the 3D visualisations, 2D dashboards, and GUIs. Both the 3D visualisation and 2D dashboards show the vessel's motions in real-time. The visualisation renders motions using six degrees of freedom, synchronous with regular waves, providing qualitative insights into motion responses. The 2D charts display time-series for relevant motion modes. When accessed from a web-browser, they give quantitative indications of responses. The GUIs allow the user to configure various simulation parameters — wave characteristics, number of ships in the simulation, and mooring line arrangement. The applications display results responsively and fulfil requirements for continuous execution in real-time, updated at several frames per second.

Table 7.1: Summary of simulations with corresponding modelling methods.

Method \ Example	Single or multiple ships floating in regular waves	Ship with lifted pendulum load	Floating barge with mooring lines	Side-by-side operations
Discrete	Floating states calculated from equilibrium between hull form and displacement.	Same as column 2.	Same as column 2.	
Continuous: time series interpolated from static results	Ship motion interpolated from RAO calculated with closed-form expressions.	Same as column 2.		Ship motions interpolated from RAO and phase difference calculated with external software packages.
Continuous: differential equations		Load modelled as spherical pendulum.	Barge and mooring line motions calculated in real-time.	

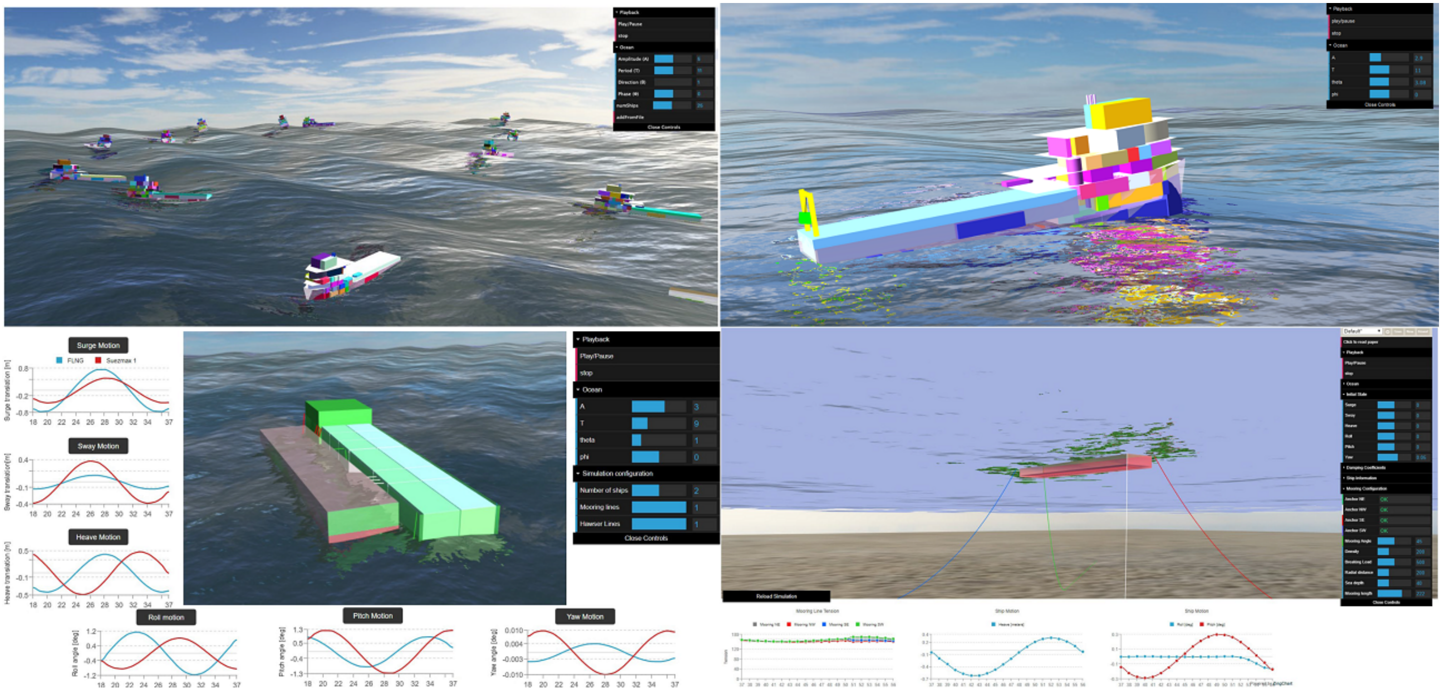


Figure 7.10: Examples of simulations developed using the library (Fonseca, Oliveira and Gaspar 2019; Miquel et al. 2020). Clockwise figures follow the order that the chapter discusses the simulations: multiple hulls based on closed-form expressions, lifted load pendulum, simplified equations of motion, and response amplitude operators imported from external software.

7.5 Discussion

7.5.1 Simulation architecture

It is worth considering how the trade-offs of Vessel.js' proposed architecture in supporting future development of digital twins in terms of data exchange among simulations and reuse of models across case studies. The open source approach to simulation development is valuable for providing clarity regarding methods and assumptions embedded into a simulation. Given commercial concerns about intellectual property, it is unrealistic to expect companies to license all developed software under permissive open source licenses. In any case, some level of access to source code is an effective manner to increase auditability and transparency of software developed by others. Another strength of the proposed approach is the development of simulations based on self-contained models with the aim of supporting reusability. While this feature is promising and relatively novel in the sector, it can benefit from two additional measures to increase its effectiveness. The first one is to further decouple these models to the ship representation and state data structures defined in Vessel.js. This should make it easier for others to reuse and compose simulation models contained in Vessel.js when developing new software applications by avoiding the necessity to specify an entire digital ship representation just to execute the desired model. It is necessary however to plan how these simulations, once decoupled, will be calibrated with the ship parameters they need to be executed accurately (for example, shape coefficients and CG for the buoyancy state being simulated) without requiring definition of an entire ship representation model. The second measure is to take advantage of existing JavaScript development resources to increase scalability of simulation reuse. This could be achieved by adopting the native JavaScript module pattern as implemented in recent versions of the language. These modules can then be deployed through popular package management tools (Wittern, Suter and Rajagopalan 2016), greatly simplifying deployment of simulation deployment over cloud infrastructures.

7.5.2 Long-term support of a maritime simulation library

In addition to improving simulation reuse, a higher level of decoupling would also simplify long-term maintenance of simulations in the long term. For instance, if Vessel.js is intended to work as a cohesive library, novel simulation models introduced to the library should operate in harmony with existing ones. However, as Vessel.js' scope and number of developed simulations increase, so does the number of interfaces to be reconciled when introducing new features. This means that the code base demands more support and maintenance work as it increases in size. This problem is natural with software projects in general, however it is intensified by the fact that the library's main goal is to support research development, which by its own nature should continuously evolve to address new topics and problems. This implies that requirements faced by Vessel.js are much more malleable compared to commercial CAE solutions targeting a well-defined scope and set of use-cases. Thus, when developing calculation of motion responses based on equations of motion, for example, a barge-like hull form was assumed to simplify solution of the hydrodynamic problem. If this simulation model is further developed to account for conventional hull forms, it will be necessary to verify that vessel models previously contained in Vessel.js are compatible with the new equations of motion and calibrated based on realistic coefficients. By changing the development toward a more modular and decoupled approach, the simulation models can be maintained individually, relinquishing expectations that they remain compatible with different ships available in the library. In that context, the parts of the library which are used repeatedly across examples (for example, the interactive ocean visualisation) should also be given priority in support for recurring use. This lesson can be transferred to development of digital twins more generally, as they will face similar maintenance and support challenges in the long-term.

7.5.3 Increasing simulation trustworthiness

Some final suggestions would add trustworthiness to the simulations developed in this study. Since physical units used during simulations were not always obvious to those unfamiliar with the Vessel.js library, there should be a structured way to register them for each input and output parameter. Since the source code is open, this could be done using comments saved to script files. In addition, the simulations are in various stages of verification and validation, but it is difficult for users to identify that clearly. Thus, simulations should be documented using labels that indicate their degree of validity, and point to research or tests that justify their status. For instance, in the case of motion simulations based on closed-form expressions, this might include documenting the deviations caused by the method's simplifications to guide its adequate usage (Ildstad, Kolstad and Halse 2017) and carrying convergence studies to identify the effects of not using a faired hull (Figure 7.2) for estimation of form coefficients.

Chapter 8

Discussion

8.1 Addressing the research questions

Table 8.1 summarises the answers to each research question (Section 2.4), mentioning the texts that give grounds to such answers. The following paragraphs elaborate the answers in the same order.

Differently from the domain of ship design, which is already supported by a mature discipline (Nowacki 2010), the digital twin (and its associated standards) is a relatively novel concept, particularly applied to the ship industry. This novelty reduced the availability of existing digital twin standardisation approaches and limited the feasibility of gathering information about them with the aim of drawing comparisons to introduce improvements. Instead, the two first research questions took and exploratory approach of identifying existing challenges to digital twin standardisation and identify guiding principles to overcome these challenges.

Paper I delineates the challenges to digital twin standardisation based on a literature review covering experiences with previous standardisation initiatives in the ship industry, comparisons with the successful standardisation attempts in the civil engineering industry, design of recent standards, and novel digitalisation trends pointed by other authors. It finds that success of standardisation initiatives is determined not only by the design choices of the standards themselves, but by a complex interaction among the incentives of involved parties, and their perceptions about the advantages and drawbacks of adopting a proposed standard.

Sections 3.10 and 3.11 synthesise the review's finding into guiding principles or drivers for overcoming the identified challenges, namely: pragmatism in scope to reduce implementation stakes, support to heterogeneous systems to achieve baseline interoperability among interacting digital tools, openness to avoid lock-in effects, and intelligibility to give engineers an understanding of the data and assumptions contained in the digital twin. These drivers can be observed in the selection of existing standards and in the proposal of novel methods for digital twin ship development, as Chapter 4 presents.

The literature review also identified a few standards apparently suitable to devel-

Table 8.1: Connection between research questions and publications. Darker shades signify greater contributions to addressing the research question.

Theme	Research question	Answered?
1. Challenges and paths to standardisation of digital twin data	1a. What are the challenges to standardisation of digital twin ship data?	Yes. Paper I finds that standardisation success is determined by the design choices of the standards and by a complex interaction among the incentives of involved parties, and their perceptions about a standard.
	1b. What drivers need to be considered for successful data standardisation in digital twin ships?	Yes. Sections 3.10 and 3.11 synthesise drivers for overcoming the challenges: pragmatism in scope, support to heterogeneous systems, openness, and intelligibility.
2. Application of existing standards	2.a To what extent do existing standards cover digital twin ship data?	Yes. Paper II identified ISO 19848, VIS, glTF and web technologies, then successfully applied them to development of a digital twin of an experiment with a scale model ship in a wave basin.
	2b. Which of the data type contained in the digital twin still lack standardisation?	Yes. The same case study delimited gaps in ship representation besides visualisation and simulations of operational behavior.
3. Further digital twin standardisation	3a. What methods can be proposed to model remaining digital twin content, based on selected standardisation drivers?	Partly. Papers (III-V) turned to the Vessel.js library and complemented it with methods for categorisation of ship representation and web-based simulation of ship motion response. These methods might inform future standardisation initiatives.
	3b. What implications do such methods bear?	Mostly. Section 8.5 summarises the implications derived directly from case studies and anticipates those which are expected in broader industrial cases, standardisation of digital twin for transport vessels and for specialised vessels.

opment of digital twin ships, such as ISO 19848, VIS, glTF. However, they had only been examined on a theoretical basis, as the candidate was not able to find previous literature on the applications of these standards to development of digital twin ships. Furthermore, Paper II surveyed that standardisation of operational context, in the form of meteorological and geographic coastal data, is more disseminated than that of ship data, thus making it less of a challenge to development of digital twins and giving the basis to leave it out of this work's scope.

The case study in Paper II joined the identified standards with a web-driven approach and demonstrated their functionality through development of a digital twin of an experiment with a scale model ship in a wave basin. The experiment applied ISO 19848 to store instrumentation and control logs in human readable files, VIS to identification of the stored logs, glTF for real-time visualisation of the scale model ship being controlled, and the web technologies to reconcile the open standards and the proprietary control and instrumentation systems into a web application, thus leveraging compatibility with a broad range of computers and mobile devices. The case study confirmed the coverage of existing standards and delimit gaps in which they need to be complemented by novel solutions: system metadata (with the aim of enabling an effective ship representation, and not simply visualisation) and simulations of operational behavior. Papers III, IV, and V investigate preliminary methods to model digital twin data, based on the stated drivers, in these two specific domains.

The first domain was ship representation. The ship visualisation in the first case study shared many similarities with the Vessel.js library developed by the candidate's supervisor: combination of 3D models with JSON metadata, and encapsulation of components in digital objects which can be replicated in different places inside the vessel model. Despite this convergence, previous Vessel.js case studies aimed mostly at concept ship design with simplified three dimensional models aimed at early stage ship design.

Paper III expanded Vessel.js to handle digital vessel models suitable for early design and for operation. It achieved that by putting forth a framework for categorisation of ship representation and sensor logs according to different taxonomies. The framework allows linking the conceptual and operational ship models in Vessel.js to hierarchical structures which adequately grasp their different levels of detail, or granularity. The case study demonstrated the framework's functionality by applying it to the conceptual and operational models of a research vessel. The case study found scaling obstacles to automation of the framework, stemming from poor support and reusability of original CAD files, as discussed in Paper III, Section 5.2. For that reason, it was not possible to include the complete detailed model in the vessel dashboard developed for the case study. Paper III broadens the research scope from a pure digital twin defined as a connected simulation supporting system operation, toward a digital thread, meaning the downstream and reuse of data through different lifecycle stages as explained in Section 1.3.

The second digital twin domain with standardisation gaps is simulation of vessel behaviour. The term is quite general, leaving room for a broad range of potential applications (for example, those in Figure 3.1). For this reason, this thesis gave only a partial answer to the related research question, in the form of a web-based library for simulation of vessel motion response. Papers IV and V presented development of motion response simulation with the Vessel.js library. The case studies demonstrated the possibility of simulating seakeeping by composing different simulation methods, applying these methods to different vessel models contained in the library, and visualising simulated operations with a web-based approach. Instead of providing simulations as part of a tight-knit software tool, the library approach provides a collection of models that can be calibrated, combined, and reused to build different simulations.

The remaining sections in this chapter (8.2-8.5) explore the different research questions (i.e. Questions 1a, b, 2a, b, 3a, b, as Table 8.1 shows). Section 8.2 explores the first pair of research questions (1a, b) and Section 8.3 discusses the second one (2a, b). Each of these sections includes one subsection for each research question, one subsection pinpointing a related contribution, and one subsection explaining the contribution's limitation with opportunities for overcoming them. The third pair of research questions deserves an expanded treatment. Section 8.4 answers the research question 3a, discussing findings in the application of standards to ship representation and taxonomies, ship motion response behaviour, their respective contributions and limitations. Finally, Section 8.5 answers research question 3b, explaining the implications of the work to the state of art in light of existing literature.

8.2 Characterisation of digital twin data and related standardisation challenges

8.2.1 Challenges and drivers to standardisation of digital twin ship data

Even before delving into the question of digital twin data standardisation, there is the problem of how to converting the digital twin, with its ambitious scope, into a tractable data modelling problem. This was accomplished with a theoretical framework based on two characteristics: first, that the digital twin might comprise different domain-specific models that should attain some independence among each other; second, that each of these models might attain different levels of maturity (i.e., cost and sophistication) depending on stakeholder's interest and technological feasibility. With that established, the research looked into past standardisation attempts in the ship industry, namely, SFI and STEP, to find the factors which influence standardisation success. Due to the small number of well-documented attempts in the literature, it was not possible to advance a comprehensive theory that would explain why standards become successful. However, it has been possible to identify a few key challenges to standard adoption based on works reporting previous and recent experiences with digital standards. One of these challenges is the historical role of proprietary formats as a competitive advantage for software vendors to acquire and retain customers. This role can stem from benign or adversarial reasons such as, respectively, the desire by software vendors to provide customised features or to complicate migration of customers to competing solutions. In practice, these reasons tend to hinder reaching of consensus when designing and migrating to new standards. Another challenge to the adoption of standards is the complexity imposed on third parties wishing to implement it. Two factors tend to exacerbate this issue: specifications which are extensive and requiring high implementation workloads, and the relatively niche scope of the ship industry in the overall CAE software market.

Quantitative evidence from the literature about IFC standards in civil engineering shows that regulatory frameworks might have a decisive impact in compelling standard adoption (Edirisinghe and London 2015). In the absence of an analogous perspective for the maritime sector, the work toward building consensus among stakeholders takes precedence. With the recent push toward digitalisation, there is now an increasingly receptive view toward interoperability, standardisation, and openness in software systems. A few measures which are judged to contribute to successful standardisation are prescribed in view of the identified challenges. To reduce workload imposed upon potential users, standardisation entities should provide not only specifications for independent implementation, but also resources to enable readily adoption by interested parties. These resources might include, for example, converters and APIs easing interoperability of a new standard with existing systems, or public examples showcasing standard use on the internet. The other prescribed measure constitutes the thesis pursued in this research work. It posits that future standardisation of digital twins should be designed with flexibility to accommodate for the needs of different stakeholders, while retaining a baseline level of compatibility to reconcile these different needs. This should be seen as a calculated compromise toward circumventing the problem of consensus-building and increasing the likelihood of standard adoption.

8.2.2 Contribution 1

The literature review contributed to a better understanding of the challenges to developing successful standards for data modelling in future digital twin ships. This was accomplished in two stages. The first stage took a retrospective look into the issue of digital interoperability in the ship industry by examining the reasons for existing data fragmentation and the outcomes with previous standardisation initiatives. The second stage updated this perspective by considering positions from other authors, comparing the standardisation of ship data to the successful experience in the civil engineering industry, and by comparing design of recent standards for ship data to that of earlier ones. The results of this literature review were synthesised in four principles to successful standardisation, which were used as drivers to the modelling work in the remaining of the thesis: pragmatism in scope, support to heterogeneous systems, openness, and intelligibility.

8.2.3 Limitations of contribution 1

The identified drivers delineate a path to increase chances of success for standardisation of future digital twin ships. However, they are not sufficient to completely specify all design choices leading to a successful standard, or to ensure that a certain standardisation will gain adoption. Those outcomes stem from complex sociotechnical factors which are not completely predictable or controllable by any single party, whether private organisation or public institute.

8.3 Application of existing data standards to digital twin ships

8.3.1 Extent of existing standards' domain coverage

The survey looked for existing standards that can be applied to development of digital twin ships and to modelling of digital twin ship data more specifically, prioritising open and non proprietary alternatives. The recent ISO 19848 standard is an example of such alternatives. It covers machinery and equipment data per se, so it was applied to modelling of ship states based on digital twin sensor logs. ISO 19848 advances the VIS and JSMEA-MAC as naming rules for shipboard data channels, though the case studies indicate that at least VIS is also suitable for organisation of ship representation data in the digital twin. Its use within that scope could be better supported if it were provided in a hierarchical data structure suitable for automatic processing in digital twin applications, as opposed to the current Excel tables in which it is published. The survey found that standardisation of operational environment data, for example, metocean conditions, tends to be more mature and consolidated than ship data. This led to the decision of not focusing case studies on that topic, as there was a greater potential for contribution on less standardised domains. The use of existing environmental data in digital twins remains an interesting subject for further investigation.

The remaining data types contained in the digital twin still lack standardisation to different extents. While there are are various generic 3D formats that can be used during real-time visualisation, they do not cover the metadata necessary to establish a ship representation in the digital twin. The Vessel.js library had previously proposed the combination of STL for visualisation of ship compartments during early design stage, plus a JSON specification containing their spatial arrangement, weight, and material data. This work extended the library with glTF for representation of ship as built, comprising detailed assemblies of parts on board. Development of ship behavioural models for use in digital twins is another topic where standardisation is still lacking. A web-based approach has functioned thorough this work as a common denominator to ensure accessibility and compatibility. I.e., by adopting web-driven implementation languages (i.e., HTML, JavaScript, CSS) and information modelling languages (i.e., JSON), it has been possible to take advantage of existing web infrastructure toward allowing access to the case studies through web-browsers. However, this does not mean the standards proposed here have their relevance circumscribed to web-based applications, since they have been selected to enable data exchange and interpretation even if the software in which they are used is not web-based. Thus, a ship geometry created with a CAD tool might be exported to STL or glTF and visualised in a graphics engine not based on WebGL, or the JSON sensor logs might be sent to a third-party for evaluation of machinery condition, read as a text file and interpreted based on the accompanying metadata.

8.3.2 Contribution 2

The case study made concerted use of ISO 19848, VIS, glTF, and web-based technologies to develop a digital twin of an experiment of a scale model ship and thus demonstrate these standards' functionality. The publication of ISO 19848 with VIS as one of its canonical naming rules is relatively recent (i.e., 2018), thus making it novel to apply these to a case study and to give an account of the scope, design choices, and compromises behind their normative specifications. Same for glTF and web-based technologies, both of which have not been extensively applied to development of digital twin ships.

8.3.3 Limitations of contribution 2

The contribution has two limitations. First is that, despite the case study aiming to adopt open standards, the control module used to manoeuvre the scale model was developed in Matlab, a proprietary software suite (Ianagui 2019; MathWorks 2013). An alternative to overcome this limitation could be to redevelop the control model with the Functional Mock-up Interface (FMI), an open standard for exchanges of simulation modules and development of distributed simulations. There are already initiatives to apply FMI to model control-oriented aspects of vessel navigation systems and crane operation in digital twin ships (Hatledal et al. 2020). The second limitation is that the equipment used for data exchanges in the wave basin experiment is not representative of infrastructures to transmit data between real ships and shore. The communication between scale model and control module occurred via radio transmitters; communication between the computer which oversaw experiments and all other wave basin equipment occurred through User Datagram Protocol (UDP) and WebSocket protocols. While it has not been this work's scope to cover network infrastructure aspects of digital twin, they will inevitably need to be considered in full scale applications. A path for future research on that topic is the recent ISO 19847 standard,[1] which specifies requirements for data servers collecting tabular and event logs from shipboard machinery and systems and to further share these logs with systems on shore (ISO 2018a).

8.4 Methods to model remaining digital twin content

8.4.1 Ship representation and taxonomies

The framework was implemented with open standards such as JSON, for vessel hierarchical structures, glTF and STL, for visualisation of 3D models, and web-technologies, for user interface. The case study unveiled prerequisites to scalable application of the framework, discussed in Paper III, Section 5.2. The first is that the detailed vessel 3D model is layered in a manner which is consistent with an engineering understanding of the asset, instead of having surfaces aggregated in groups which are solely convenient for modelling tasks in CAD suites. Admittedly, there is always a component

[1] Namely: Ships and marine technology — Shipboard data servers to share field data at sea.

of subjectivity (and even arbitrariness) in any product taxonomy, for the reason that the taxonomy is not an intrinsic feature of the product being described, but is rather a classification scheme imbued by the intentionality of the designer which devises or applies it. Pragmatically speaking, however, the three detailed taxonomies considered during this work (i.e., VIS, JSMEA-MAC, or SFI) are concerned with describing vessels in terms of roughly a common set of basic machinery components, for example, engines, boilers, pumps. Furthermore, VIS and SFI cover vessel aspects such as structural elements, outfitting, accommodation, and also converge to similar basic concepts in these domains. The divergence among the different taxonomies (at least during operational phase) tend to happen in the naming terminology and in the choice of hierarchical structures which group together the often equivalent basic concepts. Back to the main point, by arranging the detailed vessel's 3D models according to its engineering components, CAD users ensure the model will be reused more effectively during subsequent lifecycle stages, and thus will integrate more easily into a digital thread.

The second condition to scalable application of the framework is that the chosen formats provide good support to heterogeneous systems. As discussed in Section 4.6.3, one of the reasons for choosing to use glTF in the case study was that its standardisation consortium provided open source loaders in converters to other libraries and formats, thus greatly easing implementation of the standard by independent developers. This approach can be compared to Jupiter Tessellation (JT), a visualisation standard which was considered in depth during development of the case study. JT was previously owned by Siemens and later published through ISO. In theory, JT can be implemented by anyone who accesses the file format reference (ISO 2017; Siemens 2019), however the specification is quite extensive, making development prohibitive for smaller teams. Furthermore, when looking for a loader to use JT in web applications, the candidate was only able to find one inside Siemen's own proprietary suite. In one of the software applications, the user has the option of exporting a JT model to a file that contains all of the source code to visualise it as a web page. By inspecting the generated file, it is possible to identify that it uses excerpts from Three.js source code to render the JT model. This example demonstrates how the support provided by a given standards to existing, heterogeneous systems might drastically influence the convenience and simplicity of applying that standard in practice.

8.4.2 Contribution 3

Contribution 3 is a standards-based framework for linking ship data to various hierarchical schemas. A case study showed how the framework can be used to handle ship representation both during concept design and operation. Furthermore, it was explained how the framework might be combined with ISO 19848 for identification of sensor logs. The novelties of this contribution compared to extant research (Pal 2015) are its complete reliance on open standards, open source code, and its web-driven characteristic, allowing easy content access and sharing through the internet. The combination of various standards with specialised scopes enables serialisation (i.e., storage) of individual digital twin contents such as taxonomies, 3D models, and sensor data across software. This opens the possibility of reusing some of the framework's ideas and implementations without necessarily adopting all of the standards used in the case study.

8.4.3 Limitations of contribution 3

The contribution has two limitations. The first one is that the glTF and STL formats used provide only tessellated representation of ship geometry. Thus, they are unable to support tasks requiring smooth (i.e., faired) representation of surfaces, which are usually accomplished through spline curves. This makes it necessary to investigate how these formats can be seamlessly linked to smoothed CAD models in the digital thread. Recently, the Open Class 3D Exchange (OCX) format has been proposed as a format for exchange of 3D models related to class approval in the ship industry (OCX Consortium 2022). It claims to support conversion from CAD formats from different software vendors and to also enable web-based visualisation. The OCX working draft was made public after the first version of this thesis had been delivered to the evaluation committee, so it was not possible to consider it in this research. The second limitation the framework not being demonstrated to accommodate structural detailing or distributed ship service systems such as piping and wiring installations (Mukti, Pawling and Andrews 2021). In the case of machinery components included in the Gunnerus digital twin, division of 3D models is self-evident. However, attempting to divide piping spools or structural reinforcers into independent components that can be incorporated in the framework would be cumbersome, since any variation in shape (i.e., length, thickness, or diameter) would qualify as a new base object in the model. One approach is handling such parts as a database of two-dimensional profiles that can be extruded along a path inside of the ship model (Chaves, Gaspar and Borgen 2018).

8.4.4 Ship motion response behaviour

Chapter 7 expanded on existing simulation capabilities of the Vessel.js library, a significant part of which developed during the candidate's MSc thesis (Fonseca 2018), with different types of motion response simulations based on closed-form expressions, simplified equations of motion solved in real-time, and response amplitude operators imported from external software. The mathematical models were linked to web-applications displaying interactive visualisations of the simulated motion. The developed applications give a spatial sense of the operations, being a useful tool to support simulation of maritime operations. This perception would later be reinforced during preparation and execution of the wave basin experiments discussed earlier. As a direct consequence of the adopted web-based approach, the simulations offer good compatibility with different devices, being easily distributed to users, and accessed by them.

As development of the applications progressed and the simulated operations increased in complexity, it was found that development workload was increasingly related to the challenges of modelling complex maritime operations hydrodynamically, instead of being focusing on the standardisation which the thesis' main research aim. These challenges are brought by the necessity to, for example, obtain accurate motion coefficients for each vessel and to capture motion coupling among several hulls on a side-by-side offloading configuration. The implications of that fact can be observed in the evolution of the case studies: as the simulated operations increased in complexity, they forced adoption of restrictive simplifications upon the general motion response problem to make development manageable under the scope of this research. An example of this is the simplification of the hull form to a barge in a simulation accounting only for response to an initial imposed motion on still water. During development of the following case studies, it was already clear that the main contributions of this line of work were concentrated on the visualisation capabilities for complex operations and not on re-implementing hydrodynamic models on a JavaScript engine. This led to the choice of reusing RAO results calculated with external software to simplify modelling of complex operations.

In addition to informing the adjustments in direction of following research (as explained in Section 2.5.2) this issue also brings lessons against over-specialisation of digital twin standards. It seems more promising to develop standardisation by attempting to cover simpler use cases which are applicable to most transport and service vessels, rather than focusing efforts into covering advanced operations which address the needs of a smaller user base (in this case, vessels which perform side-by-side operations involving interaction with mooring systems). This perspective is further discussed in Sections 8.5.2 and 8.5.3.

8.4.5 Contribution 4

This thesis' final contribution is a web-based library for simulations of motion response. The novelty of this contribution is in proposing a web-based approach to simulation and visualisation of maritime operations. It provides a collection of lightweight methods which can be executed in a web-browser on an average consumer laptop in real-time.

8.4.6 Limitations of contribution 4

Contribution 4 has two limitations, which have already been pointed in previously. The first one is the absence of stochastic models of wave state and motion response in the Vessel.js library, as it relies entirely on regular wave states. This limits its capability of simulating complex seaways with adequate accuracy. The second limitation is the library not offering a method of connecting simulations to sensor log streams, which is required for digital twin development. This needs to be addressed through further research on simulation-aided support of ship operation, including suitability for real-time execution and connectivity to data streams.

8.5 Implications of digital twin standardisation

8.5.1 Modelling and data handling

Instead of aiming for a comprehensive digital twin model with a single standard, the modelling approach underlying the thesis and case studies was designed to support heterogeneous systems with a collection of standards. In practice, this approach can accommodate different use cases by giving room for flexibility in implementation, while maintaining a level of compatibility. This customisation is achieved with different techniques depending on the data type. For a ship's representation and sensor logs, it was done by separating the data content from the hierarchy that contextualises them. For ship behavior, it is realised with a library of reusable simulation models that can be combined to compose the desired simulation. While this type of flexibility is judged adequate to development of digital twins, it also brings trade-offs, which need to be considered before deployment at an industrial scale. The first trade-off is related to implementation workload. A monolithic system is normally developed by a single organisation, which exerts complete control over formats and other conventions adopted by the software tool or suite. This approach brings some of the risks discussed earlier, however it also ensures that these systems are completely functional out-of-the box, i.e. turnkey functionality.

On the other hand, an approach based on heterogeneous systems might incur in some implementation workload to the organisations trying to integrate a certain set of tools. For instance, it was mentioned that ISO 19848 or Chapter 6's framework allow users to choose different naming rules or taxonomies in the digital twin. If a stakeholder adopts one of these standards and wants to ensure the data channels are named consistently across a fleet of vessels, it is necessary to confirm they adopt the same taxonomies and that the names for analogous data channels or 3D models are constructed identically. The same can be said of integration with sensor streams. If a shipboard system or computer server provides an API for transmission of sensor readings, the stakeholder on the receiving end will need to develop a client or listener which is able to connect with that API before using these logs. This implementation work can be alleviated by choosing and developing standards with good open source support, as discussed in 8.4.1.

The second is related to data correctness. In this regard, it is necessary to ensure the flexibility of the proposed standards does not translate to laxity when implementing them. Adopted standards allowed creation of a digital twin in which each part is reusable and intelligible outside of the original system. This has the potential to simplify data exchanges, since 3D models, sensor logs, and simulation models can be exchanged and interpreted among applications and users. However, if this openness to data exchange might make data content more vulnerable to unintended modifications compared to tight-knit proprietary alternatives. Two examples are given. Since a considerable amount of data and metadata are stored in JSON, which is simply a text format, it is exposed to direct reading and modification. Thus, users should have have good familiarity with the standards if they are to edit data directly, otherwise risking loss of data quality if this is done indiscriminately.

The other example is ensuring consistency of modelling granularity during the design process. Mukti (2022) discusses how an effective ship design approach should be able to handle increasing design granularity from various aspects: weight, volume, component, connections, and so on. Chapter 6 presented a framework which is able to handle both concept and detailed designs with adequate detailing; however, as Section 6.7 pointed, it did not provide recommendations about how to manage the transition from one to the other. Poor execution of this detailing could lead to double counting weight of the same equipment at different levels of the vessel hierarchy. To improve upon this last example, the methods proposed in this work should be developed to accommodate other PDM dimensions (Section 3.3), such as versioning and status, while supporting simultaneous interactions with multiple users.

8.5.2 Transport vessels

In an argument that could be extended to other data-driven methods as well, Section 1.4 argued that the digital twin has the potential to increase transparency of vessel operation, and of actions taken upon that operation to increase its effectiveness. With the development of case studies and the broadening of the research to consider not only the digital twin in vessel operation, but also its interaction with earlier life lifecycle stages in Chapters 6 and 7, it is now possible to evaluate the digital twin's adequacy in response to the distinct challenges of early design and operation stages.[2] On the broader implications of the digital twins and its standardisation to the maritime industry, it is convenient to differentiate the discussion about cargo transport vessels in general from that about specialised vessels, included there offshore support vessels, research vessels and warships.

The first group is characterised by a fleet with lower degree of mission specialisation, where the mission is the cargo transport itself. This lower degree of specialisation normally leads to less design diversity because, as Levander (1991) summarises, "the price is important and serial production of similar designs keep (sic) the prices low." In that context, and considering decreasing prices of sensor technology, the merchant fleet has strong potential for further development and deployment of standardised digital twin technologies. Gaspar (2018a) proposes adaptation of DevOps practices from software development to ship design, allowing continuous monitoring, evaluation, and improvement of ships based on data. This idea can be extended to ship operations, with aims of monitoring and recommending action on topics such as fuel efficiency, voyage optimisation, structural integrity, and so on. The operation phase might even lend itself more promptly to application of DevOps principles compared to the design phase because, as the time spans of individual ship operations are shorter than that of design, they should allow more agility in the feedback loops consisting of planning, implementation and evaluation of corrective measures.

Levander's argument can also be taken to imply that the loss of creative freedom in the design of such vessels imposed by standardisation is not a major concern, at least putting aside the more recent, impending necessity for novel fuel sources with lower carbon emissions. That trend can be observed at its limit on the initiatives toward autonomous navigation in short sea shipping, for example, the autonomous container vessel Yara Birkeland (Yara International 2022). If the project is innovative as a whole due to its aim toward autonomous operation, the vessel's mission is unremarkable from a pure naval architecture perspective: container transport on a regional route. In summary, standardised connectivity and digital models for these vessels could open

[2]The choice of these two phases is due to the case studies having focused on them. This does not mean, however, that there is no potential for application of digital twins to ship construction, as Taylor et al. (2020) mention.

the way to higher synchronisation of supply chains (Sako 2022), which is per se a valuable outcome, even if the ship design concerns fall to second place.

8.5.3 Specialised vessels

The situation is more nuanced when it comes to the second group of vessels, including warships and complex offshore support vessels. These vessels are equipped with a collection of specialised systems and follow a distinct operation profile where transport is not the vessel's core mission, but it is rather carried with the intent of executing a certain task in a given location. Drawing from literature about urban and social planning (Rittel 1982), Andrews (2018) describes early stage ship design as a "wicked problem", meaning, most importantly, that the problem (in this case, the ship design problem) is intrinsically linked to its formulation, to the point where one thing is the other. Furthermore, as it is unfeasible to consider the entire solution space to a wicked problem in a formal sense, its potential solutions need to be considered in the form of different material proposals, each underlain by a set of choices (within given requirements and constraints) about what exact problem is being solved and how. Andrews (2018) denominates as "style" this set of choices guiding design concept to undertake one form rather than another.

These issues bring challenges to application of standards-based digital twins to design and operation of such vessels. It does not seem feasible to expect a modelling approach with good correspondence between digital twin standards and the more specialised vessel systems. Attempting so would either imply on more work by standardisation entities to cover increasingly special cases, or the naval architect limiting their design choices in favour of arriving at concepts which enjoy better coverage of available standards. The latter risks overemphasising means (i.e., standardisation as an enabler to interoperabilty) in detriments of ends (the effective vessel utilisation that such interoperabilty aims to enable). This is one of the reasons why, when selecting domains for simulation development, Chapter 7 turned to general naval architecture concerns like hydrostatics, stability, and motion response, instead of focusing on standardisation of specific offshore operations such as anchor handling or pipe-laying. As these operations are more complex and niche, they would not enjoy standardisation benefits to the same degree.

A feasible approach to still reap the benefits of digital twins in such case would be to leverage existing standards to the maximum extent, and then, if judged advantageous, to build custom simulation models on top of these. So, in the example of an anchor handling operation, the digital twin could measure bollard pull and vessel motions based on standardised data tags, if available, and then complement these with custom tags and simulation models for structural integrity in the anchor handling winches with the intent of forecasting its remaining operational life span. This would at least allow application of Gaspar's data loop during operation phase.

The usage of digital twins to aid early stage design concerns is a challenging topic; still, it is possible to outline two tentative applications. The first one is broadening the perspective of the DevOps loop to the lifecycle as a whole to verify the impact of style variants in a given aspect of vessel performance. This verification, however, would only be possible in a retrospective and somewhat localised fashion. For instance, imagine that a pipe-laying vessel's designer carried feasibility studies, possibly simulation-based, for a S-lay method in relation to a J-lay one, and opts for the latter due to concerns of structural integrity and possibility to execute operations in deeper waters. A digital twin could be set up during commissioning to verify that the simulation results from simulations carried during design were accurate and that the vessel is performing as expected. However, this provides only partial understanding about the original style issue because it throws further light upon a choice that has already been made, but does not elucidate trade-offs in relation to the dismissed alternative (in this example, the S-lay method).

The second potential application is using web-based methods to streamline communication and feedback about concept proposals. The different concepts generated during early stage design could be modelled with a few blocks in a web environment and annotated with explanations about their underlying drivers. The sharing of these models with stakeholders on a web environment would ensure they always have access to the latest concept version, thus setting a common basis for discussions about requirement elucidation.

Chapter 9

Conclusion and recommendations

This thesis investigates use of standards with the purpose of ensuring interoperability among upcoming digital twin systems, simplifying data consumption and exchanges. The first contribution of this thesis is in the survey of existing challenges to digital twins standardisation and identification of drivers to overcome them The second contribution is the assessment of existing standards that can be employed towards the work's objective and their application to a case study. To that purpose, standards accounting for several modelling levels are introduced, progressing from web-driven implementation technologies toward the digital twin's domains of discourse. The selection prioritises open and vendor-neutral standards, though compromising when necessary toward paid or proprietary alternatives. The remaining contributions proposed methods to model digital twins aspects of taxonomies for ship representation data and simulations of motion response. These methods are intentionally loosely-coupled to provide support to heterogeneous systems and use case flexibility across the ship industry. They also suggest directions for future standardisation initiatives. Given the digital twin's ambitious scope, this thesis has only outlined a standards-based approach to data modelling, leaving room for further research. More than hastening toward digital twins of increasingly ambitious functional scope, subsequent studies should aim to establish robust foundations upon which digital twins can be developed and supported.

The domain of behavioural models offers the greatest number of unaddressed questions — and thus directions for future research — due to traditional deficiencies of product modelling standards in covering functional aspects of engineering systems and to this work's specific limitations. As mentioned, the simulations presented in Chapter 7 do not qualify as digital twins because of their dissociation from operational vessel data. From a naval architecture perspective, it is necessary to elaborate the techniques through which the digital twin will aid vessel operation. E.g., Erikstad and Bekker (2021) outline patterns for intelligent services based on digital twins. However, this is done only through high-level functional descriptions of such patterns, without much detailing of how they would be implemented. A purposeful approach to digital twin development needs to identify, among the different vessel operation concerns, those to which the patterns would yield highest value. Instead of focusing extensively on a single simulation domain, such as motion response to waves, this identification could be carried through comparative experimentation with diverse operational disciplines. It is worth investigating the possibility of a simulation approach that can be applied during both ship design and operation. E.g., algorithms would be developed to allow prompt execution so they can first enable responsive analysis of vessel proposals during design phase and then be reused for real-time decision-making during operation. To that purpose, simulation algorithms which are evaluated over any type of range that a user defines, whether a time-series, a collection of wave scenarios, or disparate ship conditions, are suggested. This would allow them to both evaluate design scenarios and connect to operational data. From a data perspective, it is necessary to devise techniques to connect simulation models in the digital twin to the data streamed from vessels. This task would, of course, take advantage of existing resources for handling of sensor observations, including network protocols, programming interfaces and techniques. An example is the Observer pattern, which enforces synchronisation of states propagated through a sequence of algorithms (Gamma et al. 1993). Deployment and reuse of behavioural models at scale could be streamlined through package managers, avoiding centralised control by proprietary application stores and potentially establishing fairer competition among service providers.

A standardisation approach to models of operating context can be outlined in a few stages, taking the Norwegian coast as an example. First, documents with digital elevation models maintained by public authorities can be linked to a script for visualisation of coastal relief (Sandvik 2022). A user would specify a range of latitudes and longitudes, and then a script would generate the 3D model for the corresponding region automatically. It would then be necessary to survey metocean providers that operate on the Norwegian coast, ideally offering weather monitoring and forecast, and evaluate how their data can be integrated to digital twin applications. Finally, it is necessary to devise mechanisms to fetch meteorological data according to the geographic region of the simulation. Conventional geographical grids based data storage techniques might help in this regard. Another opportunity for future research is to widen the perspective of digital twin data management toward the challenges of governance at industrial scale. This could be done by considering the infrastructure and managerial practices necessary to develop digital twins supporting various users with diverse privileges sets. A related project carries an early investigation into how web-based architectures might be employed to that end (Oliveira 2021).

Bibliography

Ando, Hideyuki (2019). 'Digitalization in the Maritime Industry'. In: *ClassNK Technical Journal* 1.

Andrade, Sthefano Lande, Thiago Gabriel Monteiro and Henrique M. Gaspar (May 2015). 'Product Life-Cycle Management In Ship Design: From Concept To Decommission In A Virtual Environment'. In: *Proceedings of the 29th European Conference on Modelling and Simulation*. Albena (Varna), Bulgaria: ECMS. DOI: 10.7148/2015-0178.

Andrews, D (Dec. 2018). 'The Sophistication of Early Stage Design for Complex Vessels'. In: *International Journal of Maritime Engineering* 160.SE 18. DOI: 10.3940/rina.ijme.2018.se.472.

Andrews, D and C Dicks (June 1997). 'The building block design methodology applied to advanced naval ship design'. In: *Marine Design Conference (IMDC'97)*. Newcastle, UK.

Andrews, David and Rachel Pawling (2008). 'A case study in preliminary ship design'. In: *International Journal of Maritime Engineering* 150.

— (May 2009). 'The Impact of Simulation on Preliminary Ship Design'. In: *Marine Design Conference (IMDC'09)*. Trondheim, Norway.

Astrup, O C and C Cabos (Sept. 2017). 'A Model Based Approval Process for Basic Hull Design'. In: *International Conference on Computer Applications in Shipbuilding (ICCAS)*. Singapore.

Bekker, Anriëtte, Mikko Suominen, Pentti Kujala, Rosca Johan Oscar De Waal and Keith Ian Soal (Apr. 2018). 'From data to insight for a polar supply and research vessel'. In: *Ship Technology Research* 66.1, pp. 57–73. DOI: 10.1080/09377255.2018.1464241.

ben-Aaron, Diana (9th Apr. 1985). 'Weizenbaum examines computers and society'. In: *The Tech*. URL: http://tech.mit.edu/V105/N16/weisen.16n.html (visited on 24/10/2021).

Bertram, Volker and G D Thiart (June 2005). 'Simulation-based ship design'. In: *Europe Oceans 2005*. Brest, France: IEEE, pp. 107–112. DOI: 10.1109/oceanse.2005.1511693.

Boschert, Stefan and Roland Rosen (2016). 'Digital Twin—The Simulation Aspect'. In: *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers*. Ed. by Peter Hehenberger and David Bradley. Cham: Springer International Publishing, pp. 59–74. ISBN: 978-3-319-32156-1. DOI: 10.1007/978-3-319-32156-1_5. URL: https://doi.org/10.1007/978-3-319-32156-1_5.

Bostock, M., V. Ogievetsky and J. Heer (Dec. 2011). 'D^3 Data-Driven Documents'. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2301–2309. DOI: 10.1109/tvcg.2011.185.

Cameron, David B, Arild Waaler and Tiina M Komulainen (2018). 'Oil and Gas digital twins after twenty years. How can they be made sustainable, maintainable and useful?' In: *Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), 26-28 September 2018, Oslo Metropolitan University, Norway*. Linköping University Electronic Press, pp. 9–16.

Chaves, Olivia and Henrique M Gaspar (May 2016). 'A web based real-time 3D simulator for ship design virtual prototype and motion prediction'. In: *Proceedings of the 15th International Conference on Computer and IT Applications in the Maritime Industries, Lecce, Italy*. Lecce, Italy, pp. 410–419.

Chaves, Olivia S., Henrique M. Gaspar and Henning Borgen (Sept. 2018). 'An Open and Collaborative Knowledge-Based Approach for Automated Design of Hull Scantlings'. In: *International Conference on Ships and Offshore Structures (ICSOS)*. Gothenburg, Sweden.

Coraddu, Andrea, Luca Oneto, Francesco Baldi, Francesca Cipollini, Mehmet Atlar and Stefano Savio (Aug. 2019). 'Data-driven ship digital twin for estimating the speed loss caused by the marine fouling'. In: *Ocean Engineering* 186, p. 106063. DOI: 10.1016/j.oceaneng.2019.05.045.

Coyne, R. D., M. A. Rosenman, A. D. Radford, M. Balachandran and J. S. Gero (1990). 'Knowledge-Based Design Systems'. In: ed. by Richard Coyne. Addison-Wesley Publishing Company. Chap. 3 - Representing Designs and Design Knowledge, pp. 87–151.

Datta, Shoumen Palit Austin (Nov. 2017). 'Emergence of Digital Twins - Is this the march of reason?' In: *Journal of Innovation Management* 5.3, pp. 14–33. DOI: 10.24840/2183-0606_005.003_0003.

Defense Acquisition University (2022). *Glossary of Defense Acquisition Acronyms and Terms*. Accessed: 2022-10-01. URL: https://www.dau.edu/glossary/Pages/Glossary.aspx.

DNV (Oct. 2020). *RP-A204 Qualification and assurance of digital twins*. Recommended Practice. Det Norske Veritas, Høvik, Norway.

— (2022). *DNV sensor naming resource page*. Det Norske Veritas. Accessed: 18-04-2022. URL: https://vista.dnv.com/docs/.

Ecma International (Dec. 2017). *ECMA-404 - The JSON Data Interchange Syntax*. Ecma International, Geneva, Switzerland. Accessed: 24-04-2022. URL: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/.

— (June 2021). *ECMA-262 - ECMAScript® 2021 Language Specification*. Ecma International, Geneva, Switzerland. Accessed: 24-04-2022. URL: https://www.ecma-international.org/publications-and-standards/standards/ecma-262/.

Edirisinghe, Ruwini and Kerry London (Oct. 2015). 'Comparative analysis of international and national level BIM standardization efforts and BIM adoption'. In: *Pro-*

ceedings of the 32nd CIB W78 Conference. Eindhoven, The Netherlands, pp. 149–158.

Erikstad, Stein Ove (May 2017). 'Merging Physics, Big Data Analytics and Simulation for the Next-Generation Digital Twins'. In: *11th Symposium on High-Performance Marine Vehicles*. Cardiff, UK.

— (2019a). 'Design for Modularity'. In: *A Holistic Approach to Ship Design: Volume 1: Optimisation of Ship Design and Operation for Life Cycle*. Ed. by Apostolos Papanikolaou. Cham: Springer International Publishing, pp. 329–356. ISBN: 978-3-030-02810-7. DOI: 10.1007/978-3-030-02810-7_10. URL: https://doi.org/10.1007/978-3-030-02810-7_10.

— (May 2019b). 'Designing Ship Digital Services'. In: *18th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'19)*. Pavone, Italy, pp. 354–363.

Erikstad, Stein Ove and Anriëtte Bekker (Aug. 2021). 'Design Patterns for Intelligent Services Based on Digital Twins'. In: *20th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'21)*. Mülheim, Germany, pp. 235–245.

Fonseca, Ícaro A. and Henrique M. Gaspar (May 2015). 'An Object-Oriented Approach For Virtual Prototyping In Conceptual Ship Design'. In: *Proceedings of the 29th European Conference on Modelling and Simulation*. Albena (Varna), Bulgaria: ECMS. DOI: 10.7148/2015-0171.

— (2022). 'An Open Framework For Data Taxonomies In Digital Twin Ships'. In: *Submitted to International Journal of Maritime Engineering; re-submission being prepared after first round of reviewers' comments.*

Fonseca, Ícaro A., Felipe F. de Oliveira and Henrique M. Gaspar (June 2019). 'Virtual Prototyping and Simulation of Multibody Marine Operations Using Web-Based Technologies'. In: *Proceedings of the 38th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Glasgow, UK: American Society of Mechanical Engineers. DOI: 10.1115/omae2019-96051.

Fonseca, Ícaro Aragao (2018). 'An Open and Collaborative Object-Oriented Taxonomy for Simulation of Marine Operations'. MSc dissertation. Norwegian University of Science and Technology.

Fonseca, Ícaro Aragão and Henrique Murilo Gaspar (Sept. 2020). 'Challenges when creating a cohesive digital twin ship: a data modelling perspective'. In: *Ship Technology Research* 68.2, pp. 70–83. DOI: 10.1080/09377255.2020.1815140.

Fonseca, Ícaro Aragão, Henrique Murilo Gaspar, Pedro Cardozo de Mello and Humberto Akira Uehara Sasaki (Jan. 2022). 'A Standards-Based Digital Twin of an Experiment with a Scale Model Ship'. In: *Computer-Aided Design* 145. DOI: 10.1016/j.cad.2021.103191.

Fuller, Aidan, Zhong Fan, Charles Day and Chris Barlow (2020). 'Digital Twin: Enabling Technologies, Challenges and Open Research'. In: *IEEE Access* 8, pp. 108952–108971. DOI: 10.1109/access.2020.2998358.

Gamelab Conference (2018). *#Gamelab2018 - Jon Blow's Design decisions on creating Jai a new language for game programmers*. YouTube. Accessed: 2022-04-16. URL: https://www.youtube.com/watch?v=uZgbKrDEzAs.

Gamma, Erich, Richard Helm, Ralph Johnson and John Vlissides (July 1993). 'Design patterns: Abstraction and reuse of object-oriented design'. In: *Proceedings of the Object-Oriented Programming, 7th European Conference (ECOOP'93)*. Springer. Kaiserslautern, Germany, pp. 406–431.

Garzke, William, Dominick Cimino and Matthew D. Yoder (Oct. 2014). 'Improvements and Guidance to the Weight Classification using Expanded Ship Work Breakdown Structure (ESWBS)'. In: *SNAME Maritime Convention*. Houston, Texas, USA. DOI: 10.5957/smc-2014-p49.

Gaspar, Henrique M (May 2017). 'JavaScript Applied to Maritime Design and Engineering'. In: *16th Conference on Computer and IT Applications in the Maritime Industries*. Cardiff, UK, pp. 253–269.

— (May 2018a). 'Data-Driven Ship Design'. In: *17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18)*. Pavone, Italy, pp. 426–439.

— (June 2018b). 'Vessel.js: an open and collaborative ship design object-oriented library'. In: *Marine Design Conference (IMDC'18)*. Helsinki, Finland.

— (Mar. 2019). 'A Perspective on the Past, Present and Future of Computer-Aided Ship Design'. In: *18th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'19)*. Tullamore, Ireland, pp. 485–499.

Gielingh, Wim (July 2008). 'An assessment of the current state of product data technologies'. In: *Computer-Aided Design* 40.7, pp. 750–759. DOI: 10.1016/j.cad.2008.06.003.

Grieves, Michael and John Vickers (2017). 'Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems'. In: *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Ed. by Franz-Josef Kahlen, Shannon Flumerfelt and Anabela Alves. Cham: Springer International Publishing, pp. 85–113. ISBN: 978-3-319-38756-7. DOI: 10.1007/978-3-319-38756-7_4. URL: https://doi.org/10.1007/978-3-319-38756-7_4.

Hamer, P. van den and K. Lepoeter (1996). 'Managing design data: the five dimensions of CAD frameworks, configuration management, and product data management'. In: *Proceedings of the IEEE* 84.1, pp. 42–56. DOI: 10.1109/5.476025.

Hatledal, Lars Ivar, Robert Skulstad, Guoyuan Li, Arne Styve and Houxiang Zhang (2020). 'Co-simulation as a Fundamental Technology for Twin Ships'. In: *Modeling, Identification and Control: A Norwegian Research Bulletin* 41.4, pp. 297–311. DOI: 10.4173/mic.2020.4.2.

He, Bin, Yangang Wang, Wei Song and Wen Tang (Oct. 2014). 'Design resource management for virtual prototyping in product collaborative design'. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 229.12, pp. 2284–2300. DOI: 10.1177/0954405414551106.

Humphrey, W.S. (Mar. 1988). 'Characterizing the software process: a maturity framework'. In: *IEEE Software* 5.2, pp. 73–79. DOI: 10.1109/52.2014.

Ianagui, André Seiji Sandes (2019). 'Robust system design for consensus control in dynamically positioned vessel fleet.' PhD thesis. DOI: 10.11606/t.3.2020.tde-07012020-155948.

Ildstad, Jens B., Thomas M. Kolstad and Karl H. Halse (June 2017). 'Comparison of a Simplified Vessel Response Estimation With a State of the Art Vessel Response

Prediction Computer Tool'. In: *Proceedings of the 36th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Trondheim, Norway: American Society of Mechanical Engineers. DOI: 10.1115/omae2017-61530.

Industry Foundation Classes (IFC) (2022). *IFC.js - BIM toolkit for JavaScript*. Accessed: 2022-04-16. URL: https://ifcjs.github.io/info/.

ISO (Nov. 2004a). *10303-218 Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 218: Application Protocol: Ship structures*. International Organization for Standardization, Geneva, Switzerland.

— (July 2004b). *15926-1:2004 Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 1: Overview and fundamental principles*. International Organization for Standardization, Geneva, Switzerland.

— (Nov. 2015). *19901-1 Petroleum and natural gas industries – Specific requirements for offshore structures – Part 1: Metocean design and operating considerations*. International Organization for Standardization, Geneva, Switzerland.

— (Nov. 2017). *14306:2017 Industrial automation systems and integration — JT file format specification for 3D visualization*. International Organization for Standardization, Geneva, Switzerland.

— (Oct. 2018a). *19847 Ships and marine technology — Shipboard data servers to share field data at sea*. International Organization for Standardization, Geneva, Switzerland.

— (Oct. 2018b). *19848 Ships and marine technology — Standard data for shipboard machinery and equipment*. International Organization for Standardization, Geneva, Switzerland.

ISO/IEC (June 2016). *20922 Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1*. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) Joint Technical Committee 1.

Japan Ship Machinery and Equipment Association (JSMEA) (2022). *Description of Local ID definitions — jsmea-codebook*. Accessed: 24-04-2022. URL: https://www.jsmea.or.jp/ssap/topics/jsmea_codebook.html.

Jensen, Jørgen Juncher, Alaa E Mansour and Anders Smærup Olsen (Jan. 2004). 'Estimation of ship motions using closed-form expressions'. In: *Ocean Engineering* 31.1, pp. 61–85. DOI: 10.1016/S0029-8018(03)00108-2.

Jung, Ralf (2020). 'Understanding and evolving the Rust programming language'. en. PhD thesis. Universität des Saarlandes. DOI: 10.22028/D291-31946.

Khronos Group (2022). *glTF Overview - The Khronos Group Inc*. Khronos Group - Connecting Software to Silicon, Oregon, United States of America. Accessed: 2022-04-16. URL: https://www.khronos.org/gltf/.

Låg, Steinar, Vidar Vindøy and Kristian Ramsrud (2021). 'A Standardized Sensor Naming Method to Support Digital Twins and Enabling New Data Driven Applications in the Maritime Industry'. In: *13th Symposium on High-Performance Marine Vehicles, HIPER*. Tullamore, Ireland.

Låg, Steinar and Silje Brathagen With (Aug. 2017). *Standardisation as an enabler of digitalisation in the maritime industry*. Group Technology & Research, Position Paper. DNV GL. Høvik, Norway.

Levander, Kai (May 1991). 'System-Based Passenger Ship Design'. In: *The 4th International Marine Systems Design Conference (IMSDC'91)*. Vol. 2. Discussion about Invited Lecture (3). Kobe, Japan.
— (2012). 'System based ship design'. In: *NTNU Marine Technology.(SeaKey Naval Architecture)*.
Loisel, Sébastien (2022). *Numeric.js*. Accessed: 2022-04-16. URL: https://github.com/sloisel/numeric.
MathWorks (2013). *MATLA 8.2 (R2013b)*. Natick, Massachusetts.
Mello, Pedro Cardozo de (2012). 'Sistema de automação e controle para tanques oceânicos com múltiplos atuadores'. PhD thesis. DOI: 10.11606/t.3.2012.tde-19112012-123823.
Miquel, Sergi Escamilla i, Ícaro Aragão Fonseca, Henrique Murilo Gaspar and Daniel Prata Vieira (July 2020). 'An open-source library for hydrodynamic simulation of marine structures'. In: *Marine Systems & Ocean Technology* 15.3, pp. 160–174. DOI: 10.1007/s40868-020-00083-3.
Mukti, Muhammad Hary (2022). 'A Network-Based Design Synthesis of Distributed Ship Services Systems for a Non Nuclear Powered Submarine in Early Stage Design'. PhD thesis. University College London (UCL).
Mukti, Muhammad Hary, Rachel Jean Pawling and David J Andrews (July 2021). 'Distributed Ship Service Systems Architecture in The Early Stages of Designing Physically Large and Complex Vessels: The Submarine Case'. In: *International Journal of Maritime Engineering* 163.A2. DOI: 10.5750/ijme.v163ia2.755.
Myhre, Torstein A. (2022). *Spherical Pendulum Dynamics*. Accessed: 2022-04-16. URL: https://www.torsteinmyhre.name/snippets/spherical_pendulum.html.
Newman, John N. (2001). *Wave effects on multiple bodies*. Tech. rep. RIAM, Kyushu University.
Nokkala, Tiina, Hannu Salmela and Jouko Toivonen (2019). 'Data Governance in Digital Platforms'. In: *25th Americas Conference on Information Systems, AMCIS 2019, Cancún, Mexico, August 15-17*. URL: https://aisel.aisnet.org/amcis2019/ebusiness/ebusiness/12.
Nordby, K, E Gernez and S Mallam (2019). 'OpenBridge: designing for consistency across user interfaces in multi-vendor ship bridges'. In: *Ergoship conference, Haugseund, Norway*.
Nowacki, Horst (Nov. 2010). 'Five decades of Computer-Aided Ship Design'. In: *Computer-Aided Design* 42.11, pp. 956–969. DOI: 10.1016/j.cad.2009.07.006.
npm (2022). *npm*. Accessed: 2022-07-03. URL: https://www.npmjs.com/.
OCX Consortium (2022). *The Open Class 3D Exchange Format*. Accessed: 2022-05-31. URL: https://3docx.org/.
Oliveira, Felipe Ferrari de (2019). *Implementation of Open Source Code for 6 Degrees of Freedom Simulations in Maritime Applications*. Tech. rep. NTNU in Ålesund, Norway: Ship Design and Operation Lab.
— (2021). 'An open web platform aimed at ship design, simulation and digital twin'. MSc dissertation. Norwegian University of Science and Technology.
— (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*. Ac-

cessed: 2022-09-06. URL: `https://observablehq.com/collection/@ferrari212/from-hull-to-simulation`.

Pal, Malay (Sept. 2015). 'Ship work breakdown structures through different ship lifecycle stages'. In: *International Conference on Computer Applications in Shipbuilding*. Bremen, Germany.

Parsons, M. G. (2004). 'Ship Design and Construction, Vol. 1'. In: ed. by Thomas Lamb. Society of Naval Architects and Marine Engineers (SNAME). Chap. 11 - Parametric Design.

Qi, Qinglin, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang and A.Y.C. Nee (Jan. 2021). 'Enabling technologies and tools for digital twin'. In: *Journal of Manufacturing Systems* 58, pp. 3–21. DOI: `10.1016/j.jmsy.2019.10.001`.

Rachuri, Sudarsan, Eswaran Subrahmanian, Abdelaziz Bouras, Steven J. Fenves, Sebti Foufou and Ram D. Sriram (July 2008). 'Information sharing and exchange in the context of product lifecycle management: Role of standards'. In: *Computer-Aided Design* 40.7, pp. 789–800. DOI: `10.1016/j.cad.2007.06.012`.

Rittel, H. (1982). 'Systems Analysis of the 'First and Second Generations''. In: *Human and Energy Factors in Urban Planning: A Systems Approach*. Ed. by P. Laconte, J. Gibson and A. Rapoport. NATO Advanced Study Institutes Series (ASID) Series D: Behavioural and Social Sciences - No. 12. Dordrecht: Springer Netherlands, pp. 35–52. ISBN: 978-94-009-7651-1. DOI: `10.1007/978-94-009-7651-1_4`.

Robinet, Fabrice, Rémi Arnaud, Tony Parisi and Patrick Cozzi (Dec. 2018). 'glTF: Designing an Open-Standard Runtime Asset Format'. In: *GPU Pro 360 Guide to 3D Engine Design*. A K Peters/CRC Press, pp. 243–260. DOI: `10.1201/9781351172486-20`.

Sako, Mari (Apr. 2022). 'Global supply chain disruption and resilience'. In: *Communications of the ACM* 65.4, pp. 18–21. DOI: `10.1145/3517216`.

Sandvik, Bjorn (2022). *three.geo: Geospatial data support in three.js*. Accessed: 2022-06-04. URL: `https://github.com/turban/three.geo`.

Shafto, Mike, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne and Lui Wang (2010). 'Draft modeling, simulation, information technology & processing roadmap'. In: *Technology Area* 11.

Siemens (2019). *JT File Format Reference - Version 10.5 - Rev-A*.

Simon, Herbert A (1987). 'The Steam Engine and the Computer: What Makes Technology Revolutionary.' In: *Educom Bulletin* 22.1, pp. 2–5.

Smith, Barry (2006). 'Against idiosyncrasy in ontology development'. In: *Frontiers in Artificial Intelligence and Applications* 150, pp. 15–26.

Soal, K., Y. Govers, J. Bienert and A. Bekker (Nov. 2019). 'System identification and tracking using a statistical model and a Kalman filter'. In: *Mechanical Systems and Signal Processing* 133, p. 106127. DOI: `10.1016/j.ymssp.2019.05.011`.

Szykman, Simon, Steven J. Fenves, Walid Keirouz and Steven B. Shooter (June 2001). 'A foundation for interoperability in next-generation product development systems'. In: *Computer-Aided Design* 33.7, pp. 545–559. DOI: `10.1016/s0010-4485(01)00053-7`.

Taylor, Nicole, Carlo Human, Karel Kruger, Anriëtte Bekker and Anton Basson (2020). 'Comparison of Digital Twin Development in Manufacturing and Maritime Domains'.

In: *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*. Ed. by Theodor Borangiu, Damien Trentesaux, Paulo Leitão, Adriana Giret Boggino and Vicente Botti. Cham: Springer International Publishing, pp. 158–170. ISBN: 978-3-030-27477-1.

Three.js authors (2022). *Three.js - JavaScript 3D Library*. Accessed: 2022-04-16. URL: https://threejs.org/.

three.s - IFC loader (2022). Accessed: 2022-10-01. URL: https://threejs.org/examples/webgl_loader_ifc.html.

V8 project authors (2022). *V8 JavaScript engine*. Accessed: 2022-04-16. URL: https://v8.dev/.

Vieira, Daniel P., Edgard B. Malta, Fabiano P. Rampazzo, Joa~o Luis B. Silva and Eduardo A. Tannuri (Jan. 2011). 'Effects of Coupled Hydrodynamic in the Performance of a DP Barge Operating Close to a FPSO'. In: *Proceedings of the 30th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Rotterdam, The Netherlands. DOI: 10.1115/omae2011-49411.

Villani, Cédric, Yann Bonnet, Bertrand Rondepierre et al. (Mar. 2018). *For a meaningful artificial intelligence: towards a French and European strategy*. Conseil national du numérique.

Vindøy, Vidar (Apr. 2008). 'A functionally oriented vessel data model used as basis for classification'. In: *7th International Conference on Computer and IT Applications in the Maritime Industries, COMPIT*. Liège, Belgium.

W3Schools (2022). *Java OOP (Object-Oriented Programming)*. Accessed: 2022-07-03. URL: https://www.w3schools.com/java/java_oop.asp.

Wamit (2022). *Wamit, Inc. - The State of the Art in Wave Interaction Analysis*. Accessed: 2022-05-04. URL: https://www.wamit.com/.

Warren, Tom (May 2019). *Inside Microsoft's surprise decision to work with Google on its Edge browser*. The Verge, Vox Media. Accessed: 2022-04-16. URL: https://www.theverge.com/2019/5/6/18527550/microsoft-chromium-edge-google-history-collaboration.

Washio, Yushu, Masao Miyoshi, Katsuyoshi Takekuma, Kenji Yamada and Kazuo Kobayashi (Jan. 1994). 'Recent Research and Development in the Design of an Oceanographic Research Vessel'. In: *Marine Technology and SNAME News* 31.01, pp. 1–19. DOI: 10.5957/mt1.1994.31.1.1.

West, Timothy D and Mark Blackburn (2017). 'Is Digital Thread/Digital Twin Affordable? A Systemic Assessment of the Cost of DoD's Latest Manhattan Project'. In: *Procedia computer science* 114, pp. 47–56.

— (2018). 'Demonstrated Benefits of a Nascent Digital Twin'. In: *INSIGHT* 21.1, pp. 43–47.

WHATWG (2022). *WHATWG - Standards*. Accessed: 2022-04-16. URL: https://html.spec.whatwg.org/.

Whitfield, R. I., A. H. B. Duffy and J. Meehan (2003). 'Ship product modeling'. In: *Journal of Ship Production* 19.4, pp. 230–245.

Whitfield, R.I., A.H.B. Duffy, P. York, D. Vassalos and P. Kaklis (May 2011). 'Managing the exchange of engineering product data to support through life ship design'. In: *Computer-Aided Design* 43.5, pp. 516–532. DOI: 10.1016/j.cad.2010.12.002.

Wittern, Erik, Philippe Suter and Shriram Rajagopalan (May 2016). 'A look at the dynamics of the JavaScript package ecosystem'. In: *Proceedings of the*. Austin, Texas, USA: ACM. DOI: 10.1145/2901739.2901743.

World Wide Web Consortium (W3C) (2022). *Cascading Style Sheets home page*. World Wide Web Consortium (W3C). Accessed: 2022-04-16. URL: https://www.w3.org/Style/CSS/Overview.en.html.

Xantic (2001). *SFI Group System - A system for classification of technical and economic ship information*. Product Description.

Yara International (2022). *Yara Birkeland*. Accessed: 2022-09-06. URL: https://www.yara.com/news-and-media/press-kits/yara-birkeland-press-kit/.

Zaman, Ibna, Kayvan Pazouki, Rose Norman, Shervin Younessi and Shirley Coleman (2017). 'Challenges and Opportunities of Big Data Analytics for Upcoming Regulations and Future Transformation of the Shipping Industry'. In: *Procedia Engineering* 194, pp. 537–544. DOI: 10.1016/j.proeng.2017.08.182.

Zijl, C. van, K. Soal, R. Volkmar, Y. Govers, M. Böswald and A. Bekker (Sept. 2021). 'The use of operational modal analysis and mode tracking for insight into polar vessel operations'. In: *Marine Structures* 79, p. 103043. DOI: 10.1016/j.marstruc.2021.103043.

Zuboff, Shoshana (2019). 'The age of surveillance capitalism: The fight for a human future at the new frontier of power'. In: PublicAffairs Books. Chap. 3 - The Discovery of Behavioral Surplus.

Paper I

Ícaro Aragão Fonseca and Henrique Murilo Gaspar (Sept. 2020). 'Challenges when creating a cohesive digital twin ship: a data modelling perspective'. In: *Ship Technology Research* 68.2, pp. 70–83. DOI: `10.1080/09377255.2020.1815140`

Ship Technology Research
Schiffstechnik

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ystr20

Challenges when creating a cohesive digital twin ship: a data modelling perspective

Ícaro Aragão Fonseca & Henrique Murilo Gaspar

SHIP TECHNOLOGY RESEARCH (SCHIFFSTECHNIK)
2021, VOL. 68, NO. 2, 70–83
https://doi.org/10.1080/09377255.2020.1815140

REVIEW

OPEN ACCESS Check for updates

Challenges when creating a cohesive digital twin ship: a data modelling perspective

Ícaro Aragão Fonseca and Henrique Murilo Gaspar

Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Ålesund, Norway

ABSTRACT
A digital twin is a digital asset that simulates the behaviours of a physical counterpart. Digital twin ship literature identifies that the concept is already being applied to specialised problems, but no clear guide exists for creating broader interdisciplinary digital twins. Relevant dimensions of product data modelling and previous attempts at standardizing ship data elucidate the requirements for effective data modelling in a digital twin context. Such requirements are placed in a broader perspective for digital twin implementation that encompasses challenges and directions for future development of services, networks, and software. Finally, an open standardization for digital twin data is proposed based on lessons extracted from this panorama, proposing its application to a research vessel.

ARTICLE HISTORY
Received 30 April 2020
Accepted 17 August 2020

KEYWORDS
digital twin; digital thread; simulation; product data modelling; data standard; open source; internet of things; digitalization

1. Origins, definition and purpose of the digital twin

A focus on digitalization of the maritime industry has been increasing significantly, with new technologies expected to support faster completion of processes and data use during decision-making in the maritime value chain. The concept of digital twin aligns with this overall trend. Boschert and Rosen (2016) trace the origins of the digital twin to the aerospace industry, in which replicas of complex physical systems were commonly constructed, as, for example, during NASA's Apollo space programme or by Airbus with its Iron Bird test rigs. Before system deployment, such replicas can be used to test systems integration and train crew members. During operational phases, engineers can use them to simulate operational alternatives and study issues that appear on a working aircraft by mirroring its behaviour. More recently, advances in simulation methods for engineering are expected to the enable reproduction of these practices using digital simulations, thus conceiving a digital twin system. In the early 2000s appeared the first mentions the possibility of extending product lifecycle management (PLM) platforms with data collected from the physical product in order to mirror it with the virtual counterpart (Grieves and Vickers 2016). At the same time, simulations were already used to support the operation of physical systems, even if with a relatively narrow scope. Cameron et al. (2018) cite some examples in the oil and gas sector which are analogous to a digital twin of a multiphase pipeline in the context of a broader oil and gas installation, or even to the digital twin of a valve, with components including sensors and actuators.

The vision established by NASA in a 2010 draft report (Shafto et al. 2010, p. 18) has greatly influenced the general perception of a digital twin.[1] The report outlines the concept as 'an integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin.' The report lists vehicle systems modelled by a digital twin and its final role played in supporting mission success. As West and Blackburn (2017) note, the vision of the digital twin described in the report is ambitious, with descriptions of 'ultra-realistic' integrated models that are so detailed that they accurately represent an aircraft's 'manufacturing anomalies,' while remaining suitable to conduct simulations that assist operations continuously. In a later work (West and Blackburn 2018), they argue that while the digital twin concept on such moulds is impractical to implement fully in the following decades, it can still facilitate the more streamlined system sustainment even without achieving a perfect degree of realism.

It is thus critical to identify ways the concept can prosper, even considering its limitations, with the intention of reaping some expected benefits. The first step is recognizing that despite its grandiosity, the digital twin concept revolves around a central principle –

CONTACT Ícaro Aragão Fonseca icaro.a.fonseca@ntnu.no

[1]The Defense Acquisition University (DAU) glossary (2020) from the United States Department of Defense (DoD) defines a digital twin similarly, with the addition that it should not only mirror asset behaviors but predict them.

mirroring and assessing a physical asset's constitution and behaviours using a simulated counterpart. This principle is reflected in the four expected use cases of the digital twin specified in the NASA report – simulate a mission before it is executed, mirror the behaviours of its physical twin during operations, perform in-situ forensics of a potentially catastrophic fault or damage, and serve as a platform for studying the effects of modifications in mission parameters that were not considered during design. These use cases are similar to the physical system replicas discussed previously; they share the purpose of supporting decision-making during operations, either in (near) real-time or pre-emptively. The digital twin represents a tool for verification and validation of system behaviours, and the study and debugging of operational problems. After the aerospace sector put the digital twin forward, it has been adopted by an array of other industries. This process has been aided by developments in industrial Internet of Things (IoT), sensor technology and miniaturization, which contribute to obtain and store measurements as digital data (Figure 1).

2. Examples of digital twin ship initiatives

With the popularization of digital twins, the first applications to the ship domain have started to appear. Digital twin ship implementations found in the literature accord with the potential applications discussed above, and they can be generally clustered into two main groups. The first is decision support for ship operations, with a focus on condition monitoring and calibration of simulation models based on real operational data. Coraddu et al. (2019) estimate speed loss caused by marine fouling using a simulation model based on a neural network; the network receives data measured from a vessel and returns an estimate of speed loss. The tools demonstrated superior performance in comparison to the ISO standard for estimating fouling. Given this method based on machine learning, the proposed model requires considerable amounts of data.

Schirmann et al. (2019) present a digital twin for ship motion and estimation of structural fatigue due to wave response. Given weather forecast data for a given route, the digital twin estimates expected cumulative damage the ship would endure. Different from the previous example, the authors used specialised formulas, not machine learning, to simulate ship behaviours. Danielsen-Haces (2018) apply a digital twin to autonomous vessels, in this case, a ship model built for research. The digital twin has two use cases – condition monitoring and calibration of the propulsion system simulation models based on operational data. Bekker (2018a) details plans to implement a digital twin of a polar supply and research vessel, based on a comprehensive sensor infrastructure installed previously on the vessel. Figure 2 shows that the plan covers aspects from context, such as waves and ice, and from ship states, such as a rigid body and structural responses to waves, and the effects of motion on human factors. Sensor readings will be processed using data analysis techniques such as machine learning, with initial applications already yielding positive results (Bekker et al. 2018b).

The second group includes digital twins used as tools for system integration testing and personnel training. Tofte et al. (2019) describe a system for emulation of control systems, in which a detailed simulation model of the lay tower clamp in a pipe-laying vessel is linked to controllers to include hardware and a human in the loop system, which can be used for hardware testing and operation training. Dufour et al. (2018) also discuss applications of a digital twin for system integration testing of naval ship power systems with hardware in the loop.

3. Digital twin ship content and usage

3.1. Digital twin data

Given the systemic complexity of vessels, it is natural that most current digital twin implementations do not yet include overarching, integrated digital twins; they are usually in early stages of development or were created to address specific problems. Similarly, commercial solutions from software vendors are

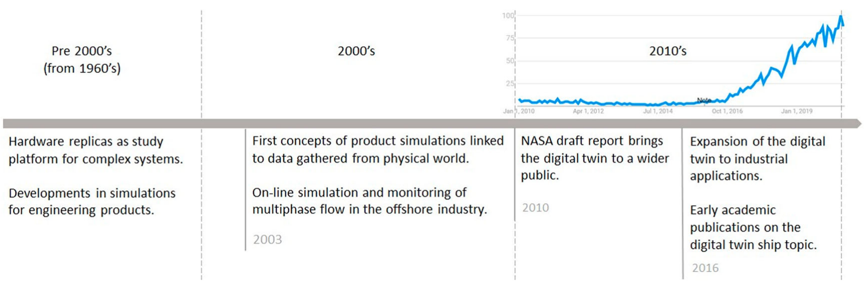


Figure 1. Timeline of development of the term 'digital twin' – the term as we know appeared in the last decade, depicted by the chart with the normalized quantity of Google searches on the top right.

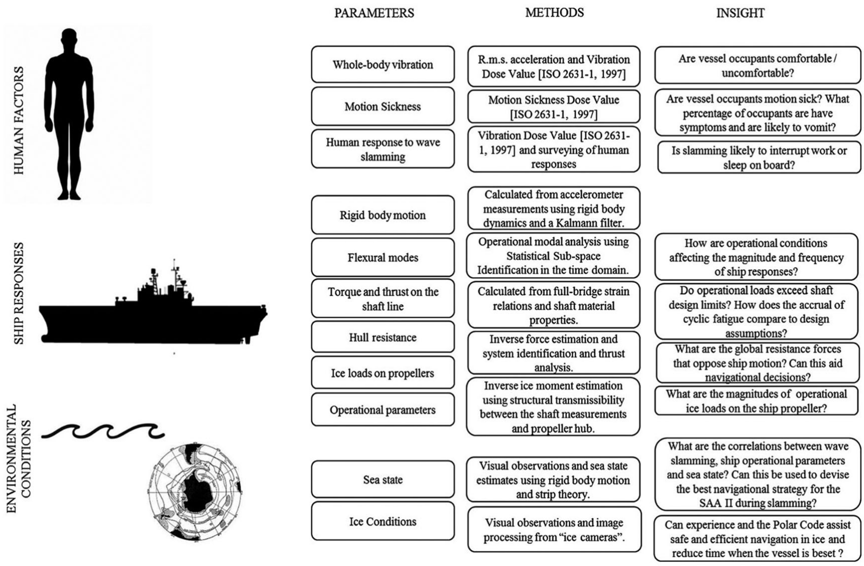


Figure 2. Measured parameters, analysis methods and expected insights for the proposed digital twin by Bekker et al. (2018b).

comprehensive ship representations, but functionalities for monitoring and simulation of ship behaviours during operation are comparatively new. Attempts to characterise the data content of digital twins usually converge on a typology based on three constituents – asset representation, behavioural models, and measured data (Cabos 2018; Cameron et al. 2018). The last category can be broken down into data measured from the asset itself and that from its operational context (Erikstad 2017). Figure 3 illustrates how this data interacts to realise different digital services, with the digital twin working as a central hub giving access them. The collected data must align closely to interact efficiently, and thus the exact contents of a service implementation depends on its domain.

The production and use of data in each of those groups occur differently throughout an asset's lifecycle. Simulation models that define an asset's constitution and behaviours also define permanent aspects of the asset, and states and environmental contexts are much more transitory because they are perceived in real-time during operation as data are gathered from sensors and other perceptual devices. Given that distinction, models of assets and behaviours should be reused from previous lifecycle stages, thus establishing a digital thread of data use over the product's life.

3.2. Asset representation

A ship's representation commonly revolves around a 3D model, which can be assembled by aggregating several CAD files generated during the ship's design, such as the hull, structure, and outfitting modelled by designers, complemented by machinery and component models acquired from third parties. A digital twin might also include 3D models of these elements

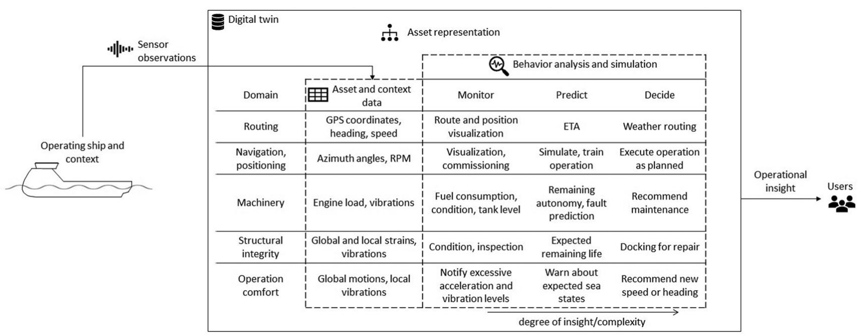


Figure 3. Digital twin elements and usage in the implementation of services in various domains.

stored as lightweight visualization formats that are adequate for rendering animations. To complete the physical representation, a 3D model must be complemented with metadata that describes weight distributions and material characteristics. The same metadata structure can be used to organise component descriptions, maintenance comments, and other accessory information.

3.3. Behaviour models

The behaviour model establishes a bridge between a digital asset representation and the physical reality measured as data. This bridging occurs in ways that depend on the purpose of the digital twin; a simulation can feed directly from sensor log streams to provide real-time support, or it can rely on static (i.e. stored) data to analyse current asset conditions and performance on previously executed operations. Such models might employ various methods to achieve these goals, whether based on physics or statistics, including signal processing and machine learning (Erikstad 2018). Erikstad (2017) presents a progression from behaviour modelling, as commonly used in engineering design analysis, to operation support, as proposed by the digital twin, emphasising the importance of sensor input in distinguishing typical engineering analyses and simulations from a digital twin. Thus, context and behaviour in the digital twin are not only modelled but measured to obtain insights about a ship's operating performance. This means that if such behavioural models are to be reused throughout the product's lifecycle, some changes to the current engineering design methods might be necessary. Simulation models created during design must be conceived to not only analyse the product's behaviour under prescribed conditions but support its operation with services such as validation of the designed product and forecasting of various operational situations. This ultimately creates the potential to analyse operational data to aid design decisions of new vessels, thus closing the loop of data use during the lifecycle.

3.4. Measured data

Data describe an asset's state in its operating context, which can be accomplished using sensor logs, reports that describe its physical condition, and other data. With advances in connected devices and remote monitoring technology, the novel value introduced by a digital twin will be extracting insights from sensor data rather than being an archival system for written reports. To ensure that a digital twin realises that value, considering a few issues when planning an effective sensor setup is required. When mirroring the asset's operating state, it is important to acknowledge factors regarding its context that influence measured behaviours. One important concept is the distinction between raw and net data. Not all sensor logs, or raw data, will be immediately useful for a digital twin, so it might be necessary to perform post-processing to extract physical meaning from it (i.e. the net data). For example, consider a digital twin designed to measure the motion response of a vessel that is operating near an oil platform. The setup must measure not only vessel motion, but wave characteristics in its geographic context. If the wave is measured using a buoy, it is necessary to employ algorithms that extract period and significant wave heights from buoy elevation measurements. Similarly, it might be necessary to decompose readings of the motion sensors installed on the vessel into motion components on six degrees of freedom. These implications highlight the importance of purposeful state measuring on digital services, which we discuss later.

4. Approaches to handling of ship data

4.1. Data modelling and usage in the ship industry

Traditionally, digital management of a ship's lifecycle relies on many specialised software tools that produce discrete solutions to their respective problems, rarely influenced by interoperability. Several factors lead to this scenario, some of which are common to other engineering domains. One example is the perception of proprietary data formats as a competitive advantage by adversarial software providers (Rachuri et al. 2008). Other factors are specific to the ship industry. In comparison to other engineering disciplines, the ship industry represents a small segment of potential computer-aided design (CAD) customers, an obstacle to justification of big investments in software development (Gaspar 2019). Since the ship industry usually focuses on individual production, its data management is commonly based on tenders, each of which in turn is complemented by an entire framework of choices and variables that influence a vessel's characteristics, and thus its digital representation, throughout the lifecycle. Ship data is, therefore, not necessarily organised systematically, and even less so when the organization is compared among ships, design offices, and yards (Gaspar 2018). Although discussed particularly in the context of ship design up to commissioning, this variability extends to subsequent operational stages, i.e. the domain of interest for a digital twin.

In the absence of an established ship model standard in the industry, alternatives for modelling product data are few and niche, examples of which include some formats and libraries for CAD and 3D visualization (e.g. IGES, STEP, and JT). Gaspar (2019) describes the current scenario of ship software integration as consisting of two trends – suites of PDM and PLM systems by major software providers that offer tools for a ship's lifecycle and specialised tools that allow more flexible connections with such suites. The following sections discuss some extant approaches proposed to model ship data, which are mentioned as a reference point for subsequent examination of how a multidisciplinary digital twin can be modelled and developed.

4.2. Principles of product data modelling applied to digital twins

We begin by comparing the digital twin asset representation to a typical characterization of product model during design. van den Hamer and Lepoeter (1996) decompose the problem of managing product data into five dimensions – views and hierarchy of the designed system, version and status of design data, and variants of the designed product (Figure 4). The tender-based characteristic of the ship industry drastically reduces the importance of the variants dimension; presenting a set of designs as a product family is too preliminary to displace the practice of managing ship data individually. Other dimensions might play a role during a ship's design stage, but they are less relevant in the context of the digital twin. Versioning, for example, accounts for the modification and evolution of design information. Since a digital twin models the asset representation as consolidated after the building stage, it does not need to accommodate multiple versions of the product model. At most, the dimension plays a role in archiving instrumentation logs and other documents collected during operation. The status version tracks validity and consistency of information, identifying valid and superseded versions and controlling change propagation across the digital model. Since the necessity of tracking validity of the digital twin data itself is small or absent, eventual use of the dimension will be to describe the physical status of the asset (i.e. indicating that a component is functioning as expected or that it requires maintenance).

The two other dimensions, views and hierarchy, are central to a ship's lifecycle generally and to the digital twin specifically. Given their importance, they represent recurring themes in the literature, also appearing under the name taxonomies (Otto et al. 2016; Låg and With 2017; Gaspar 2018). This term is used to describe both the various perspectives of a ship and the hierarchical breakdown that organises the data under that view. For example, Siemens presents a 4th generation design (4GD) approach in its PLM package for shipbuilding (Siemens PLM Software 2013; Levisauskaite 2016) that promises to manage design elements independently using a flat structure. A design element contains the global position, CAD geometry, and the lifecycle data of a part, with which a user is able to filter sets of design elements according to rules, for example, by system, location, or attribute. Siemens argued that due to these characteristics, the 4GD approach is able to handle multiple taxonomies without duplication of data, reduce required storage space for files, and provide better support to concurrent modification of parts.

4.3. The SFI group and classification system

One example of an early and successful taxonomy for organization of ship data is the SFI group and classification system, which today is licensed commercially by SpecTec (Xantic 2001). The system was developed as a response to the challenges of exchanging data consistently inside and among organizations. As electronic data-processing technologies were introduced in the industry, corporate players searched for a standardised

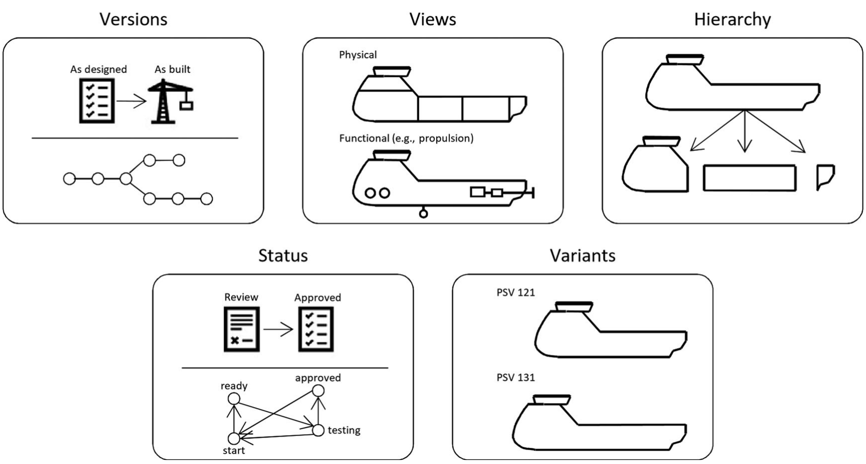


Figure 4. Product data management decomposed into five dimensions – versions, views, hierarchy, status, and variants. Adapted from van den Hamer and Lepoeter (van den Hamer and Lepoeter 1996).

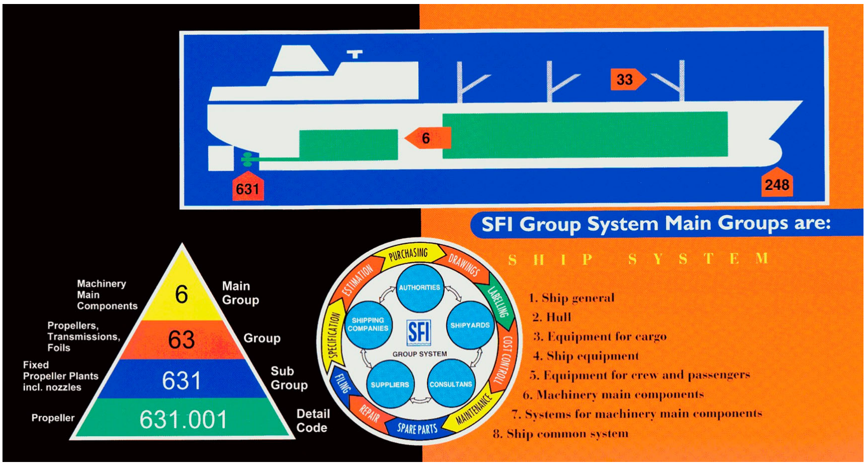


Figure 5. SFI summary with use cases, main groups in the hierarchical system, and an example of a tag system applied to a propeller component (Xantic 2001).

solution for data handling that could be adapted to yards and ships and that was understandable to humans (Manchinu and McConnell 1977). Development was conducted by a consortium of private shipyards and the Ship Research Institute of Norway, currently SINTEF. The first tests were performed using the system as an on-board maintenance code for types of ships, a system released in 1972. Its use cases centred on indexing and identification of drawings and specifications, and control and accounting of parts, work, and materials (Figure 5). The SFI group system is based on a hierarchical, numeric tag system that is guided by a strictly functional view of the ship that indexes components not by system, but by groups according to component function. The numeric tag system consists of three levels that are supplemented by a detail code for individual components and materials. This hierarchy is mapped to the indexed information, including drawings, specifications, and accounting registers for material and labour.

4.4. Standard for the exchange of product model data

Another standardised approach to ship product model data is the ISO 10303 standard, informally known as STEP (Standard for the Exchange of Product Model Data). Whitfield et al. (Whitfield et al. 2003) conducted a literature review on the topic of ship product modelling, identifying STEP as the most significant development by far in that area. By standardizing data for exchange among software systems, especially pre-commissioning phases of the lifecycle, STEP was expected to establish the basis required for an ecosystem of heterogeneous tools to flourish. Development began in 1984, and the standard was released a decade later. In the following years, five application protocols (APs) were released specifically for ship data, covering arrangements, moulded forms, piping, structures, and mechanical systems (Figure 6[2]). One AP included an application activity model that described the intended process for where the standard would be used, an application reference model that described information requirements, and an application-interpreted model that described schema with which the modelled data should comply. The schema was based on entity types, which are analogue to objects in object-oriented systems, that use concepts such as property and inheritance.

In practice, STEP's adoption was rare in the shipbuilding and other industries, falling short of its original purpose of becoming a de-facto standard. Gielingh (2008) noted that STEP had its use limited to the exchange of 3D CAD models, clustering the reasons for low adoption into three groups – business models, legal aspects, and industrial readiness. One obstacle was that competing software providers lacked interest in complying with standards, due in part to an asymmetry in which vendors that invested in standard compliance would not reap the benefits of such compliance. Besides reasons for low adoption, research reported poor technical performance of the standards that occurred due to differences in CAD representations, information scope, and entity-oriented schema. These differences increased the change of information losses when exchanging data among applications. For example, attempting to enable data

[2]Figure 6. Diagram of STEP standards and Figure 8. Possible inputs and services of a shipboard data server are reproduced by Ícaro A. Fonseca in Challenges when Creating a Cohesive Digital Twin Ship: A Data Modelling Perspective under licence from Standard Online AS July 2020.

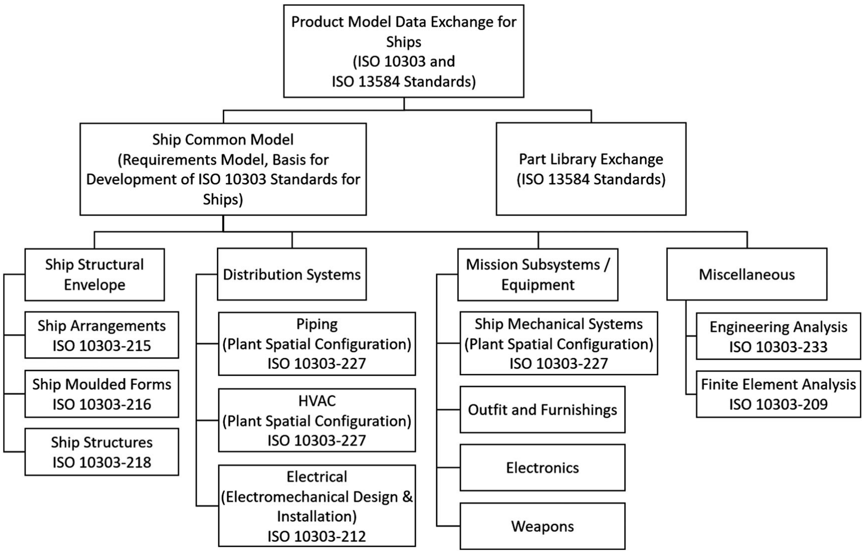


Figure 6. Diagram of STEP standards applicable to the ship product model according to ISO (ISO 2004). The APs include both specialised standards for ships and general standards that can be applied to that domain.

exchange among multiple ship design programmes, Whitfield et al. (2011) chose a custom neutral model instead of STEP after identifying that the latter would disallow a complete exchange of digital content among extant software.

In retrospect, business factors appear to have had great influence over the success of product data technologies. Consider construction, another tender-based sector that delivers large, complex systems. Industry Foundation Classes standards, the basis for building information modelling (BIM), have been gaining significant adoption in the industry, allowing interoperability among software packages. From a technical viewpoint, IFC shares many similarities with STEP, also relying on entity-oriented schema. IFC standards even include elements of STEP in its composition, such as some APs and the information modelling language EXPRESS, which is commonly perceived as complicated and technically limited (Whitfield et al. 2011; Cameron et al. 2018). From a business viewpoint, development of IFC was vendor-driven more than STEP was (Gielingh 2008), and adoption of IFC has been aided by the public sector in some European countries, which now require compliance with IFC from companies bidding to participate in the construction of public projects.

5. Towards a cohesive digital twin ship

5.1. A broader picture

The comparison between standardization approaches in the previous section leads us to argue that models for ship data must be supported by matching business models and must be motivated by parties interested in implementing them. In the case of a digital twin ship, these challenges are compounded by the presence of multiple stakeholders (e.g. component suppliers, yards, and ship operators) and new requirements when using the digital model to interact with sensor streams gathered from operation in real-time. Figure 7 organises these concerns into three layers, from a higher service and business level, downstreaming to network infrastructure, and finally to software. Each of these layers influences the chances of establishing a successful, cohesive digital twin based on standardised data models that can be reused across tools. The following sections discuss not only challenges that each layer imposes, but directions for overcoming them found in the literature regarding digital twins and digital services generally.

5.2. Services and business

Regarding specific digital twin purposes, the possibilities of applying digital twin principles to operational problems have not yet been identified exhaustively. A great breadth of potential domains and approaches exist that are being gradually matched to supporting business models as these are discovered, tested, and operationalised. Nokkala et al. discuss problems with data governance in an environment of shared data use using a case study in which the authors interviewed employees of a shipyard and its collaborative network (2019). They found that despite that the interviewees

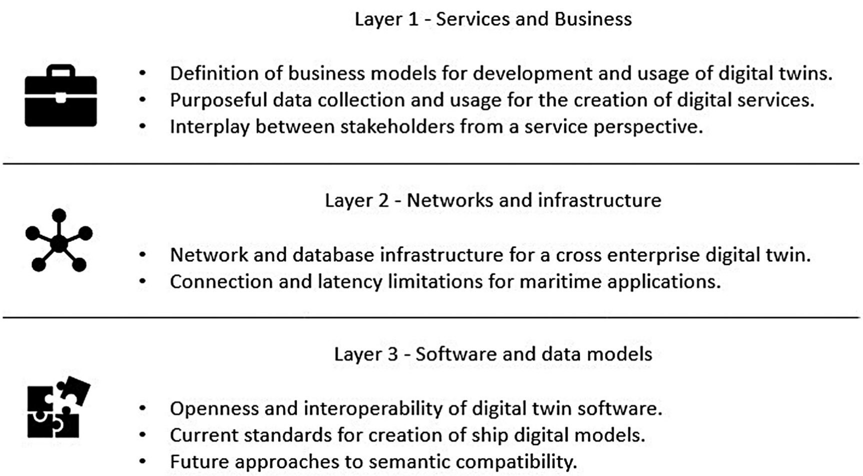


Figure 7. A division of challenges for implementation of a cohesive digital twin ship from higher to lower levels (layer 1–3, respectively).

recognised the potential business value of data use easily, there was strain between the necessity of defining a clear business model and the necessity of conducting exploration to identify ways in which the data could used. The interviewees mentioned the large amounts of data that are collected and stored by default during operation, even if without a clear business case; data are collected and stored 'just in case.' Erikstad (2019) corroborates this phenomenon, which is characterised by the contradiction that although there are large amounts of operational data available from sensors and instruments, there is not a clear use for them. As a result, development and implementation of many existing sensor-driven services have been opportunity – rather than needs – driven, and thus the author calls for purposeful design of digital services in which user needs regarding decision support for operations are traced to specific sensors installed on the vessel. Resolving these challenges might pass through creation of innovative business models among stakeholders, one of which could be shipbuilders transitioning to service providers that offer information and support to a ship and its digital twin after delivery (Van Os 2018). Morais and Goulanian (2019) offer the case of Ulstein, which started to provide solutions for integration among control, power management, and energy management systems for the vessels it builds.

5.3. Networks and infrastructure

Besides higher-level concerns regarding business models, the interplay among stakeholders will also move downstream to network infrastructures for data-sharing among parties. Bole et al. (2017) discuss that in such contexts, delivery of documents from the shipbuilder, the party that builds and integrates the vessel as a system, to the shipowner will evolve from being a formality to a crucial step during implementation of a digital twin system.

Given the niche aspect of digitalization in the maritime industry when viewed from a broader industrial perspective, it may be a wise strategy to adapt solutions from broader domains to the maritime context in order to reap benefits from adoption, support, and future developments. Rødseth and Berre (2018) propose Maritime Data Space (MDS) as an enabler of a vision of a digital twin linking databases from various stakeholders. MDS is an extension of the Industrial Data Space (IDS) framework developed by the Fraunhofer research organization with the goal of facilitating exchange of data in business ecosystems by using standards and common governance models (Otto et al. 2016). The IDS framework suggests a decentralised approach to data storage, while allowing parties to determine conditions for access to the data they own. IDS aims to assimilate platforms and services by establishing an app store and an open, neutral approach in which decisions are made jointly by the research team and users.

Another relevant initiative in that domain is the recently published ISO 19847 'Shipboard data servers to share field data at sea' (ISO 2018a). The document specifies requirements for servers collecting and sharing data from shipboard systems, which can be streamed to services such as condition monitoring, performance analysis and weather routing (Figure 8). The standard also describes concepts for the communication between the shipboard server and an onshore server for long-term data storage.

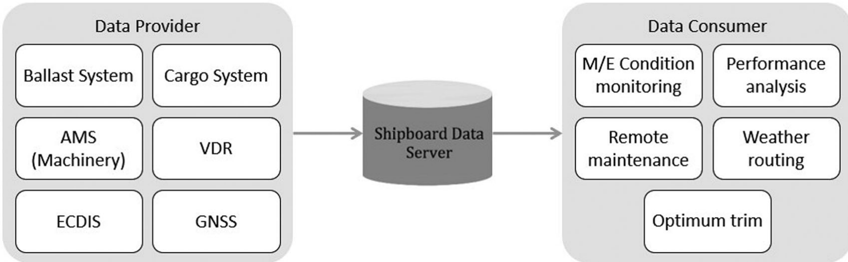


Figure 8. Possible inputs and services of a shipboard data server according to ISO 19847 (ISO 2018a).

Both the initiatives mentioned above take measures to deal with the peculiarities of ship operation such as system complexity and limited connectivity. Regarding the last point, Datta (2017) recommends that edge data processing could be prioritised in order to ease demands for latency and frequency of package delivery.

5.4. Software and data models

Recent advances in interoperability correspond with lessons learned from previous ship modelling. One example is the continuous rise in the perception of the importance of interoperability among digital systems, to a point at which even providers of proprietary solutions claim esteem for format and platform openness. Stachowski and Kjeilen (2019) discuss two CAD formats as examples of Siemens' willingness to promote openness – JT, which has become an ISO standard, and Parasolid, which is licensed to other companies and has had significant adoption. DNV GL (Sharma et al. 2017) present a preliminary project for digital twin platforms, Nauticus Twinity, as an open collaboration platform. Morais and Waldie (2018) emphasise the importance of approaches that allow interconnection of digital platforms such as open architectures, software modularization, and development of application programming interfaces (APIs).

Rachuri et al. (2008) classify PLM standards into four major types – type zero for implementation languages, type one for information modelling, type two for content standards, which can be further divided into sub-classifications, and type three for standards of architectural frameworks. Web-based solutions have been gaining popularity in engineering software such that HTML and JavaScript represent some of the most important type-zero standards currently in use. Stachowski and Kjeilen (2019) state the importance of Web technologies in the Siemens suite of PLM software. Bole et al. (2017) reported that AVEVA's digital twin technology is predominantly browser-based, and Schroeder et al. (2016) proposed a Web-based digital-twin architecture for monitoring offshore oil and gas platforms, including augmented reality visualization of the system. The advantages of Web services during development of digital twins include platform independence, compatibility with an array of devices and operational systems, preclusion of installation processes, and ease of deploying updated experiences to distributed users.

Regarding type-one standards, XML has been the traditional choice in maritime, but given the recent rise of importance of Web-based solutions, JavaScript Object Notation (JSON) offers a lightweight alternative for data exchange, better human readability, which allows users to interpret data and reduces the knowledge barrier for data manipulation, and broad support in the information technology industry. Adoption of type-zero standards ensures compatibility of software applications on executing devices, and type-one standards ensure that disparate software applications can parse the same information schema. To enable interoperability among applications, it is necessary to establish standards for the information content to be modelled, which means type-two standards. A modern example of a type-two standard is ISO 19848 'Standard for shipboard machinery and equipment data' (ISO 2018b). It was developed in companion with the ISO 19847 and describes sensor metadata such as variable type, unit of measurement, and update frequency both in JSON and XML schema. In the Norwegian maritime context, an continuing research consortium aims to apply the functional mock-up interface (FMI) to enable co-simulation among behavioural models of ship subsystems (Skjong et al. 2017; Hatledal et al. 2020).

Establishing further type-two standardization remains a challenge, but provision of APIs might lower the stakes of overcoming it. Such interfacing approaches might be the most feasible implementation alternative in the short- to medium-term, but they represent a compromise because they do not ensure interoperability among systems per se, but among their interfaces. The most ambitious solutions to this problem suggest transcendence of standardization from something modelled passively to something capable of adaption to enable communication between systems actively. Gielingh (2008) argued that given the burden of establishing common ontology templates for standardization of product data, it is desirable to search for templates that can explain themselves, rather than requiring systems to understand them. Similarly, given the tendency of information technology to evolve faster than standardization consortia are able to keep up with it, the author comments that it would be desirable that such standards provide dynamic features for updating. Datta (2017, pp. 23–24) argued that the difficulty in achieving communication among systems on a semantic level needs to be resolved on a higher cognitive level, where software can 'understand what needs to be understood' (e.g. assembling required communication APIs automatically) and hardware 'senses what needs sensing,' automatically activating and using relevant sensors to address the current problem.

6. Digital twin ship data standardization

6.1. Open digital twin platforms

Challenges identified in previous sections call for standards to enable systematic creation of digital twins that are suitable to modern data infrastructure and that attend to business and service requirements. As in extant Web services, a neutral core data standard for digital twins could represent the basis for an ecosystem of heterogeneous tools, allowing choices among various platforms and connection to external services and applications. This section outlines an approach to such a standard, focusing on the importance of selecting data views and hierarchies that align with the digital twin purpose. The standard provides a mapping between taxonomies and data content in a manner that facilitates understanding and use by humans and computer systems alike. So far, the proposed approach has been applied to a few simplified case studies, including a digital twin experiment performed with a platform supply vessel (PSV) scale model (Fonseca and Gaspar 2020). The digital twin was able to successfully monitor and control the scale model operation, accounting for motion response, navigation, and station keeping on waves. Future research aims to apply the standard to the digital twin of the NTNU's research vessel Gunnerus, as proposed in the following paragraphs.

6.2. Integration of digital twin data

Given the complexity of vessels as systems, it is difficult for single stakeholders, such as shipbuilding companies, to develop an associated digital twin on their own. To distribute the task among various ship stakeholders, digital twins must be created according to standards that enable serialization and exchange among users across companies, allowing yards to integrate such models on an overarching digital twin ship at the same rate at which they integrate the physical subsystems of the real asset. Suppliers are thus able to provide a new feature with the asset, which could, in turn, be commercialised with multiple clients. The integrating party manages an overarching data structure that collects data that are consolidated from the ship design and construction, gradually filling it with component data sent by suppliers and following a top-down approach (Figure 9). The proposed standardization models digital twin elements using repeated objects associated with individual tags for identification of assets, analyses, and data. These elements can be mapped to hierarchies that represent disparate views of data content where systems and SFI establish alternative hierarchies that map data contained in the three other objects.

The asset representation for the digital twin accounts for the ship structure and its physical systems, with data pointing to relevant CAD and visualization files, listing position coordinates and weight data. The digital twin maintainer should aggregate the information necessary to carry simulations that require a holistic view of the asset; and analyses performed during design can also be archived for subsequent use during simulation. For example, results of motion response analyses on relevant loading conditions can be stored in a separate JSON file containing the response amplitude operators (RAOs) for various waves. The file is listed as an analysis under an identification tag and can later be validated with operational data or used to create dashboards simulating an operation to be performed (Miquel et al. 2020). The last component, measured data, is collected from sensor streams and archived for subsequent use. The data are stored according to the ISO 19848 standard, which specifies two types of packages: one describing sensor configuration and other with log readings. In the digital twin, the packages are grouped according to their originating system, and a single sensor system may stream several channels with measurements of different quantities.

The digital twin integration is responsible for organizing individual elements from various suppliers into a meaningful data structure. The digital twin view is primarily system – and operation oriented, so it must be supported by a corresponding hierarchy that organises elements into systems, so the digital twin can close the loop between operational performance and designed functionality. Two taxonomies are suggested on ISO 19848: one by the Japan Ship Machinery and Equipment Association (JSMEA-MAC) and the Vessel Information Systems by DNV GL (DNVGL-VIS). The selected taxonomy is used as a schema to navigate digital twin data by referring to elements' unique identification tags for assets, analyses, and sensors. As observed in the flowchart, it is possible to include multiple hierarchies to arrange the data according to the current task. Besides a system-oriented taxonomy, the SFI group

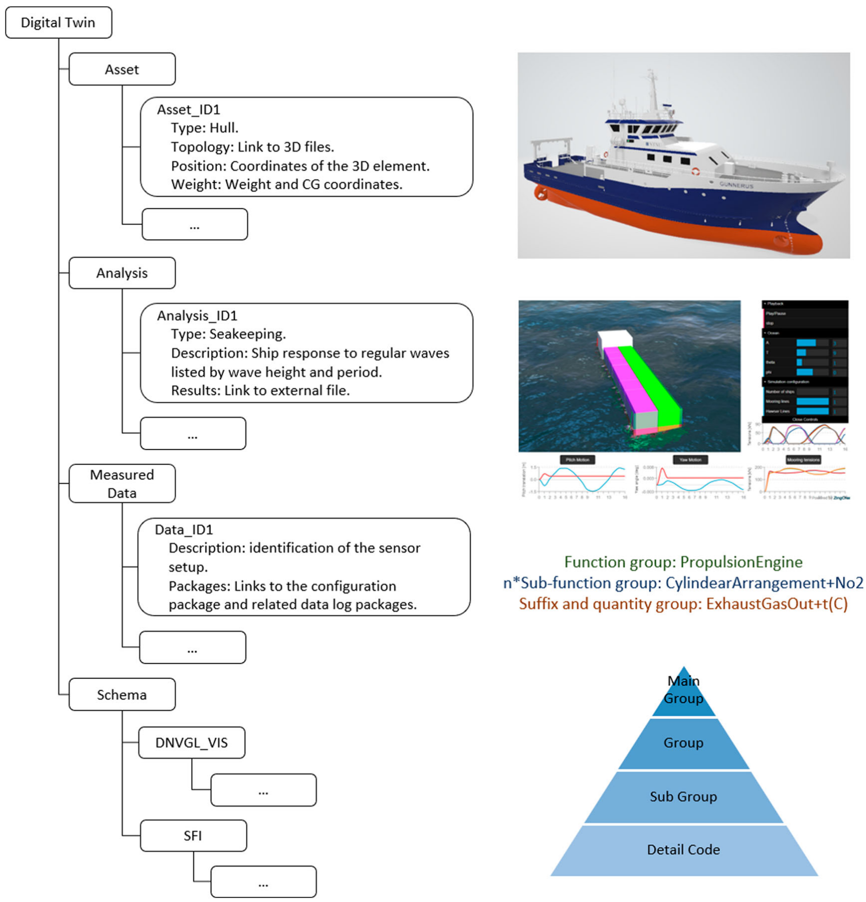


Figure 9. Overall digital twin schema with illustrations of asset visualization, motion simulation dashboard and taxonomies for data organization.

system for shipbuilding can be incorporated in the same manner, and both can be handled independently.

6.3. Digital twin components

In this business scenario, suppliers sell not only the physical subsystems to be installed in the vessel to shipyards, but also an accompanying digital model for integration in the digital twin ship. Data exchanged are serialised into a package with digital twin contents, metadata, and documentation so that use by a receiving party is straightforward. An example of a propulsion system, with dynamic positioning functionality, is detailed in Figure 10.

Asset representation comprises three elements – one bow thruster and two azimuth assemblies with casing and propellers. When linked to operational data, they can be used to visualise the operating propulsion with rotations per minute and azimuth direction. The GL transmission format (glTF) is being assessed as an alternative to serialization and exchange of visualization models of this type. The format allows storage of entire 3D scenes as binary or JSON files, which can be used to transmit geometric models as articulated assemblies that would simplify inclusion on a digital twin visualization.

We previously mentioned approaching standardization of digital twin behavioural models using a bottom-up perspective, which means that if separate component manufacturers adhere to standardised interfaces that allow communication of simulated models, it is possible to include these on the exchange package for aggregation to the overall digital twin ship. The FMI standard provides that type of functionality by allowing the user to create functional mock-up units (FMUs) that can be compiled and exchanged as binary files. Those units can be prepared for co-simulation, encapsulating also the necessary solvers, or as model-exchange units, which rely on external solvers for execution. The supplier would provide additional

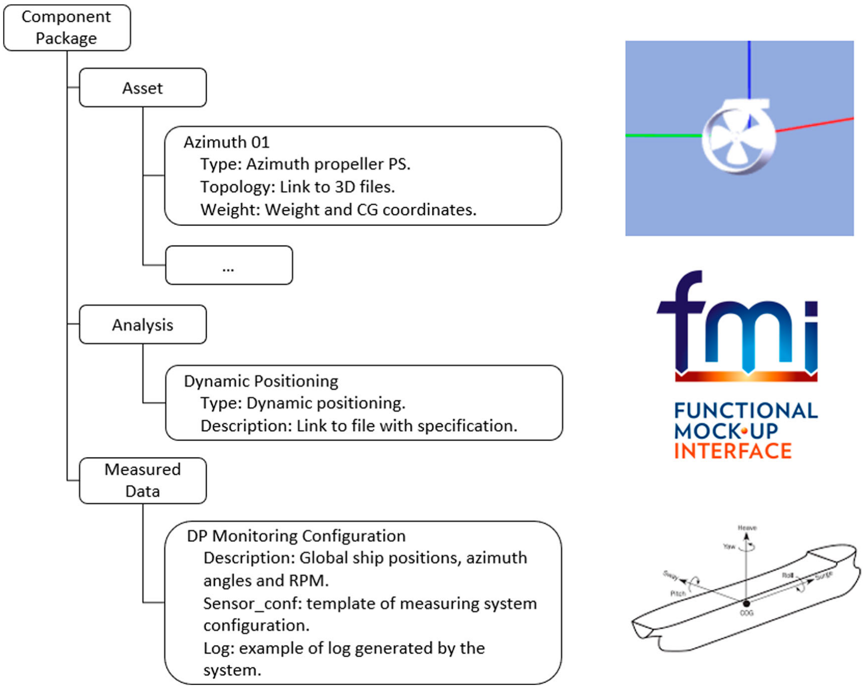


Figure 10. Component schema for serialization and exchange of packages among stakeholders.

documentation describing a recommended instrumentation setup for the physical system and explaining how data gathered from that setup can be linked to visualization and behavioural models. Another approach is to provide the analysis as a self-contained system, with API documentation specifying inputs and outputs. The system is linked to a sensor setup and communication occurs only through the exchange of input commands and output logs that measure the system behaviours of interest. The logs link with the digital twin visualization and are later stored in the digital twin database. That alternative might give more freedom to the supplier, but it requires the effort of developing and maintaining the complete simulation without the basis provided by a neutral standard.

7. Conclusion

The creation of a cohesive digital twin ship faces various challenges, not least the tradition of siloed software systems and data handling in the maritime sector. In the same way that web-based approaches have contributed to compatibility regarding device support and data parsing, the adoption of open standards for engineering models may enable advances in use and exchange of digital twin content. The last years have seen progress in that direction with the appearance of open standards aimed specifically towards those purposes, but further work is necessary. To achieve success, it is necessary that actors in the ship value chain are willing to adopt a collaborative approach to data handling. With this mindset, they can recognize the opportunities of getting involved with standardization initiatives as means to shaping future digital services and research according to their desires and necessities.

Acknowledgements

The authors acknowledge Stein Ove Erikstad and Hans Petter Hildre for valuable comments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Ícaro Aragão Fonseca http://orcid.org/0000-0003-0189-2576
Henrique Murilo Gaspar http://orcid.org/0000-0003-2128-2863

References

Bekker A. 2018a. Exploring the blue skies potential of digital twin technology for a polar supply and research vessel. In: Kujala Pentti, Liangliang Lu, editors. Marine design XIII.

Vol. 1. Helsinki: CRC Press; p. 135–145. doi: 10.1201/9780429440533.
Bekker A, Suominen M, Kujala P, Waal RJOD, Soal KI. 2018b. From data to insight for a polar supply and research vessel. Ship Technol Res. 66(1):57–73. doi: 10.1080/09377255.2018.1464241.
Bole M, Powell G, Rousseau E. 2017. Taking control of the digital twin. In SNAME Maritime Convention (Houston, Texas, USA, Oct. 2017), The Society of Naval Architects and Marine Engineers, p. 19.
Boschert S, Rosen R. 2016. Digital twin—the simulation aspect. In: Hehenberger Peter, Bradley David, editors. Mechatronic futures. Switzerland: Springer; p. 59–74. doi: 10.1007/978-3-319-32156-1_5.
Christian Cabos, 2018. C. R. Digital model or digital twin? In 17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18).
Cameron DB, Waaler A, Komulainen TM. 2018. Oil and gas digital twins after twenty years. How can they be made sustainable, maintainable and useful? In Proceedings of The 59th Conference on Simulation and Modelling (SIMS 59), Sep 26–28 2018, Oslo Metropolitan University, Norway, Linköping University Electronic Press, p. 9–16.
Coraddu A, Oneto L, Baldi F, Cipollini F, Atlar M, Savio S. 2019. Datadriven ship digital twin for estimating the speed loss caused by the marine fouling. Ocean Eng. *186*:106063. doi:10.1016/j.oceaneng.2019.05.045.
Danielsen-Haces A. 2018. Digital twin development-condition monitoring and simulation comparison for the ReVolt autonomous model ship [Master's thesis], NTNU.
Datta SPA. 2017. Emergence of digital twins-is this the march of reason? J Innov Manag. 5(3):14–33.
Defense Acquisition University. 2020. Glossary of defense acquisition acronyms and terms. [Accessed 2020 Apr 30]. https://www.dau.edu/glossary/Pages/Glossary.aspx.
Dufour C, Soghomonian Z, Li W. 2018. Hardware-in-the-loop testing of modern onboard power systems using digital twins. In 2018 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM) (jun 2018), IEEE. doi: 10.1109/speedam.2018.8445302.
Erikstad SO. 2017. Merging physics, big data analytics and simulation for the next-generation digital twins. In 11th Symposium on High-Performance Marine Vehicles.
Erikstad SO. 2018. Design patterns for digital twin solutions in marine systems design and operations. In 17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18), pp. 458–469.
Erikstad SO. 2019. Designing ship digital services. In 18th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'19), pp. 354–363.
Fonseca IA, Gaspar HM. 2020. Fundamentals of digital twins applied to a plastic toy boat and a ship scale model. In *ECMS 2020 Proceedings edited by Mike Steglich, Christian Mueller, Gaby Neumann, Mathias Walther*, ECMS. doi: 10.7148/2020-0207.
Gaspar HM. 2018. Data-driven ship design. In 17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18) (2018), pp. 426–439.
Gaspar HM. 2019. A perspective on the past, present and future of computer-aided ship design. In 18th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'19) (2019), pp. 485–499.
Gielingh W. 2008. An assessment of the current state of product data technologies. Comput Aided Des. 40(7):750–759. doi: 10.1016/j.cad.2008.06.003.
Grieves M, Vickers J. 2016. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: Kahlen Franz-Josef, Flumerfelt Shannon, Alves Anabela, editors. Transdisciplinary perspectives on complex systems. Switzerland: Springer International Publishing; p. 85–113. doi: 10.1007/978-3-319-38756-7_4.
Hatledal LI, Zhang H, Collonval F. 2020. Enabling python driven co-simulation models with PythonFMU. In ECMS 2020 Proceedings edited by Mike Steglich, Christian Mueller, Gaby Neumann, Mathias Walther (jun 2020), ECMS. doi: 10.7148/2020-0235.
ISO. 2004. 10303-218 industrial automation systems and integration - product data representation and exchange - part 218: Application protocol: Ship structures.
ISO. 2018a. 19847 ships and marine technology — shipboard data servers to share field data at sea.
ISO. 2018b. 19848 ships and marine technology — standard data for shipboard machinery and equipment.
Låg S, With SB. 2017. Standardisation as an enabler of digitalisation in the maritime industry. DNV GL.
Levisauskaite G. 2016. Implementation of 4gd framework in ship design for improving exchange and 3d reuse [Master's thesis], NTNU.
Manchinu A, McConnell F. 1977. The SFI coding and classification system for ship information. In Proceedings of the REAPS Technical Symposium.
Miquel SE, Fonseca IA, Gaspar HM, Vieira DP. 2020. An open source library for hydrodynamic simulation of marine structures. Mar Syst Ocean Technol. doi:10.1007/s40868-020-00083-3.
Morais D, Goulanian G. 2019. The future reality of the digital twin as a cross-enterprise marine asset. In RINA, Royal Institution of Naval Architects – International Conference on Computer Applications in Shipbuilding, ICCAS 2019 (2019), vol. 1, pp. 67–72.
Morais D, Waldie M. 2018 How to implement tech in shipbuilding: Charting the course to success. In SNAME Maritime Convention (Providence, Rhode Island, USA, Oct. 2018), The Society of Naval Architects and Marine Engineers, p. 5.
Nokkala T, Salmela H, Toivonen J. 2019 Data governance in digital platforms. In 25th Americas Conference on Information Systems, AMCIS 2019, Cancún, Mexico, August 15–17, 2019.
Otto B, Auer S, Cirullies J, Jürjens J, Menz N, Schon J, Wenzel S. 2016. Industrial data space: digital sovereignity over data. *Fraunhofer White Paper*.
Rachuri S, Subrahmanian E, Bouras A, Fenves SJ, Foufou S, Sriram RD. 2008. Information sharing and exchange in the context of product lifecycle management: role of standards. Comput Aided Des. 40(7):789–800. doi: 10.1016/j.cad.2007.06.012.
Rødseth ØJ, Berre AJ. 2018. From digital twin to maritime data space: Transparent ownership and use of ship information. In 13th International Symposium on Integrated Ship's Information Systems & Marine Traffic Engineering Conference.
Schirmann M, Collette M, Gose J. 2019. Ship motion and fatigue damage estimation via a digital twin. In 6th International Symposium on Life-Cycle Civil Engineering (IALCCE), Ghent, BELGIUM, Oct 28–31, 2018, Caspeele, R and Taerwe, L and Frangopol, DM, Ed., pp. 2075–2082.
Schroeder G, Steinmetz C, Pereira CE, Muller I, Garcia N, Espindola D, Rodrigues R. 2016. Visualising the digital twin using web services and augmented reality. In 2016 IEEE 14th International Conference on Industrial

Informatics (INDIN) (July 2016), pp. 522–527. doi: 10.1109/INDIN.2016.7819217.

Shafto M, Conroy M, Doyle R, Glaessgen E, Kemp C, LeMoigne J, Wang L. 2010. Draft modeling, simulation, information technology & processing roadmap. Technol Area. 11:18–19.

Sharma P, Hamedifar H, Brown A, Green R. 2017. The dawn of the new age of the industrial internet and how it can radically transform the offshore oil and gas industry. In Offshore Technology Conference, Offshore Technology Conference. doi: 10.4043/27638-ms.

Siemens PLM Software. 2013. *Providing the next-generation design paradigm for shipbuilders*.

Skjong S, Rindarøy M, Kyllingstad LT, Æsøy V, Pedersen E. 2017. Virtual prototyping of maritime systems and operations: applications of distributed co-simulations. J Mar Sci Technol. 1–19. doi: 10.1007/s00773-017-0514-2.

Stachowski T-H, Kjeilen H. 2019. Holistic ship design - how to utilise a digital twin in concept design through basic and detailed design. In RINA, Royal Institution of Naval Architects - International Conference on Computer Applications in Shipbuilding, ICCAS 2017 (Singapore, Singapore, 2019), vol. 2, pp. 101–110.

Tofte BL, Vennemann O, Mitchell F, Millington N, McGuire L. 2019. How digital technology and standardisation can improve offshore operations. In *Offshore Technology Conference*, Offshore Technology Conference. doi:10.4043/29225-ms.

van den Hamer P, Lepoeter K. 1996. Managing design data: the five dimensions of CAD frameworks, configuration management, and product data management. Proc IEEE. *84*(1):42–56.

Van Os J. 2018. The digital twin throughout the lifecycle. In SNAME Maritime Convention (Providence, Rhode Island, USA, Nov. 2018), The Society of Naval Architects and Marine Engineers, p. 8.

West TD, Blackburn M. 2017. Is digital thread/digital twin affordable? a systemic assessment of the cost of dod's latest manhattan project. Procedia Comput Sci. *114*:47–56.

West TD, Blackburn M. 2018. Demonstrated benefits of a nascent digital twin. INSIGHT. *21*(1):43–47. doi: 10.1002/inst.12189.

Whitfield RI, Duffy AHB, Meehan J. 2003. Ship product modeling. J Sh Prod. 19(4):230–245.

Whitfield R, Duffy A, York P, Vassalos D, Kaklis P. 2011. Managing the exchange of engineering product data to support through life ship design. Comput Aided Des. 43 (5):516–532. doi:10.1016/j.cad.2010.12.002.

Xantic. 2001. SFI Group System – A system for classification of technical and economic ship information. Product Description.

Paper II

Ícaro Aragão Fonseca et al. (Jan. 2022). 'A Standards-Based Digital Twin of an Experiment with a Scale Model Ship'. In: *Computer-Aided Design* 145. DOI: 10.1016/j.cad.2021.103191

Computer-Aided Design 145 (2022) 103191

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A Standards-Based Digital Twin of an Experiment with a Scale Model Ship

Ícaro Aragão Fonseca [a,*], Henrique Murilo Gaspar [a], Pedro Cardozo de Mello [b], Humberto Akira Uehara Sasaki [b]

[a] *Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Ålesund, Norway*
[b] *Numerical Offshore Tank, University of São Paulo, São Paulo, Brazil*

A R T I C L E I N F O

Article history:
Received 9 February 2021
Received in revised form 18 December 2021
Accepted 23 December 2021

Keywords:
Digital twin
Simulation
Data standard
Internet of things
Ship
Remote operations

A B S T R A C T

We propose the use of existing data standards and web technologies to modeling and development of digital twin ships. Our research provides an open framework that can be linked to services such as visualizations, simulations and remote control. The case study applies the standards-based framework to an experiment that involved a scale model ship equipped with a dynamic positioning system under artificial waves. The digital twin prototype illustrated the capability of mirroring and controlling the model's position in real-time, and predicting motion responses across wave conditions via a web application. Thus, it closes the loop between test and design in the life cycle by allowing validation of results in comparison to empirical data during operation. The results from these experiments are used to discuss an expanded version of the digital twin for validation and optimization of motion response, as well as its implications to the system's (ship) taxonomy and data management. The conclusion summarizes lessons when using the adopted standards, as well as challenges when scaling the approach to real life operations. Future research is proposed toward extending the standardization to more complex cases.

1. Digital twins in maritime applications

Digital twins were popularized in aerospace industries, based on the principle of using digital models, sensor information, and input data to mirror and predict behaviors of a corresponding physical asset [1]. This seems to be done out of need rather than choice, given the impossibilities of having a physical, wired connection with any asset in the outer space. The maritime industry shares this characteristic with the aerospace industry, as a ship can be observed as an asset that operates in the interface of two fluids, without a physical connection to the shore. Engineering analyses and simulations have been used for decades in the maritime sector to predict the operational behaviors of a system during design, with results providing insights into how to select, modify, and refine design alternatives according to their expected functional performance. However, once a ship is launched, it is tested according to a standard row of criteria during a sea trial and then handed over to the ship owner, who will, in most cases, operate it independently from the design office. This causes segmentation, in which simulations prepared during design are not used to aid operation, and behaviors that occur during operation are not used to guide design of new vessels. A digital twin might overcome this trend by centralizing data management; during design and manufacturing, it collects the product model and simulations of behaviors, and once the system goes to operation, it is animated with data streamed from sensors and connected services. Thus, the digital twin is an essential enabler to the "digital thread", a term meaning the use of software, data, and governance models to obtain a comprehensive digital view of an engineering system throughout its life cycle [1].

Given the complexity of ships as systems and the relative novelty of the digital twin concept, digital twin implementations in both maritime industry and research are still emerging [2]. They commonly occupy a narrow domain, or they have not yet attained a high degree of maturity. Challenges when implementing a comprehensive digital twin ship span several levels. At the most fundamental, there is need to identify business and governance models that take advantage of the opportunities created by digital technologies, whether based on servitization or otherwise. Regarding infrastructure, there is need to build streaming services with reliable databases, with transmission hardware and protocols capable of exchanging large amounts of data (bandwidth) between sea and land reliably. When it comes to system integration, the maritime industry has a history of fragmentation and incompatibility of digital tools during ship design and operation.

* Corresponding author.
E-mail address: icaro.a.fonseca@ntnu.no (Í.A. Fonseca).

https://doi.org/10.1016/j.cad.2021.103191

Different hardware and software systems often adopt incompatible proprietary formats for computer aided design models, sensor measurements, engineering analyses, and simulations [3]. The tender-based approach to data management in the maritime sector opens up for several customizations tending to undermine digital interoperability, e.g., conventions imposed by different classification societies and shipyard building strategies [4]. This represents a central concern because a digital twin uses data and models generated by various stakeholders during multiple stages of the life cycle, requiring a systematic approach to development and data integration.

Development and use of data standards are central to enabling this objective, since they should allow compatibility and exchange of models developed by disparate parties who use various software [5]. We recognize that even development and adoption of technical standards are influenced by variables that are measured not only from an objective perspective of system performance, but subjective positions of the involved parties. A standard that is technically functional and apparently reasonable might be rejected if one or more prospective user perceives it as too prescriptive, cumbersome, or for any reason not advantageous. This work serves two purposes in that context. The first one is assessing use of existing standards to digital twin development, with the objective of identifying their respective advantages and limitations. This aspect of the research to some extent builds upon previous work by the authors about web-based development of maritime simulations, as will become clear throughout the paper. The second purpose is to outline methods which might inform future standardization initiatives in the maritime sector, starting from a set of design choices aimed at dealing with the stated problems.

2. Standards for digital twin data

2.1. Modeling approach

By our design choice, when selecting standards for use in this work, we prioritized standards that are open (distributed freely) and neutral (not favoring particular vendors or actors in the industry). To avoid lock-in, we choose formats which enjoy broad support across software tools rather than being tied to a particular suite. With data becoming increasingly important through the maritime value chain, it becomes necessary that it is intelligible to employees of various backgrounds besides software development. Thus, it is necessary to select formats of simple interpretation by users to allow independent data exchange and reuse. To accomplish such goals, we look to the stack of web-based technologies. The reasons for this choice are discussed in a previous publication [6], but in summary, a web-based approach is compatible across devices and operating systems, it offers access to a broad pool of reusable, open-source code for application development, and it enables nearly instantaneous access to geographically distributed users.

Table 1 summarizes the selection with a typology of standards for product information-sharing and exchange, as defined by Rachuri et al. [7]. The typology follows a hierarchy, starting with the most basic or fundamental standards and moving toward the most comprehensive and sophisticated. Type zero standards are for implementation (i.e., programming) languages. Type one standards represent information modeling standards that model information but do not impose a specific domain of discourse or schema on the data. Type two standards are content standards that model domains of discourse, subdivided into five sub-types. The three first are for product information modeling and exchange, information exchange, and product visualization. The remaining two – standards for e-business and value chain support, and standards for security – are not considered in this work. The last type of standard concerns architectural frameworks that reconcile data use during the product life cycle from various viewpoints (e.g., enterprise, technology, and engineering). We do not consider these standards because they would require implementing a digital twin with a holistic perspective of an organization, thus lying beyond the scope and stage of maturity of this research. However, they might enable integration among future digital services in maritime and to codify a data management strategy from a business perspective. See, for example, the ISO 19847 standard for shipboard data servers to share field data at sea [8].

Regarding the first two standard types, implementation languages are those used for web development, i.e., HTML and JavaScript. For the type one standard, we use JavaScript Object Notation (JSON), which offers readability to users unfamiliar with software development, thus bridging the gap between development and engineering disciplines. JSON originated from JavaScript but is now widely supported in the information technology industry, including tools for conversion to other information modeling standards such as XML.

2.2. Digital twin content

Digital twin content is here organized into: representation of the physical asset, states collected from the asset, operating context, and simulations or analyses of behavior (based on [9]). A digital twin contains asset models that are shared among all simulations. The sensor observations collected from a system and its context feed services for analysis and simulation of across ship behavior domains. Although simulations and data used to aid operation of engineering assets are not new, the digital twin applies recent developments in technologies for simulations, digital services, and industrial internet of things to centralize all relevant digital models of the physical system it represents, thus operating as a hub that a user can access to perform key activities related to the assets' data. West and Blackburn comment on the feasibility of developing a digital twin of a fighter aircraft, drawing attention to the great burden of developing high-fidelity simulation models across several domains as a single platform [10]. They conclude that piece-wise progression is advisable, in which disparate modules that cover various simulation aspects of the asset are linked to the digital twin. Every module would be developed with relative independence to achieve the necessary balance between fidelity and simplification. Erikstad discusses balancing capability with the cost of implementing digital services for ship operations [11], proposing a service maturity index to grasp the progression from simpler, but not necessarily ineffective, to advanced services using five levels—observe, measure, model, predict, and decide. Fig. 1 summarizes this framework with examples of possible digital twin services in different domains. For succinctness, we condense the maturity scale to three levels only: monitor, predict, and decide.

Asset representation. Representation is based primarily on visualization files and textual metadata, such as specifications. Given the intention of using the digital twin for real-time monitoring of ship operation, we turned to polygonal formats which are lighter than NURBS-based alternatives such as IGES or STEP. This comes at some expense for geometry accuracy, but we assume the original NURBS files will remain available in case such accuracy is eventually needed. Suitable format alternatives are JT (Jupiter Tessellation), Wavefront OBJ, STL and glTF (Graphics Library Transmission Format). We adopted glTF due to it enjoying broad support and free licensing. It can be saved as text or binary and supports entire scenes with assemblies, colors, materials, and lights [12]. The 3D models should be complemented by metadata containing coordinate positions, weight, authorship, remarks, and identification tags.

Table 1
Typology of standards adapted to digital twins (based on [7]).

Type	Scope	Adopted
0	Implementation languages	JavaScript, HTML, CSS
1	Information modeling	JSON
2	Content standards (domains of discourse)	glTF, ISO 19848, DNV VIS
3	Architectural frameworks	None[a]

[a]Integration at enterprise-level not considered in this research.

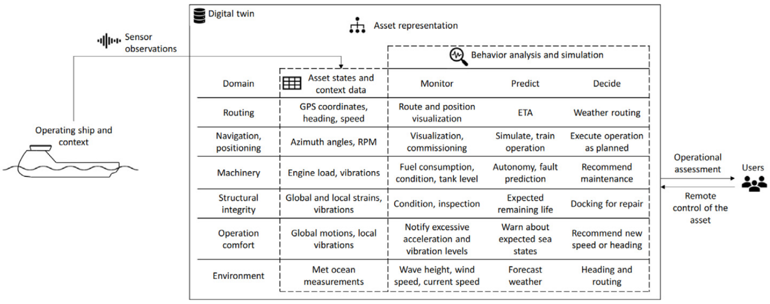


Fig. 1. Digital twin content groups with examples of simulations at different maturity levels.
Source: Updated from [2].

Ship states. A digital twin needs to gather observations from a vessel's instrumentation device in order to model the current operational situation. Measurements might be readily usable once obtained, such as in GPS coordinates showing the current vessel position, or might require further processing in order to derive a comprehensive state, such as in an exhaust temperature that gives partial insight into the functioning of an engine. This distinction depends on the digital twin's purpose and observed variable. Relevant standards within this scope are the IEC 61162 series for digital interfaces between navigation equipment [13] and the ISO 19848 standard data for shipboard machinery and equipment [14], the latter of which we adopted in this work, as schematized in Fig. 2. Ando discusses background development of the ISO 19847 and 19848 standards in Ref. [15][1]:

- Data channel list: a data channel transmits one measured variable. Depending on channel type, the variable will be classified as an instantaneous value or observation, a calculated value derived from observations, or a manual input by the crew, among others. A data channel list contains metadata describing type, purpose, update cycle, units of measure, expected range, and identification tags of one or more data channels.
- Time series data: they contain observation logs stored as one of two types: tabular or event data. Tabular data are suitable for raw numeric values or statuses sampled from sensors at regular rates, and event data model alarm information, status information, and input data at intervals not specified previously.

A user might also model data as XML, as it is fully compatible with JSON or, if a more compressed time series format is needed, CSV. The organization of shipboard data in packages provides some independence and modularity, allowing data serialization into self-contained files for exchange among parties or systems.

[1] Fig. 2: Diagram representation of ISO 19848 data structures is reproduced by Ícaro Fonseca in A Standards-Based Digital Twin of an Experiment with a Scale Model Ship under license from Standard Online AS November 2021. © All rights are reserved. Standard Online makes no guarantees or warranties as to the correctness of the reproduction. See www.standard.no.

Other than that modular structure, the standard does not prescribe any schema for organizing the packages in an overarching structure. This is not necessarily a limitation, but probably an intentional gap intended at increasing the standard's versatility for use with different database arrangements. For instance, Annexes B and C illustrate how the shipboard data can be identified according to two different naming rules provided by different vendors: the JSMEA-MAC by the Japan Ship Machinery and Equipment Association and the Vessel Information System (VIS) by DNV [16]. Because ISO sells PDF copies of the standard to users, it is reasonable not to consider it "open" [17]. If much, it might be considered "neutral" for not prioritizing the necessities of any one company in particular. Still, given the lack of competitors sharing its scope, we see the standard as one of the few viable alternatives for adoption.

Operating context. This includes representation of the surrounding environment (e.g., topographic maps) and its states (e.g., wave state, temperature, currents, and winds). The obstacles to effectively manage environmental data in digital twins are slightly different from those of ship data. From one side, metocean data has been more standardized than ship data with formats such as the Network Common Data Form for various meteorological conditions [18]. For instance, in 2013 the Norwegian Mapping Authority released free topographic data sets in the Digital Elevation Model standard by the U.S. Geological Survey, a legacy and still used format. These files provide a useful resource for automating the creation of 3D environments inside which vessel operations can be simulated [19]. On the other hand, most ship owners do not own or control weather monitoring infrastructure, so even if quality data from a certain region is available, access to it may not be straightforward. ISO 19901-1 on metocean design and operating considerations for offshore structures provides an overview of data collection infrastructure for different seas in Annexes A to G, showing that it is owned by several public and private entities around the world [20]. The challenge then becomes first to acquire access to data in the operating region of the vessel, often through external suppliers, and then link it to the digital twin. The existence of projects to document and harmonize existing metocean databases might make this process smoother [21].

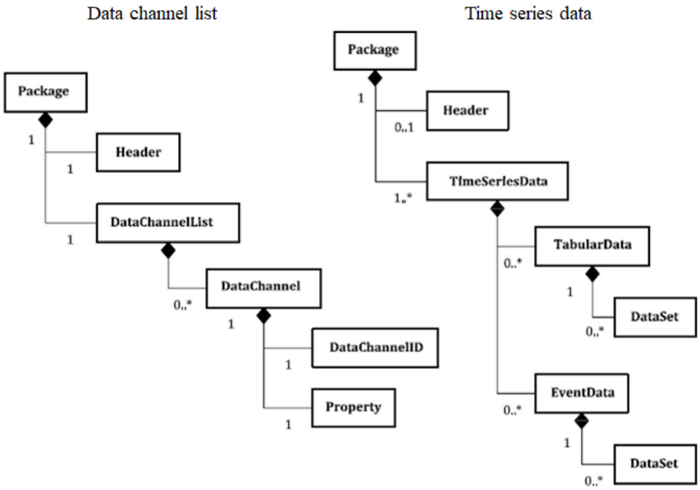


Fig. 2. Diagram representation of ISO 19848 data structures [14].

Simulations of behavior and other analyses. Recent work introduce a few considerations regarding this aspect of digital twin ships. For instance, the Functional Mock-up Interface (FMI) is a relevant alternative for simulation of dynamic systems, allowing exchange and co-simulation of cyber–physical models contained in modular units. It has been applied across different engineering domains, including maritime [22]. The authors have previously presented web-based simulations for ship operations, where geometries and weights are used to calculate hydrostatic, stability, resistance, motion response and fuel consumption, among others [23]. The simulation code does not have the formal rigor expected from a technical standard, but it outlines an approach we aim to continue exploring in the future. Finally, ship states give a data pool for training and testing of machine learning models, e.g., those targeted at predictive maintenance. Standardized schemas and metadata should simplify model reuse by allowing automation of data access and parsing during calibration.

2.3. Establishing a coherent taxonomy of a digital twin ship

Appropriate choices of views and hierarchies play a central role during management of engineering data, including a digital twin; one must first choose an adequate view to describe the system digitally and then model a hierarchical schema that organizes data according to those views, i.e., establish a taxonomy of the system. During early design stage, cargo vessels are traditionally described in lightweight groups for structure, machinery, outfitting and accommodation. Alternatively, methods for design of specialized vessels such as the Design Building Block Approach and System Based Ship Design prefer to emphasize architectural concerns by describing the vessel in terms of spaces with corresponding properties such as volumes and areas [24,25]. Such spaces are arranged in a hierarchy of systems or functions so they can be roughly traced to the tasks a ship is required to perform. During the construction phase, yards commonly adopt a physical view of the ship decomposed into blocks, assemblies, panels, and so on.

A digital twin should provide a meaningful structure to support data management during asset operation. Given this scope, it should be based in a functional taxonomy that can be studied and evaluated in comparison to desired or expected performance. This makes organization based on a systems-oriented view of the vessel desirable. In addition, ship states can be collected from very specific vessel components (e.g., the shaft of a pump that controls flow toward a tank), and thus the taxonomy must provide an adequate level of detail. Unfortunately, we did not find many alternatives which fulfilled these requirements. For instance, the Expanded Ship Work Breakdown Structure by the U.S. Navy is targeted at military vessels [26]. We identified two alternatives, incidentally both from the Norwegian context: the SFI Group System by SpecTec [27] and VIS [16]. We adopted VIS as the primary taxonomy for ship data for two main reasons: it is maintained with open documentation on the web (while SFI is distributed under payment) and it is being positioned to also manage sensor logs (in contrast to purely static product data). VIS is a code scheme that uses a functional view of a vessel, prioritizing a description of functions that can be assigned to the correct component. The coding scheme is aimed at clustering ship data in corresponding systems, and the ISO 19848 Annex C specifies how it can be used to identify data logs collected during operation. Thus, it can be used to map asset representation and sensor observations in a digital twin.

This choice still leaves open the task of finding suitable taxonomies for remaining digital twin content, namely operating context, and models of behavior. ISO 19901-1 puts forth a list of relevant metocean parameters to be included in databases. The parameters are organized in six main categories, which we adopt as a preliminary schema [20]: waves, currents, winds, ice, water levels and others. We see two alternative paradigms to organize simulations of behavior in a digital twin. The first one is illustrated by modern app stores, which place software in categories defined by the store owner or maintainer. The second is the tag system used in package managers and source hosting services, where publishers are allowed to define one or more keywords for each project. We judge the latter is more suitable to an open digital twin platform, as it avoids the prescription of categories and provides flexibility for a service to be described according to more than one taxonomy, such as behavior domain, simulation method and purpose in the ship value-chain.

3. Digital twin development

3.1. Standardization framework

Given the prospect of digital twins increasing in scope and importance in the maritime value chain, we put forth a modeling approach supporting gradual addition of content to the digital twin as new functionalities are implemented. We aim to achieve flexibility for adaptation to different data taxonomies while making use of the discussed standards to allow data exchange and interoperability among software tools. In terms of content, the standards establish a framework for asset representation and states, with methods for simulation of behavior to be approached in a future work. Digital twin data is arranged into individual packages, each storing textual descriptions about their purposes and characteristics (i.e., JSON metadata) and links to binary files if needed. Packages are stored in a flat hierarchy, but they contain identification tags that map them to the desired taxonomy for organizing and browsing digital twin content. We expect such modularization to maintain independence among resources, facilitating data exchange among digital twin stakeholders. We apply a standardization approach based on the principles outlined above to a simplified case study intended as starting point to future applications of greater complexity. We developed a digital twin of an experiment with a scale model platform supply vessel (PSV) in a wave basin. The digital twin data is linked to an algorithm developed on a proprietary platform to monitor and control the scale model PSV while accounting for motion response, navigation, and station keeping on waves. Thus, it offers a higher degree of service maturity compared to a simple online monitoring tool. The following steps describe the framework applied to the case study:

1. Identify the existing physical setup and installed sensors (Section 3.2).
2. Define the digital twin's intended purpose 3.3.
3. Map the data required to develop the digital twin 3.4.
4. Prepare the digital asset representation 3.5:
   (a) Write overall ship specification (JSON).
   (b) Convert necessary 3D models to standard format (glTF) and prepare corresponding metadata.
5. Prepare templates (ISO 19848) for storage of PSV states during the experiments 3.6:
   (a) Write data channel list.
   (b) Write time series data templates.
6. Model the operational environment, i.e., wave basin 3.7:
   (a) Create a digital model accounting for relevant features of the basin.
   (b) Prepare templates to store a log of wave conditions encountered by the PSV during operation.
7. Develop models for simulation of behavior 3.8:
   (a) Convert existing results from motion response analyses to self-standing JSON files.
   (b) Develop simulation models and graphical user interfaces (as web applications).
8. List the digital twin content according to the chosen taxonomies 3.9.
9. Aggregate individual components into a functional web-based system 3.10.

3.2. Experimental facility and instrumentation

The experiment was conducted using a PSV scale model in the Numerical Offshore Tank laboratory at the University of São Paulo (TPN-USP). In terms of experimental equipment and data-gathering infrastructure, the digital twin followed a bottom-up approach by linking to systems used already in the laboratory workflow. The tank itself measures 14 meters on each side and is 4.1 meters deep (Fig. 3). It is equipped with flaps that are capable of generating regular and irregular wave states from virtually any direction of propagation. The flaps also operate as wave absorbers that minimize wave reflection to avoid interference with desired wave characteristics [28]. A commercial solution for test and measurement reads the water elevation using several probes installed in the tank. To simplify identification of the wave state that occurred in the basin with the digital twin, the experiment was conducted with only regular waves that approached from a single, previously known direction, allowing extraction of wave amplitude and period using a single wave probe.

The hull used during the experiment was a 1:70 scale model of a PSV that measured 1.24 meters in length, 0.345 of beam, and 0.082 of draft (Fig. 4). It was actuated with a dynamic positioning (DP) system for navigation and station keeping that comprised two azimuth systems on the stern and a tunnel thruster on the bow. A commercial solution tracked motion of the PSV in 6 DOF (degrees of freedom) using a stereoscopic camera setup. The DP control module developed at TPN connects with that solution to operate the propulsion remotely by transmitting commands to the PSV model over radio (further details in Appendix A.1).

3.3. Digital twin purpose

The digital twin platform was developed to offer monitoring and control capabilities during experiments, with the following objectives:

- Display the operational situation in real-time:
  - The vessel's DP parameters (position in 6 DOF, propeller rotation rates and azimuth angles).
  - Physical characteristics of surrounding waves.
- Allow capabilities to access the data and control the physical asset (ship model) remotely via a web-based app.
- Provide complementary functions for assessment of motion response during operation:
  - Validation of stored response operators based on actual measurements.
  - Automatic maneuvering to minimize motion response according to stored operators.

3.4. Data content and schema

The schema clusters the digital twin's content into four groups, based on the categories discussed in Section 2.2—asset representation, sensor observations, environment observations, and behavior models. A fifth group in the data schema then stores the taxonomies that were used to map the digital twin content by referring to the identification tags stored in the individual data packages. Fig. 5 shows an overview of the digital twin schema.

3.5. Digital ship

Ship specification. The PSV specification was stored as a JSON file that contained primary dimensions and weight distribution (Fig. 6, left). Besides serving as a reference about the PSV model, the specification was used to extract the draft and overall length when rendering the 3D ship visualization.

Components and 3D models. The other packages in the asset representation described the ship's components, providing coordinate positions, VIS identification tags, and links to files with the respective 3D models (Fig. 6, right). Since the objective of the study was to monitor global motion response, the hull was modeled as a single part, without further divisions (e.g., sides and bottom). As intended, glTF allowed modeling of system assemblies with articulations for movements, illustrated by the local coordinate axes for propeller rotation in relation to the nozzle.

Fig. 3. Wave tank facilities at TPN-USP [29].

Fig. 4. Scale model PSV used during the experiments [30]. Six reflective, spherical markers for motion tracking were attached to the model.

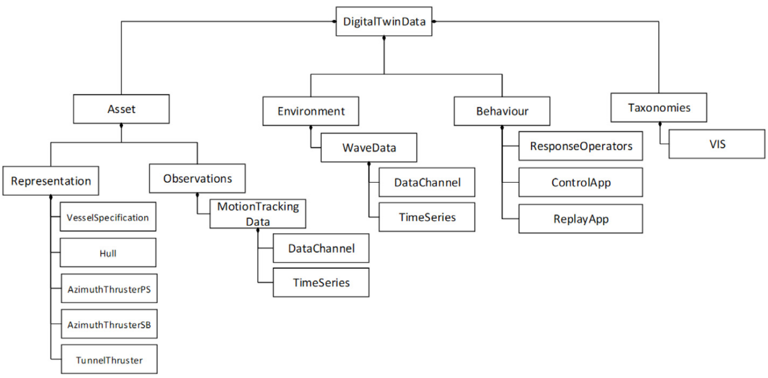


Fig. 5. Case study content schema.

3.6. PSV operational states

All ship states were transmitted in a single data channel list when communicating with the DP system, following the format defined by ISO 8601 as presented in 2. A data channel file was prepared with the metadata to describe the meaning of each value in the logs (Fig. 7, left). The digital twin received a stream of operational states at 100 Hz, modeled as tabular data. Commands that defined the DP setpoint were modeled as events triggered by the user. Results were stored in a time-series file for posterior use, e.g., during an experiment rewind. The file listed the time and corresponding measurements (Fig. 7, right).

3.7. Operating environment

Wave basin model. The digital twin must identify incident waves that the vessel encounters during operation to evaluate its motion response, so a digital model was created to display the wave basin

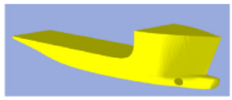
Asset Representation

Specification


```
Remarks: "A ship
specification draft.",
Author: "Employee_1",
ShipID: "IM01234567",
MainDimensions: {
  LOA: 1.24,
  B: 0.3450,
  DesignDraft: 0.0820,
   UnitSymbol: "m"
},
Weight: {…},
CG: {…},
AssetID: {
  LocalID: "/dnv_vis/071",
  …
}
```


Components


```
Remarks: "High level description of the hull.",
Author: "Employee_1",
Topology: {
  Visualization: {
    STL: "hull.stl",
    GLB: "hull.glb"
  },
  Position: {…},
  UnitSymbol: "mm"
 },
AssetID: {
  LocalID: "/dnv_vis/111(mm)",
  …
}
```

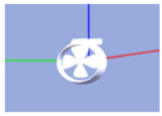


Fig. 6. Excerpt of the ship representation metadata with corresponding 3D models.

Dynamic Positioning States

Data channel list


```
DataChannelID: {
  LocalID: "/dnv_vis/453+I101/
  ShipHeave+d(mm)",
  ShortID: "PosZ",
  …
  }
},
Property: {
  DataChannelType: {
    Type: "Inst",
    UpdateCycle: "0.01"
  },
  Format: {
    Type: "Decimal"
  },
  Unit: {
    UnitSymbol: "mm",
    QuantityName: "Heave Displacement"
  },
  Name: "Ship Heave Displacement"
  }
}
…
```


Tabular data


```
DataChannelID: ["PosX","PosY","PosZ",…],
DataSet: [{
    TimeStamp: "2020-01-22T17:47:17.52Z",
    Value: ["-194.91974","2.24150",…]
  },{
    TimeStamp: "2020-01-22T17:47:17.53Z",
    Value: ["-195.64748","1.62077",…]
  },{
    TimeStamp: "2020-01-22T17:47:17.54Z",
    Value: ["-194.95781","2.21758",…]
  },{
    TimeStamp: "2020-01-22T17:47:17.55Z",
    Value: ["-198.62386","1.71179",…]
  },{
    TimeStamp: "2020-01-22T17:47:17.56Z",
    Value: ["-198.67555","1.70286",…]
  },
  …
]
```


Fig. 7. Excerpt from DP state data.

area and experiment waves in proportion to the PSV. The visualization is based on an open source example provided with the Three.js library, so it can be reused during development of web applications. It displays a water surface which can be configured to render a regular wave according to its height, period and phase. The wave length is automatically derived from the dispersion relation for deep waters and the direction is fixed during the experiments.

Environment states. Despite the scope of ISO 19848 being data for shipboard machinery and equipment, we chose to also use it to model wave characteristics. This was possible because the experiment environment is simple and controlled. Should the approach be scaled to real life operations, it would be necessary to account for the geographic perspective, so that a metocean condition can be assigned to an operating region. The instrumentation system installed in the wave basin measured water elevation at a 100 Hz frequency. Since we simplified the experiment to assess regular waves, it was sufficient to store only the wave characteristics as post-processed from the water elevation log, allowing reduction of the data footprint. Each time the algorithm that processed water elevations identified new characteristics of regular waves, it triggered a new event and saved the values to the data log (see Appendix A.2).

3.8. Behavioral models

Motion response operators. The motion response operators for amplitude and phase were obtained using a commercial boundary element method software [31] and converted to a JSON file. The file organized results using one of six motion modes, then 60 periods and 25 wave headings, starting from 0° and with an interval of 15°, up to 360°. Periods were defined at a real-life scale, so they had to be converted when applied to responses of the PSV model. The definition of JSON schema for storage of the results required a choice among alternatives which, though syntactically different, are equivalent in meaning. This observation applies to a large variety of results that can be stored in the digital twin. To minimize the effect of syntactic differences, a descriptive header was included in the package describing the results' meaning, units, and conventions. Although this is not

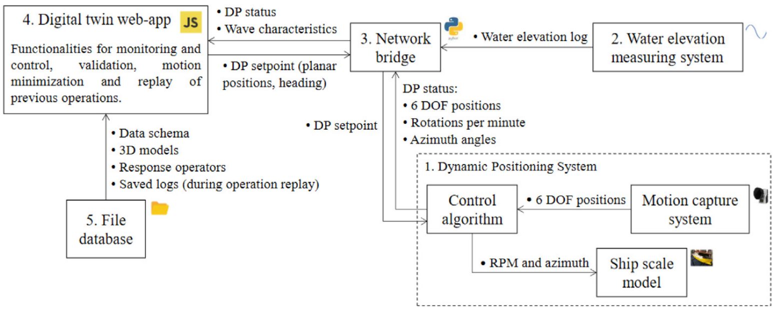


Fig. 8. Communication diagram of the digital twin system during experiment streaming.

as effective as having a common schema across the industry, it allows clear interpretation by those who use it.

Monitoring and control. Two web applications were created for the digital twin, one is a dashboard for monitoring and controlling an experiment while it occurs and another to rewind a previous experiment based on stored data. The monitoring dashboard also includes functionalities for validation of scale model motion and optimization for minimal motion response (see A.3 for details about the algorithms). During the experiment, the dashboard connects to the wave measurement and DP control systems over a local network. The DP control algorithm is handled as an external, self-contained module, and communication with it occurred only through exchanges of inputs and outputs. The DP module received as input from the user the desired setpoint on the navigation plane (i.e., coordinates and heading), and the control system automatically maneuvered the model to attain the desired position. In return, the module outputs logs containing vessel positions in 6 DOF and propulsion states. This arrangement was planned to allow for remote vessel control with the web-based application, an interaction discussed in more detail in Section 3.10.

3.9. Data taxonomies

Once all digital twin content had been collected and modeled using the appropriate metadata, they were mapped using taxonomies to allow a structured overview of the data by referring to an element's unique identification tag. We used VIS as the primary data taxonomy for ship representation and states. The schema were firstly created manually by reading the JSON files and searching for necessary information, but in the future, an algorithm could be created to assemble the hierarchies automatically from the identification tags contained in each data package. Since the digital twin contained only a few packages, the content did not span many groups modeled in VIS. In addition, VIS documentation provides flexibility when deciding the level in the hierarchy to which a sensor should be allocated. The hull model was mapped to a high level because it is not decomposable into component parts, but the thrusters were assigned to lower levels in the propulsion hierarchy. State logs were also assigned to corresponding systems—positions in 6 DOF to the navigation systems and the azimuth angles and RPMs to the thruster arrangements. The remaining digital twin contents were sparse to justify use of full-fledged taxonomies, so they were identified with suitable tags/labels. This applies to environment states, which consists only in wave data. Similarly, response operator files and web applications were identified as "motion response" and "maneuvering", respectively.

3.10. Aggregation into a web-based system

Fig. 8 schematizes the data architecture for execution of experiments, aggregating various subsystems into 5 (five) main components. Component 1 represents the DP system, including the motion capture and the controlled PSV model. Component 2 shows the water elevation measuring system. We developed a network bridge in Python (Component 3) to centralize communication between the measuring systems and the digital twin web application (Component 4) in both directions. The bridge performs two functions in the first direction—one to receive the motion and propulsion logs from the DP system to repass them to the web app, and the other to receive the water elevation signal, calculate the wave characteristics from it, and repass them to the web application. In the reverse direction, it communicates by receiving the DP setpoint that the user defined on the app and repassing it to the control system. When transmitting packets among systems, the bridge also establishes compatibility between transmission protocols. The web interface uses WebSocket, a protocol broadly used by modern streaming applications, and other systems use the older User Datagram Protocol. The last digital twin component is the database (5), which stores the digital twin files in a local computer folder. The web application can access the folder and retrieve relevant files during simulations. We did not include a specialized database system in the digital twin architecture to simplify development and deployment. The file-based approach was sufficient for this case study, which generated only a few megabytes of digital twin and experimental data. For operations with a large amount of data, a solution would be to include a robust commercial alternative such as Amazon Web Services into the data loop. The streaming architecture was later adapted to rewind experiments based on stored data. In practice, this greatly simplifies setup by removing the need for communication with the measuring systems, since the web application retrieves all relevant data from the database folder.

4. Digital twin functionalities

4.1. Web-based control dashboard

The web application (Fig. 8, Component 5) provides a user with a graphical interface during operation of the digital twin, as well as allowing remote access of the data and control of the physical asset. The interface displays four elements—a 3D scene at the center, a set of 2D plots at the bottom, a monitoring panel on the left and controls at the top right (Fig. 9). The 3D visualization shows wave states, ship position in 6 DOF, and propeller rotations in nearly real-time. All such states are listed on the leftmost panel. The 2D dashboard shows time-series for various motion modes, exemplified as heave and pitch in the figure. Top-right

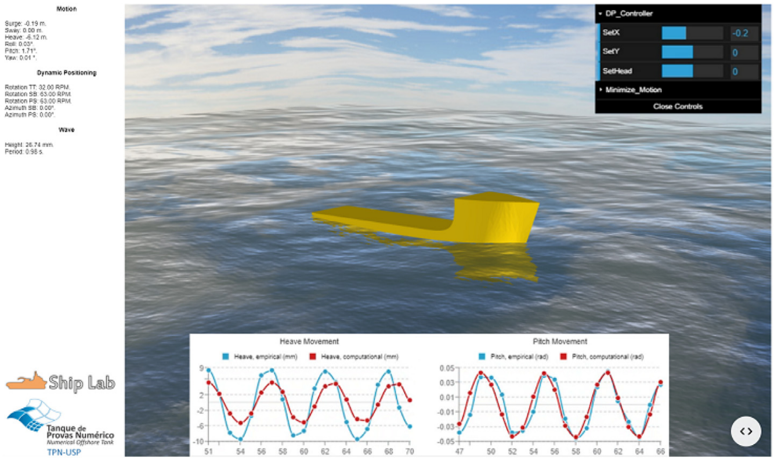


Fig. 9. Screenshot of the web-based digital twin dashboard.

elements provide DP controllers for the PSV model. The three graphic sliders allow a user to control the setpoint (i.e., two planar coordinates and the heading). A drop-down menu contains a list of the model's six motion modes so that a user can select one that he/she wants to minimize.

4.2. Digital twin use cases

Monitoring and control. The most fundamental functionality of the digital twin is monitoring and control of the experiment (Fig. 10). This capability was useful during the experiments by allowing debugging and testing of the experimental setup. For example, it was used frequently to verify propulsion system functionality, especially when the model was inside the tank. It also helped to identify whether the model was drifting toward an undesired position, out of the field of view of the motion tracking system, and thus risking getting out of control by breaking the DP feedback loop. Such situations could be addressed quickly using the digital twin interface itself.

Motion validation. The app allows validation of motion responses that are estimated during design in comparison to those measured during the experiment. The 2D plots compare ship coordinates as measured during the experiment in real-time and expected coordinates as reconstructed from response operators; respectively, blue and red lines in Fig. 9. In practice, the application allows quick qualitative evaluation of the difference between expected and measured amplitude responses. Further development is needed to provide indicators at the level of detail that is required by a wave basin facility, namely quantitative comparisons of amplitude and phase for all 6 DOF.

Motion optimization. This functionality allows a user to adapt a response to waves for one decoupled motion mode. The application uses stored responses to minimize ship motion during station keeping by controlling its heading relative to the wave direction. While the PSV is floating during a wave condition, a user can select one of the six motion modes from a drop-down list, and once selected, the optimization algorithm searches for a heading that minimizes that motion mode and sends a command for the DP module to maneuver the vessel toward the specified heading. This type of optimization offers basic and self-evident results. For example, if the PSV is floating during head waves and the user selects an algorithm to minimize pitch motion, it automatically turns the model in the position of beam waves (Fig. 11). Although the maneuver minimizes pitch, the new heading also maximizes roll response. Using an advanced optimization algorithm, operators can minimize the coupled motion response on a strategic point of the vessel, e.g., the location on deck where a crane is performing a lifting operation.

5. Discussion

5.1. Digital twin as support to experiments

Robust remote control technologies are an essential enabler to a shore control center and thus a necessary step toward autonomous ship operations, especially when combined with algorithms for situational awareness and decision making [32]. The monitoring use case illustrates the potential of a digital twin to aid ship operation by allowing easier interpretation of states in comparison to traditional means, such as displaying logs and charts, or triggering alarms. This type of responsive interaction was enabled by a communication system that exchanged data with the physical setup at high frequency and low latency. We used a modular approach to link the DP control system to the digital twin. From one side, it reduced development effort and increased flexibility in comparison to the digital twin as a single integrated system. From the other, even for a simple digital twin compared to full-scale applications, it created unforeseen interactions during operation of distributed subsystems. This might happen, for example, if operators are unfamiliar with assumptions made during development of the modules, or if an operator overrides simulation parameters in a module and the system does not propagate new values to remaining ones. In our experience, these problems can be ameliorated by communicating assumptions clearly among operators and developers, either in the form of written software documentation or spoken instructions. At the industrial scale, these measures might be complemented by access permission management that suits each user's level of knowledge about the systems they are developing or operating.

5.2. Application of the standards

ISO 19848 standard was simple to understand and implement, and demonstrated a good degree of flexibility, being able to accommodate all data involved in the case study, including

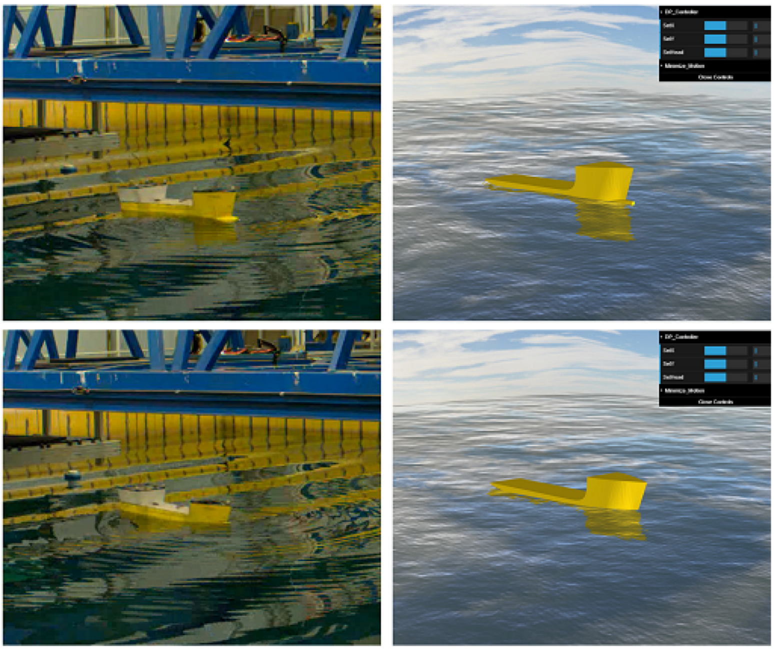


Fig. 10. Screenshots of the monitoring application in comparison to the physical experiment.

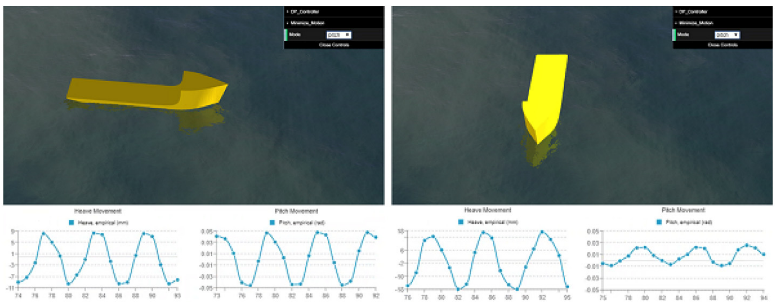


Fig. 11. Motion minimization function maneuvering the model toward a heading that minimizes pitch motion (beam waves).

setpoints and values calculated from raw observations. The document ensures consistency between descriptive metadata and the corresponding log, making sure they can be correctly interpreted by users. As a consequence of the document's scope, its application does not per se guarantee data cross consistency among different vessels. For those looking to automate creation of digital twins for a fleet of vessels, it is necessary first to verify every ship is doted with the necessary sensor setup (obviously) and then confirm they adopt the same naming rules with identical names for equivalent data channels. I.e., the ISO standard specifies a Local ID which should in principle be identical for the same kind of sensor in different ships, though in practice this parity depends on how strictly the adopted taxonomy constructs this ID. Although the document specifies that regular measurement intervals are expected for consecutive readings in tabular data, the data structure can accommodate readings at variable time steps without modification. A few problems arose when trying to reinforce the standard time format across measuring systems. ISO uses the convention defined by document 8601, which gives an unambiguous format for storage of date that can be interpreted regardless of variations due to, for instance, time zones or summertime [33]. Commercial instruments commonly follow internal conventions for time labels, sometimes by marking the time in seconds since measurements started during the current execution (i.e., zero seconds at the beginning of the experiment), complicating synchronization of time among distributed systems. However, this is a problem encountered when using instruments designed for a wave basin; the time conventions used by actual shipboard logs might be different.

VIS provided an adequate degree of detailing to map the ship representation inside a digital twin. The functional perspective has, so far, been consistent with the digital twin purpose by grouping components according to ship systems. Surveying the taxonomy during application suggested that it is expansive in terms of modeling variations of physical systems (e.g., mechanical or electrical propulsion). VIS also offers a good base in terms of mapping sensor logs, whether they are taken from a vessel component or navigation system, such as headings and positions.

Instead of opting for a stricter standardization, VIS provides flexibility when choosing the level in the hierarchy to which a sensor should be allocated and when creating a Local ID for a sensor, leaving margin to the creation of various alternatives with the same meaning. E.g., the document requires users to construct a suffix describing the quantity being measured. As the suffix is not standardized, an exhaust gas inlet could be denoted by ''ExhaustGasIn'' or ''EXH_G_IN''. The glTF format was considered as an alternative for storing geometric models as articulated assemblies that would simplify inclusion on a digital twin visualization. The format supports highly sophisticated structures with assemblies, sub-assemblies, etc. From one perspective, such sophistication is promising in allowing exchanges of models with articulations for movement; from the other, the hierarchies contained in a single file might conflict with or become redundant over the taxonomy chosen for the ship representation as a whole. For this reason, use of intricate glTF models needs to be guided by clear understanding of whether they suit a chosen taxonomy for asset representation.

5.3. Toward full-scale digital twin ships

There are several challenges to extending this early case study toward a full-scale digital twin. In terms of data management, if the approach outlined here is to be developed into a standard, it will be necessary to define templates for asset metadata and taxonomies (such as VIS) more formally, specifying all mandatory and optional fields the data structures should contain. There are also concerns about the capability of handling large amounts of content with the proposed framework. A related project is investigating development of platforms with support to multiple users and integration with databases, allowing digital twin content to be transmitted to the client on demand [34]. As the scope and complexity of a web-based project increases, it becomes challenging to manage and propagate changes on the code base. Technologies such as the TypeScript language (for static checking of variable types) and automated testing might alleviate this issue.

In terms of functionality, the use of a test basin made identification of wave characteristics much simpler compared to estimation of real ocean states. Ideally, a full-scale digital twin would be connected to external services providing weather conditions derived from *in situ* measurements or estimation models in various operational regions. Once the digital twin receives the vessel's geographic position, it would automatically search for the service covering the corresponding region and use the data (in this case, wave direction, significant period, and height) in the simulation. In practice, such architecture would require overcoming the challenges in data availability and access discussed earlier. At this stage, the use cases with motion validation and optimization serve as proofs of concept for showing how a digital twin can help close the data loop toward using design analyses during ship operation. In real situations, a vessel could have accelerations measured with sensors and linked to a decision aiding system, i.e., a reactive decision aiding system, rather than a proactive one. This could be done locally on the bridge or remotely on an onshore control center, though the later would require much work to ensure secure data exchange with controlled vessels. While we are not approaching these issues, current research on autonomous vessels may provide insights into the topic [35].

5.4. Open source approach and reuse of source code

From its beginning, the study aimed at an open-source approach to digital twin development, so most of the source code is publicly available on an online repository. There were however a few compromises. The control module for dynamic positioning, can be found in extant work [30], but was developed in Matlab, a proprietary platform. The motivation for using an open approach was the possibility of reusing source code to develop other projects, and such reuse can occur in various ways (Fig. 12). For example, a user can adapt the entire digital twin by linking it to an alternative wave basin with similar instruments to achieve the functionalities discussed in this paper. Once adapted, the digital twin can then be extended with new functionalities added to the original source code (Fig. 12, case 1). In another context, it might be useful to extract an excerpt of the source code, such as a function, visualization, or standard template, to develop a project with a different purpose (case 2). The open approach enables verification and reproduction of results, including access to data collected during experiments. To illustrate this, an application for experiment rewind is available on the public repository (case 3). From an engineering perspective, the application demonstrates how a digital twin can be used to review a previously executed operation and study the behaviors of the systems involved. From a research perspective, it illustrates how results can be reproduced from source code and data that are publicly available.

6. Conclusion and further research

As digital twins and other digital services increase in importance for the maritime value chain, it becomes necessary to establish frameworks and data structures to support modeling and storage while enabling data exchange and interoperability. The case study applied existing standards and web-technologies to the digital twin of a wave basin. The web app illustrated how digital twins can take advantage of dashboards and visualizations to communicate operational situations clearly, allowing effective response from operators. The approach was successful in yielding a functional digital twin web application which is compatible across digital devices and operational systems. When applying the ISO and DNV standards in practice, we found out that they have been designed to provide users with flexibility to customize their own implementations, even if this leaves room for reduced consistency among different digital twin projects. We judge this design compromise as adequate because it suits the tender-based character of data management in the maritime industry, thus increasing likelihood of adoption. It also implies that companies aiming to enable automated data processing across vessels will need to further specify their implementation approaches internally. In addition to using existing standards, the work proposed methods for modeling ship representation and handling data taxonomies. These methods require stricter formalization before they can be considered as standards, especially regarding specification of normative data structures. However, they contain principles that might inform future standardization initiatives.

The proposed framework encompassed only some of the aspects necessary to develop full-scale digital twins. It should be complemented by back-end integration with databases, standardized simulation models, access to quality metocean data, and secure infrastructure for remote operations. An important topic for future research is investigation into how the standards perform with complex digital twins, eventually scaling to a broader digital architecture. Development of a digital twin of NTNU's research vessel Gunnerus offers an opportunity to approach such issues using a case study that applies the standards to a real ship [36]. Various vessels' systems are currently instrumented with sensors, with shipboard logs streamed to a cloud platform. Thus, future research should develop a real-time monitoring application of Gunnerus, which would serve as a basis for a digital twin system of the vessel. From engineering and operation support perspectives, we investigate how taxonomies accommodate

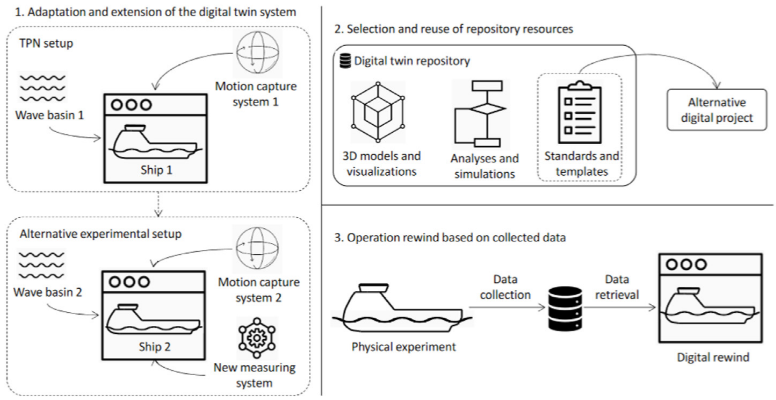


Fig. 12. Reuse cases for the digital twin system, source code, and experiment data.

a greater quantity and variety of data during ship operation. As this study did not propose standardized behavioral models, the digital twin linked various algorithms developed in different programming languages. Future work should investigate how digital twin data could be linked to simulations in a systematic manner by making use of standardized application programming interfaces.

7. Source code and open data

The digital twin source code, experiment data and documentation are available on: https://github.com/shiplab/dt_cv.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Mathematical modeling of behavior

A.1. Dynamic positioning control

The DP module was reused from work [30]. It applies robust techniques to control the ship's position while filtering disturbances from wave motions, ensuring stable performance across environmental conditions. During the current experiment, the algorithm was applied to the control of a single PSV model, but it has been also tested with consensus control of complex operations, during which vessels in a fleet moved in coordination. Fig. A.13 shows the architecture of the DP control module in connection with PSV operation. The optical motion tracking system parses PSV positions in 6 DOF, streaming them to the DP control algorithm, which estimates the propellers' rotation rates to maneuver the ship toward a user-defined setpoint and sends them as a command to the PSV over the radio frequency link (Fig. A.13, Component 1). This process operates iteratively at 100 Hz. For simplicity of the mathematical problem, the algorithm locks both azimuth angles to a neutral position, making the system of maneuvering equations determinate. This led to sub-optimal use of the propulsion system in terms of energy consumption, but it did not impair the DP functionality for the purposes of this study.

A.2. Extraction of wave characteristics

The wave model extracts wave characteristics from water elevations using a simple approach (Fig. A.14). The algorithm receives the stream of water elevations and begins storing crests and valleys through the signal using corresponding time labels (stage 1). Once the water elevation crosses the zero line, the algorithm identifies that a wave cycle ended and retrieves the last two saved values, one crest and one valley, in whichever order, to estimate the wave height, period, and phase — that is, whether the newly detected zero-crossing is an up-crossing leading to a new crest or down-crossing leading to a new valley (stage 2). After characteristics of the first wave cycle are received and saved, the algorithm updates them only when the new wave height or period has a difference of more than 2% in relation to it. This tolerance was implemented to avoid excessive identification of new waves due to light, random noise in water elevation readings.

The calculation of wave characteristics is performed by the network bridge (Fig. 8, Component 3) to reduce the necessary bandwidth and computational effort required from the web app client (Fig. 8, Component 4). To validate the algorithm, we generated a regular wave state with known characteristics in the tank and compared its values to parameters that the algorithm extracted. We set the flap system to generate a wave period of 0.978s with 28.6 mm of height. Under these conditions, the digital twin algorithm extracted from the wave signal a period of 1.020s and a wave height of 26.8 mm, a difference of 4.3% and 6.2%, respectively. We attribute these differences mainly to variations between the wave characteristics given as input to the wave generator system and the wave state that occurred in the wave tank, especially because it might suffer minor interference from the reflection on the tank's walls. The random variations in readings of the water elevation measuring system might also have contributed to a lesser extent.

A.3. Handling of response operators

Identification and retrieval. Once the web app client receives current wave characteristics, it searches and retrieves corresponding operators for validation of motion responses from the stored results (Fig. 8, Component 5) according to the following steps:

1. Scale wave period to full size.
2. Search closest period among the stored results.
3. Build time series from results.

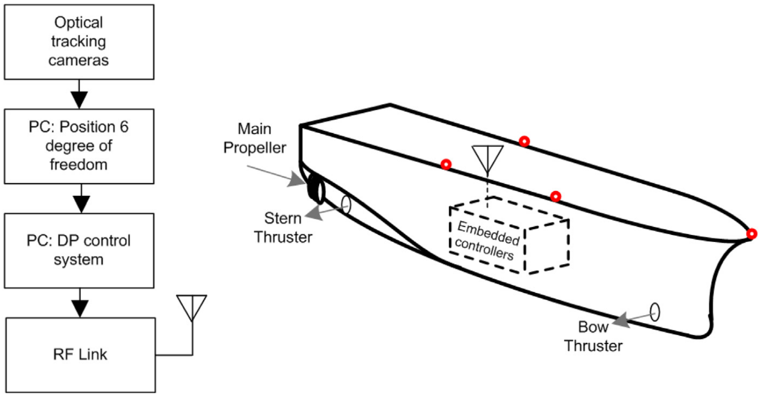


Fig. A.13. Scheme of the DP control module architecture [37].

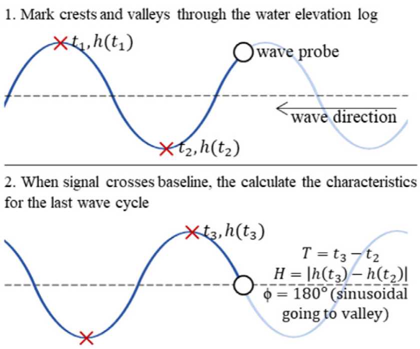


Fig. A.14. Extraction of wave characteristics from the water elevation log.

In step 1 the algorithm scales the current wave period to real-life size according to Froude relation:

$$T_{Real} = \frac{T_{Scaled}}{\sqrt{coeff}}$$

Step 2 searches for the period in the JSON schema that approaches the calculated value most closely. Taking, for example, the wave period mentioned above, 1.020s, the algorithm will scale at:

$$T_{Real} = \frac{1.020}{\sqrt{1/70}} = 8.534$$

The algorithm searches for the closest period stored in results, which in this case is 8.666s, and, considering the PSV is floating with a known heading in relation to the waves, it retrieves the corresponding amplitude and phase operators. They are used to construct a time series that is plotted graphically for visual comparison with experimental measurements in step 3. This is done with a simple sinusoidal curve:

$$\eta(t) = RAO \cdot A \cdot \sin\left(\frac{2\pi}{T}t + \phi\right)$$

where RAO is the response amplitude operator, A is the wave amplitude, and ϕ is the phase angle.

Search for minimal motion response. The optimization algorithm in the web app can be executed to minimize the vessel's decoupled motion response according to the steps:

1. The user selects as input one of the six motion modes to be minimized.
2. The algorithm searches for the heading that minimizes the response for the selected mode.
3. The algorithm sends command to the DP module to maneuver the ship toward the heading.

As verification of the algorithm, we conducted tests during which minimized results were known and which confirmed that the algorithm performed the search and selection correctly. See Section 7 for a link to the scripts the last two appendices discuss.

References

[1] Defense Acquisition University. Glossary of defense acquisition acronyms and terms. 2021, https://www.dau.edu/glossary/Pages/Glossary.aspx. (Accessed 11 June 2021).
[2] Fonseca ÍcaroA, Gaspar HM. Challenges when creating a cohesive digital twin ship: a data modelling perspective. Ship Technol Res 2020;1–14. http://dx.doi.org/10.1080/09377255.2020.1815140.
[3] Whitfield R, Duffy A, York P, Vassalos D, Kaklis P. Managing the exchange of engineering product data to support through life ship design. Comput Aided Des 2011;43(5):516–32. http://dx.doi.org/10.1016/j.cad.2010.12.002.
[4] Gaspar HM. Data-driven ship design. In: 17th Conference on computer and IT applications in the maritime industries. 2018, p. 426–39.
[5] Låg S, With SB. Standardisation as an enabler of digitalisation in the maritime industry. 2017.
[6] Gaspar HM. Javascript applied to maritime design and engineering. In: 16th Conference on computer and IT applications in the maritime industries. 2017, p. 253–69.
[7] Rachuri S, Subrahmanian E, Bouras A, Fenves SJ, Foufou S, Sriram RD. Information sharing and exchange in the context of product lifecycle management: Role of standards. Comput Aided Des 2008;40(7):789–800. http://dx.doi.org/10.1016/j.cad.2007.06.012.
[8] ISO. 19848 Ships and marine technology – standard data for shipboard machinery and equipment. Geneva, Switzerland: International Organization for Standardization (ISO); 2018.
[9] Erikstad SO. Merging physics, big data analytics and simulation for the next-generation digital twins. In: 11th Symposium On high-performance marine vehicles. 2017, p. 139–49.
[10] West TD, Blackburn M. Demonstrated benefits of a nascent digital twin. INSIGHT 2018;21(1):43–7.
[11] Erikstad SO. Designing ship digital services. In: 18th Conference on computer and IT applications in the maritime industries. 2019, p. 354–63.
[12] Robinet F, Arnaud R, Parisi T, Cozzi P. Gltf: Designing an open-standard runtime asset format. In: GPU Pro 360 guide to 3D engine design. A K Peters/CRC Press; 2018, p. 243–60. http://dx.doi.org/10.1201/9781351172486-20.
[13] IEC. Maritime navigation and radiocommunication equipment and systems – digital interfaces – part 1: single talker and multiple listeners. Geneva, Switzerland: International Electrotechnical Commission (IEC); 2016.
[14] ISO. 19847 Ships and marine technology – shipboard data servers to share field data at sea. Geneva, Switzerland: International Organization for Standardization (ISO); 2018.

[15] Ando H. Digitalization in the maritime industry. ClassNK Tech J 2019;1(1).
[16] Vindøy V. A functionally oriented vessel data model used as basis for classification. In: 7th International conference on computer and IT applications in the maritime industries, compit, Vol. 8. 2008, p. 60–9.
[17] Smith B. Against idiosyncrasy in ontology development. Front Artif Intell Appl 2006;150:15–26.
[18] Quaeghebeur E, Zaaijer MB. How to improve the state of the art in metocean measurement datasets. Wind Energy Sci 2020;5(1):285–308. http://dx.doi.org/10.5194/wes-5-285-2020.
[19] Sandvik B. three.geo: Geospatial data support in three.js. 2021, Available at https://github.com/turban/three.geo. (Accessed 11 June 2021).
[20] ISO. 19901-1 Petroleum and natural gas industries – specific requirements for offshore structures – part 1: metocean design and operating considerations. Geneva, Switzerland: International Organization for Standardization (ISO); 2015.
[21] Schaap D. SIMORC - system of industry metocean data for the offshore and research communities. In: OCEANS 2007 - Europe. IEEE; 2007, p. 1–2. http://dx.doi.org/10.1109/oceanse.2007.4302405, URL http://www.simorc.com/.
[22] Hatledal LI, Skulstad R, Li G, Styve A, Zhang H. Co-simulation as a fundamental technology for twin ships. Model Identif Control Nor Res Bull 2020;41(4):297–311. http://dx.doi.org/10.4173/mic.2020.4.2.
[23] Fonseca ÍA, Gaspar HM. A prime on web-based simulation. In: Iacono M, Palmieri F, Gribaudo M, Ficco M, editors. ECMS 2019 proceedings. ECMS; 2019, p. 23–9. http://dx.doi.org/10.7148/2019-0023.
[24] Andrews D, Dicks C. The building block design methodology applied to advanced naval ship design. In: Marine design conference. 1997, p. 1–8.
[25] Levander K. System based ship design. In: NTNU marine technology.(seakey naval architecture). 2012.
[26] Pal M. Ship work breakdown structures through different ship lifecycle stages. In: International conference on computer applications in shipbuilding. 2015.
[27] Xantic. SFI group system - a system for classification of technical and economic ship information. Product description. 2001.
[28] de Mello P, Carneiro M, Tannuri E, Kassab F, Marques R, Adamowski J, et al. A control and automation system for wave basins. Mechatronics 2013;23(1):94–107. http://dx.doi.org/10.1016/j.mechatronics.2012.11.004.
[29] Mello PCd. Sistema de automação e controle para tanques oceânicos com múltiplos atuadores (Ph.D. thesis), University of São Paulo; 2012.
[30] Ianagui ASS. Robust system design for consensus control in dynamically positioned vessel fleet (Ph.D. thesis), University of São Paulo; 2019.
[31] WAMIT. Wamit, inc. - the state of the art in wave interaction analysis. 2021, https://www.wamit.com/. (Accessed 17 December 2021).
[32] Hoem ÅS, Rødseth ØJ, Johnsen SO. Adopting the CRIOP framework as an interdisciplinary risk analysis method in the design of remote control centre for maritime autonomous systems. In: Advances in safety management and human performance. Springer International Publishing; 2021, p. 219–27. http://dx.doi.org/10.1007/978-3-030-80288-2_26.
[33] ISO. 8601 Date and time — representations for information interchange — part 1: basic rules. Geneva, Switzerland: International Organization for Standardization (ISO); 2019.
[34] de Oliveira FF. An open web platform for ship analysis and simulation. Norwegian University of Science and Technology; 2021.
[35] Kavallieratos G, Katsikas S, Gkioulos V. Cyber-attacks against the autonomous ship. In: Computer security. Springer International Publishing; 2019, p. 20–36. http://dx.doi.org/10.1007/978-3-030-12786-2_2.
[36] Fonseca ÍA, Gaspar HM. An open framework for data taxonomies in digital twin ships. 2022, (In Press).
[37] Ianagui ASS, Mello PCD, Tannuri EA. Robust output-feedback control in a dynamic positioning system via high order sliding modes: Theoretical framework and experimental evaluation. IEEE Access 2020;8:91701–24. http://dx.doi.org/10.1109/access.2020.2994515.

Paper III

Submitted to International Journal of Maritime Engineering. The first round of reviewer's comments was received by the candidate in July 12, 2022. A new version of the manuscript with the requested corrections is being prepared for re-submission.

This paper is awaiting publication and is not included in NTNU Open

Paper IV

Sergi Escamilla i Miquel et al. (July 2020). 'An open-source library for hydrodynamic simulation of marine structures'. In: *Marine Systems & Ocean Technology* 15.3, pp. 160–174. DOI: 10.1007/s40868-020-00083-3

Marine Systems & Ocean Technology (2020) 15:160–174
https://doi.org/10.1007/s40868-020-00083-3

ORIGINAL PAPER

An open-source library for hydrodynamic simulation of marine structures

Sergi Escamilla i Miquel[1] · Ícaro Aragão Fonseca[1] · Henrique Murilo Gaspar[1] · Daniel Prata Vieira[2]

Received: 9 December 2019 / Accepted: 8 July 2020 / Published online: 28 July 2020

Abstract
The work focuses on an open and collaborative approach for hydrodynamic simulations of multibody operations. It builds on Vessel.js, an existing web-based ship design library, by modeling the interaction between entities and creating multibody models able to output different responses. To develop the cases here studied, the simulations are decomposed into single elements to understand their behavior separately before making them interact with other elements to create a multibody simulation. In the process, different hydrodynamic models are used to analyze the bodies according to the requirements of the simulations and the needed level of complexity. The simulations are coded in JavaScript and visualized in a web environment, with the option of using external hydrodynamic analyzes, which in this work were exemplified using a commercial software that adopts the linear potential wave theory. The paper concludes with a discussion about future applications of methods and simulations.

Keywords Multibody · Open source · Hydrodynamic · Simulation · Marine structures · Marine operations

1 Open hydrodynamic simulations

This study presents new developments of Vessel.js, an open-source library introducing methods for simulation of vessels and marine operations with a web-based approach [1]. In conjunction with other tools and libraries, Vessel.js allows the creation of simulations composed of individual entities such as ships, mooring lines, and hawsers. The library models the motions of such objects and their interactions to create simulations of multibody operations. In the process, the user can choose among the hydrodynamic models which meet the simulation purpose and requirements adequately.

✉ Ícaro Aragão Fonseca
icaro.a.fonseca@ntnu.no

Sergi Escamilla i Miquel
sergiescamilla@gmail.com

Henrique Murilo Gaspar
henrique.gaspar@ntnu.no

Daniel Prata Vieira
daniel.prata@tpn.usp.br

1 Department of Ocean Operations and Civil Engineering, NTNU in Ålesund, Ålesund, Norway

2 Numerical Offshore Tank, University of São Paulo, São Paulo, Brazil

As Vessel.js is open-source and web-based, the applications developed with it are easily accessible on the web, and its source code can be reused to create new simulations. The simulations make use of different analysis models: motion responses can be evaluated with closed-form expressions, by solving the equations of motion or with Response Amplitude Operators (RAOs) imported from external software packages. The hydrodynamic models and the web-based approach are brought together to perform multibody motion simulations more interactively compared to a traditional approach.

The following section presents the web-based approach and the principles guiding the development of the simulations. Section 3 introduces the available hydrodynamic models. Section 4 describes the process of assembling a simulation case with Vessel.js, also presenting the usage of other software when necessary. This process is applied to the case studies in Sect. 5. Section 6 provides guidelines for users who want to obtain and modify the Vessel.js source to create customized examples. Section 7 concludes the paper and describes current and future work.

2 Simulation and visualization

2.1 Web-based simulations

The web-based approach presents the advantage of making engineering simulations available to any user with a web browser connected to the internet. Web applications are supported by three key programming languages: JavaScript, which allows the execution of algorithm scripts on a web environment; HTML, which deals with document presentation; and CSS, which takes care of the style of the website. These open standards present a solution to several problems of compatibility between the software application, on one side, and different devices and operating systems, on the other[2, 3].

Being the most used programming language in the world, according to statistics by the hosting website GitHub, JavaScript enjoys a wide base of openly available resources, libraries, and documentation. Several of such libraries are used in this work, e.g., Three.js to create and render 3D graphics[4] and Numeric for the solution of ordinary differential equations[5].

JavaScript supports object-oriented programming, allowing the developer to define and organize variables and functions inside objects with an intelligible data structure[6]. In that programming approach, the code development relies on different objects, which work as encapsulated elements that can handle information, call functions stored as methods, make calculations, among various other operations. Different objects can be combined to accomplish a task or reused inside a script, for example, with instance patterns.

2.2 The Vessel.js library

The simulations presented in this work are developed with Vessel.js, an open-source library for ship design[1]. Vessel.js provides tools to represent a ship as an object in JavaScript. Top-down design can be done, starting with a hull, payload, and traditional equations[7]. Bottom-Up approaches, such as Andrews' design building blocks[8], are possible considering blocks that are created with "derived objects", which will have a parent called "base object". The derived objects are placed in the right coordinates to create an approximation to the vessel[9]. Every derived object has a state that collects the positions and changing characteristics of it. Once the ship is defined, these states can be used to simulate different behaviors. The library can calculate the weights (considering the lightweight and the filling ratio of the tanks) and, therefore, the overall displacement of the ship. Also, draft, hydrostatic, stability coefficients, and small angles of trim can be calculated as part of the state of the ship[10].

The library aims to develop code following a well-defined structure that is understandable for other users that might be interacting and improving the code in the future. The simulations are hosted on an open-source platform to be accessible and encourage a collaborative community (http://vesseljs.org). The website provides various examples where it is possible to interact with the simulations presented in the following sections. Additionally, other examples start to work on multi-entity configurations by including new bodies on the ship deck, having independent or coupled behaviors, such as a lifted load with a pendulum motion.

2.3 Simulation models in the library

The simulation approach employed in Vessel.js is based on the representation model for virtual prototyping in the design of engineering systems proposed by He et al.[11]. It is constituted of three elementary submodels (Fig. 1):

- Entity model, which defines the physical product to be simulated, including design specification data and information about the product, 2D, and 3D models.
- State model, which represents the entity model exposed to internal and external state constraints, such as a working position in a kinematics simulation, thus analyzing the entity by assigning it a state.
- Process model, which is an accumulation of the state models, representing a behavior over time, from the initial to the final state. This model can also be obtained by subjecting the entity model to a dynamic constraint.

This taxonomy was adapted for application to virtual prototyping of marine operations on previous works[9, 12]. The entity models represent a maritime system with the desired level of detail, such as the 3D model of a ship with a component specification or the characteristics and visualization of a mooring line. The state models represent the entities subjected to internal or external disturbances. For a ship, this could be the environmental conditions or instantaneous

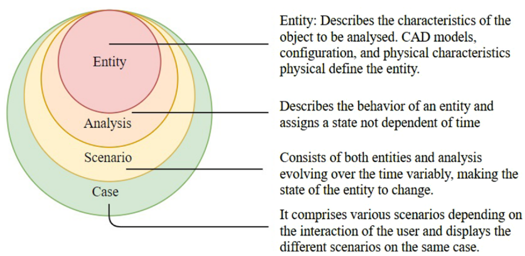


Fig. 1 Configuration of the simulation approach, from an entity to a case study

resistance at a given speed and loading condition. They are evaluated with analyses performed with Vessel.js or even with external software.

The process model is a sequence of states that change over time, offering a dynamic simulation scenario. Finally, the case studies comprise several scenarios that can be accessed by modifying simulation parameters to study the system behavior under different conditions, e.g., different wave characteristics. Figure 1 illustrates these concepts.

3 Hydrodynamic analyses

3.1 Modeling approaches

The simulator can be used for evaluation of motion behavior and visualization of motion results calculated with other software. There are three methods to accomplish that. Following the taxonomy presented previously, the state model represents static constraints or stimuli applied to an entity model. The states are stored in a specific object in the Vessel.js library and linked to 3D visualization. As the states evolve over, this state object is updated to account for this time variation, thus representing a scenario.

The first method calculates the motion response with closed-form expressions implemented inside the Vessel.js library. In the second method, the equations of motion are evaluated as the ship and the mooring lines move in the simulation. The motion coefficients are estimated based on the physical characteristics of the ship and the mooring lines modeled in the simulator. Some simplifications are taken that make this method only suitable for small motions of barge-shaped hulls.

However, hydrodynamic models are not always simple to be implemented directly in the source code. For more complex cases, a module was designed in which the user can supply the hydrodynamic coefficients through external inputs provided, for example, by any commercial software. This method allows the importing of RAOs evaluated externally, so the simulation can be modeled with Vessel.js and then animated with results obtained from other software.

These methods are detailed with their corresponding hydrodynamic models in the following sections, with emphasis on the third one which is later explored in the main case studies in this paper.

3.2 Ship motions with closed-form expressions

The first method, based on Jensen et al.[13], is a semi-analytical approach to derive frequency response functions for the wave-induced motions of monohull ships. This approach was developed to obtain a quick and close approximation of the wave-induced motions and accelerations in the conceptual design phase. Thus, it relies on parameters known during this stage of the design, such as length, breadth, draft, block coefficient, waterplane area coefficient, heading, and speed.

The method calculates the heave and pitch amplitudes with an analytical strip theory formulation by approximating the hull with a box-shaped vessel while neglecting motion coupling and assuming a constant sectional added mass equal to the displaced water. For the roll amplitude, the hull is modeled with a composition of two prismatic beams.

The Jensen's work presents a comparison between the proposed closed-form expressions, a seakeeping analysis based on the strip theory method, and experimental results from model tests. The comparison shows that the closed-form results are reasonably close to the other methods, except for the following cases:

- Heave is too small for $\lambda/L \leq 1$.
- Pitch is too large around $\lambda/L = 1$ for Froude numbers larger than 0.2.
- Roll is too large around the resonance frequency.

The results of the analysis are the amplitude response in heave, roll, and pitch for a given regular wave and heading angle. These are calculated with specialized methods in the Vessel.js library, and they access all the parameters necessary to perform the simulation inside the ship object. For that reason, the entire simulation can be performed on a web browser, and there is no dependence on external software for calculating the responses. The ship motion amplitudes in heave, roll, and pitch are converted to a time series by applying Eq. 1.

$$\eta_i = \xi_i \cdot A \cdot \cos(\omega t - \lambda_i), \tag{1}$$

where η_i is the vessel displacement (in meters or degrees), ξ_i and λ_i are the RAO amplitude and phase of the i-th degree of freedom (DOF), respectively, A and ω are the wave amplitude (in meters) and the wave frequency (in radians per second), respectively, and t is the time (in seconds).

3.3 Ship motions with differential equations

3.3.1 Vessel motion

Like the previous method, the motion calculation is performed totally on the web browser. The Vessel.js library provides time-domain models of motion response based on the equations of motion for the vessel. A work by Fossen and Fjellstad[14] was used as a reference for modeling of marine vehicles in six degrees of freedom (DOF). The model follows a Newtonian motion. The equation of motion is given as follows:

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu = \tau_{RB}. \tag{2}$$

The term τ_{RB}, which stands for a generalized vector of external forces in the six degrees of freedom, can be expanded, yielding the following equation of motion:

$$(M_{RB} + M_A)\dot{\nu} + (C_{RB}(\nu) + C_A(\nu))\nu + B(\nu)\nu + g(\eta) = \tau, \tag{3}$$

where M_{RB} is the rigid body inertia matrix, M_A is the inertia of the added mass, $C_{RB}(\nu)$ is the rigid body Coriolis, centripetal matrix $C_A(\nu)$ is the hydrodynamic added Coriolis, centripetal matrix $B(\nu)$ is the hydrodynamic damping matrix, $g(\eta)$ is the vector for generalized gravitational and buoyancy forces, and τ is the vector of external forces. The formulation was originally developed for control of marine vehicles, so it considers two frames of reference, one global and the other body-fixed, requiring the addition of the Coriolis matrices to the equations of motion. This approach makes the formulation well suited for future expansion with maneuvering forces, currents and others, thus being applicable to a wide range of marine operations.

Even with the equations of motion in place, the estimation of motion parameters is still challenging due to the estimation of appropriate motion coefficients. For that reason, a series of assumptions were made to simplify their estimation[15], and, at the moment, the motion simulation is only suitable for small motions of a barge-shaped hull. It does not yet account for wave interaction. The inertia and restorative coefficients are derived from the hull shape, while the damping coefficients need to be entered manually by the user. The equation is solved in synchrony with the 3D animation, with variable time steps.

3.3.2 Mooring line motion interaction and visualization

The interaction between ship and mooring can be modeled considering a mooring force applied by the cable to the vessel[16]. The model assumes a quasi-static behavior of the catenary mooring line with part of its length lying on the seabed, considering that only traction forces acting along the tangent are applied to the line. The model also disregards any force applied to the cable after it passes to semi-catenary geometry. Figure 2 shows a scheme with the model parameters.

The model considers that the vertical force applied by the catenary is given by the suspended length of the line multiplied by the linear density of the mooring line, ω. The method assumes that the total distance d is predefined by the user, but that the horizontal force H and the suspended length s are unknown. A given rope configuration can be solved by finding the a that satisfies the following equation:

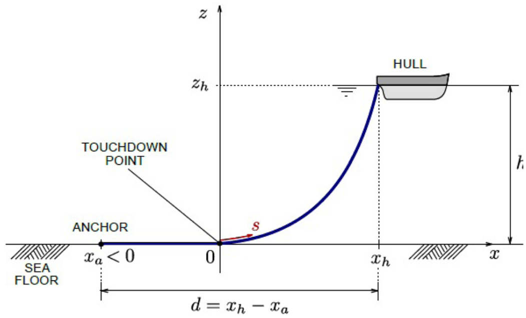


Fig. 2 Catenary mooring line configuration with relevant parameters[16]

$$l - a \sinh\left(\cosh\left(\frac{h}{a}\right) + 1\right) + a\left(cosh\left(\frac{h}{a}\right) + 1\right) - d = 0, \tag{4}$$

where l is the mooring line length and $a = H/\omega$. The equation is solved with an iterative method. For any given rope configuration, after the horizontal and vertical forces are calculated, they can be included in the vector of external forces in the equations of motion to simulate their interaction with the moored vessel.

3.4 Multibody motions

3.4.1 Importing external analysis results to Vessel.js

A third approach to performing simulations with Vessel.js is to rely on an external file with the RAO results to model the motion of the entities in the simulation. This allows the user to execute complex analyses with an external commercial software package of their preference and then visualize the results on the simulation environment. As the Vessel.js library is open, it can be linked to results incoming from any other software, as long as they are expressed as a textual list containing wave characteristics, i.e., amplitude, period, heading direction, and vessel response, i.e., amplitude and phase.

This procedure is exemplified here with the WAMIT (Wave Analysis MIT) software package, which solves the diffraction and radiation problem to analyze the interaction between waves and structures[17]. The software solves the velocity potential in the wet surface of the structure, and it is based on the linear potential theory[18], which solves the problem by using the Boundary Element Method (BEM) with three-dimensional panel elements[19]. The analyses described here do not account for second-order wave effects such as mean drift forces and moments. Additionally, the evaluation of restoring matrices for inclusion in WAMIT analyses is performed with a second software, Mooring Analysis Program (MAP++). The following sections are going to present the modeling principles of multibody simulations as a foundation for the usage of that software in the creation of simulations with Vessel.js. A comprehensive study on multibody dynamics simulation can be consulted in[20], for example.

3.4.2 Coupled motion of multiple vessels

When multiple vessels float in proximity, the motion of each vessel will affect the wave elevation field surrounding the other ones, a phenomenon called hydrodynamic interactions or hydrodynamic coupling, as pointed in[21]. To account for this, the motion of a system with N_{body} vessels is described to include $6 \times N_{body}$ degrees of freedom instead of only six. The coupled motion equation for the interacting vessels can be expressed with a set of $6 \times N_{body}$ coupled linear equations, as presented in Eq. 5.

$$\sum_{j=1}^{6\times N_{body}} [-\omega^2(M_{ij} + A_{ij}) + i\omega(B_{ij} + B^E_{ij}) + (C_{ij} + C^E_{ij})]\xi_j = X_i \text{ for} i = 1...6, \times N_{body} \tag{5}$$

where the subscripts i and j identify the elements of the following matrices: M is the mass-inertia matrix, A is the added mass matrix, B is the potential damping matrix, B^E is the external damping matrix, C is the hydrostatic stiffness matrix, and C^E is the external stiffness matrix. Furthermore, ξ_j is the motion complex amplitudes in the j-th DOF and X_i is the wave force or moment for the i-th DOF.

B^E is usually provided to account for viscous damping effects that are not calculated by potential theory. These values can be obtained through model-scale experiments or regression for similar cases. C^E generally represents the linearized restoration matrix due to the mooring system or due to any other mechanical coupling. To evaluate the motions ξ_j, as presented in Eq. 5, the BEM model provides the evaluation of A, B, C, and X_i, and the user must provide the external matrices B^E, and C^E.

3.4.3 Mooring line motion interaction and visualization

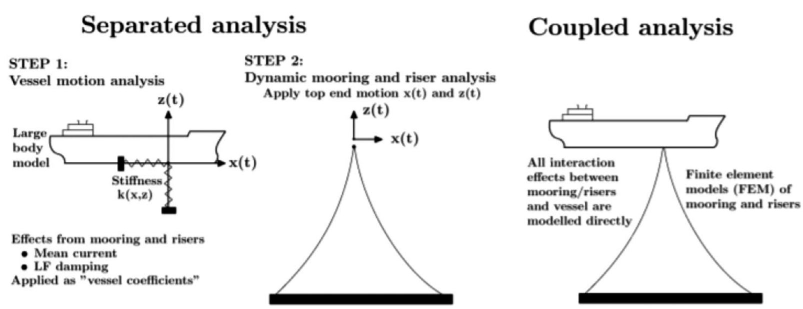
When evaluating the responses of multibody side-by-side configurations, it is essential to consider other aspects that can affect the interaction, such as the mechanical coupling between the bodies and other structures such as fenders, mooring lines, risers, and hawsers[22]. Ormberg and Larsen[23] proposed two different approaches to analyze multibody interactions, as seen in Fig. 3. The first one consists of analyzing the vessel and the mooring line separately, while the second is a coupled analysis which evaluates the whole system simultaneously.

On the difference between both approaches, they comment that "the turret motions estimated by a separated analysis also compare well with both coupled analysis and experiments if mean current loads and low-frequency damping from moorings and risers are included accurately. Otherwise, the use of separate analysis will severely underpredict the mean offset and overpredict low-frequency motions".

For simplification of the simulation method, this study will carry the motion analyses with a decoupled approach, even if, due to the approach limitations, this may lead to overprediction of frequency motions and underprediction of the mean offset. In the studied case, a vessel will be considered as a rigid body, and the ropes will be regarded as flexible bodies. For the sake of simplification, some of the bodies will have only 3 DOF, such as the ropes, where the rotations will be disregarded. The link between two or more bodies will be defined by kinematic constraints which restrict their motions by limiting the relative translation or rotation between two or more bodies. In general, the constraints were defined by setting the relative motion between two bodies. For the anchoring point of a mooring line to the seabed, however, a fully fixed constraint is applied.

For the mooring lines, the inertia forces are proportional to the acceleration, consisting of rigid body mass and added

Fig. 3 Coupled and decoupled approach[23]



mass. Also, if the load frequency is higher than the natural frequency, then the systems gain some inertia. These influences can be more accurately modeled as a linear restoring matrix accounting for the effect of such forces, with elements depending on the combination of motion for each degree of freedom.

However, given that these restoring coefficients are often complex to estimate due to the many factors they depend on, these calculations are made by external software (MAP++) developed by the National Renewable Energy Laboratory (NWTC). This software uses a theory for catenary lines based on a work by Peyrot and Goulois[24] to determine the external linear restorative matrices for mooring lines and hawsers. These mooring lines will act as spring for the ship, soothing and restraining its motion according to the coefficients for each degree of freedom.

In the examples using external hydrodynamic results, the catenary visualization is built using the model proposed by Irvine[25]. It assumes that the mooring system is composed of two parts: a half free hanging catenary and a line resting on the seabed. The model is more complex than the previous one, as it assumes elongation of the line and that the horizontal force applied to the cable is transmitted even to the resting section. For simplification of the calculations performed on the web browser, the visualization of hawsers was modeled as a straight line linking two floating systems.

4 Creating a simulation with Vessel.js

4.1 Information flow

The preparation of a simulation requires the modeling of all elements in the simulation taxonomy: entities, analysis, scenarios, and cases, as illustrated in Fig. 4.

The first step to create the simulation is to define the involved objects, such as ships, mooring lines, and ocean. Before proceeding to perform the required analysis, which will be used to evaluate the states of the entities, it is necessary to choose among one of the three models for motion

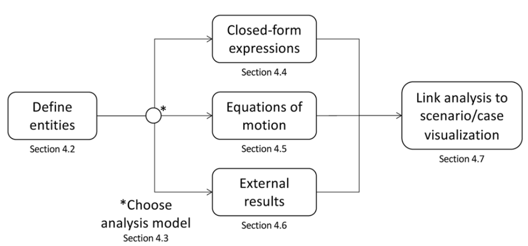


Fig. 4 Information flow for the preparation of a simulation case as discussed in the following sections

calculation discussed in the previous section. Once the analysis is functional, it can be linked to a visualization of the case with its scenarios to obtain a simulation that evolves with a dynamic character.

4.2 Defining entities

The simulation entities are the elements that are displayed and manipulated in the simulation. Some examples of entities included in the cases presented on this work are ships, ocean, seabed, mooring line, and hawsers. The entity visualizations are created and rendered with the Three.js library.

The most complex entity among these is by far the ship. It needs to be created with a Vessel.js specification written on the JavaScript Object Notation (JSON) file format. The specification defines the hull shape, structure, and other elements contained in the vessel. This data will be used by the analysis models in Vessel.js to derive the ship motion response. On the other hand, if the user plans to import the motion RAOs from external software, then it is not important to define the hull topology and the weight distribution accurately on the ship specification. In this case, a simplified barge-like geometry is enough to visualize the global motion response of the vessel.

Similarly, the mooring line needs to be defined with a geometric arrangement and physical characteristics on the Vessel.js library. The lines were divided into small segments to create visualizations of a catenary mooring line touching the seabed. A simple line geometry is created with several vectors containing empty positions. Then, the positions of each line segment are calculated and stored on these vectors, which are used to create the 3D catenary visualization.

Finally, the ocean and seabed entities are defined for visualization purposes. The simulation script synchronizes the 3D animation to ensure that the displayed wave corresponds to the motion response exhibited by the entities.

4.3 Choosing the analysis model

When developing a simulation, it is necessary to choose an analysis model that addresses the simulation purpose with adequate accuracy while avoiding excessive detailing. The different hydrodynamic methods provided described in Sect. 3 can be selected and combined according to these principles. The closed-form model can be used to perform the lightweight evaluation of motion response during the preliminary design stage. As the design is detailed, the simulation can incorporate accurate motion results from the external BEM software. The following sections detail how these different analyses methods can be used to calculate states in simulation scenarios.

4.4 Motion simulations with closed-form expressions

This method allows the user to compare the motion response between, e.g., different wave conditions, vessels, design proposals, or load conditions of the same vessel. It does not account for any kind of dynamic interaction between the vessel and other entities, e.g., interaction with mooring lines or shadow effect.

When executing the analysis, the web application accesses the required parameters on the ship object to derive hydrostatic and stability characteristics that are used to calculate the motion response. These characteristics are calculated and stored for the simulated ship. At every change of wave characteristic, they are consulted to evaluate the motion amplitude response for the new wave condition, allowing the simulation of the scenario. This calculation happens in real time as the user manipulates the simulation parameters. These parameters can be related to the wave condition or to the ship itself, e.g., its main dimensions.

4.5 Motion simulation with differential equations

The analysis model solving the equations of motion in real time on the browser is at the preliminary stage. It represents a direction for a future development of web applications with advanced hydrodynamic models.

The analysis allows the user to simulate the response of a barge to an initial excitation, with or without mooring lines. Similarly to the previous example, the analysis is performed based on the contents of a Vessel.js ship object. The calculation of hydrodynamic coefficients is suitable for small movement amplitudes of barge-shaped hulls and the motion coefficients are estimated based on the physical characteristics of the barge. It is complemented with damping coefficients that are entered as input by the user.

To specify mooring lines when creating a simulation, the user defines the number of lines and their geometric arrangement around the vessel. The simulation models their effect on the barge as external forces and moments to be included in the barge's equations of motion.

As the simulation does not yet account for wave interaction, it requires the user to provide an initial excitation to the barge so that the motion response can start. After that, the equations of motion will be solved to evaluate the barge oscillation, with an eventual motion with decay due to damping effects.

4.6 Multibody simulations with external RAO results

Boundary Element Method The creation of multibody simulations with external hydrodynamic evaluation requires the chaining of several tools, so the hydrodynamic coefficients can be imported to the Vessel.js simulation. An overview of the calculation process used in this study is shown in Fig. 5, where WAMIT is integrated with a custom TPN-Petrobras software[26] to simulate side-by-side operations. A typical analysis requires as input the wet-surface 3D panels mesh, the mass-inertia matrix of each vessel, and the periods and the wave incidence angles. The software deals with two main subroutines: POTEN, which solves the velocity potential of the body, and FORCE, which evaluates physical parameters such as force and motion coefficients, fluid pressure, velocity, and free surface elevation.

The software outputs hydrodynamic data such as added mass, potential damping coefficients, restoring terms, wave exciting force (calculated via Haskind's Relation), and 6 DOF motions for a given geometry in a specific wave period and direction.

Mooring analysis If the ship motion is constrained by mooring lines or hawsers, these elements need to be included in the model with proper restoring matrices. These matrices

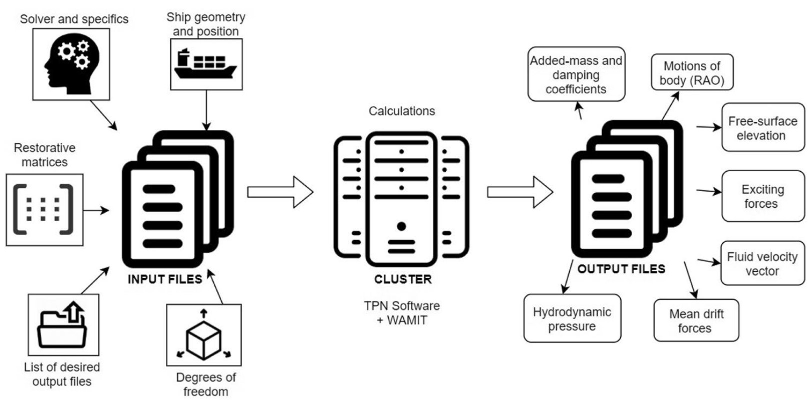


Fig. 5 Process followed to run the cases on this study with the custom TPN-WAMIT software

are calculated with the software MAP++, which receives as input a file with the characteristics of the mooring lines and the fairlead position on the vessel.

Similarly, the inclusion of hawsers on the operation requires the addition of an external linear restorative matrix to the motion model of both ships connected by it. The stiffness matrix, in that case, is fully coupled, and the matrix is computed by assuming that the displacement of the attaching point from one vessel is the same as the negative displacement of the attaching point of the other, resulting in a 12 x 12 stiffness matrix, i.e., 6 DOF for each body.

Multibody analysis The BEM software is executed with the associated inputs, which include the analysis setup, ship geometry, and restorative matrices for the mooring lines, among others. When more than one vessel is simulated in a side-by-side configuration, a resonant effect occurs on the free surface elevation in the gap between vessels, as pointed by[27]. So a new body consisting of an artificial rectangular damping lid is placed on the gap, and external damping can be inserted in the model, as suggested by[28]. Theoretical background of this dampening method can be obtained in[29].

An example of mesh configuration is presented in Fig. 6, where the gray panels represent two adjacent barges, and the blue panels represent the lid surface. Generally, the lid length has the order of magnitude of the shortest vessel length.

Interface BEM-JavaScript It is necessary to establish an interface between the BEM software and JavaScript to use the analysis results to create simulation scenarios with Vessel.js. The web application reads the data from the text file and turns it into an object containing the amplitude and phase RAOs, separating them according to the different degrees of freedom, number of ships, number of wave periods, and headings.

There are two ways to link these results to the visualization. In the first one, a display with a pre-determined configuration of vessels and mooring lines is created. The user is then able to upload the RAOs obtained with the external software. In this case, the user needs to be aware of the adequate data formats and be sure that the uploaded data is representative of the operation being simulated. In the second approach, the visualization is already pre-loaded with a data file containing the results from different simulation scenarios. This option is less flexible, but is also simpler to use.

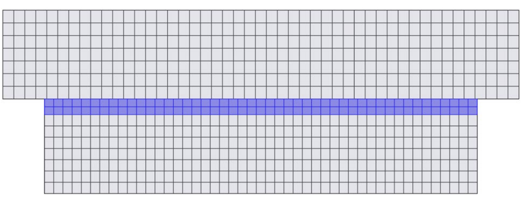


Fig. 6 Example of mesh used as input for the hydrodynamic model considering a lid for adding external damping in gap between vessels. Gray panels: adjacent barges; Blue panels: damping lid (top view) (Color figure online)

4.7 Linking analysis to scenario visualization

The scenario visualization is created to represent the simulation entities, which may include ships, mooring lines, hawsers, ocean, and seabed. The ships can be positioned with different heading angles and locations in the scenario. When adding a mooring system, it is essential to certify that its geometric arrangement in the visualization, including fairlead points and catenary ropes, is consistent with the one assumed when preparing the analysis models.

Once the simulation is started, the simulated states are continuously tracked at every time step of the visualization. At the same time, the user can modify simulation parameters to observe the system behavior in different scenarios. For the scenarios which model the ship motion as an amplitude response, i.e., closed-form expressions and RAOs, the motion is converted to a sinusoidal time series, as in Eq. 1. In the case of simulations evaluating the equations of motion, the behavior is animated by solving the equations with variable time steps delimited by each frame of the visualization.

The mooring line visualizations are also updated as the simulation progresses. As the fairlead of a given line moves, the positions of the segments constituting the lines are recalculated, and their vertices in the 3D visualization are updated.

5 Simulated cases

5.1 Overview

The cases are presented following a progression that goes from more straightforward simulations to more complex ones. The first two cases present the motion simulation performed with closed-form expressions, which can be achieved with one or more vessels. The two following ones show simulations with equations of motion solved in real time on the web browser. Then, the following cases show various simulations created with hydrodynamic analysis imported from external software, including interactions with mooring lines and hawsers in offshore operations. The following list summarizes these cases:

1. Closed-form expressions:
   (a) Single vessel (Sect. 5.2).
   (b) Multiple vessels (Sect. 5.3).

2. Differential equations solved in real time:
   (a) Free-floating barge (Sect. 5.4).
   (b) Moored barge (Sect. 5.5).
3. RAO imported from external software (verification of BEM results in Sect. 5.6):
   (a) Free-floating FPSO (Sect. 5.7).
   (b) Side-by-side operation (Sect. 5.8).

The accompanying visualizations were developed with a focus on intuitive user interaction. The graphic user interfaces (GUIs) show sliders that allow configuration of sea state, number of ships, presence of hawsers, and mooring lines, among others.

5.2 Motion of a single vessel with closed-form expressions

The first case is developed to simulate the motion response of a single ship in regular waves with closed-form expressions. The entities included in this simulation are one vessel and an ocean. The ship is defined as a PSV (Platform Support Vessel) specification with 106 derived objects and a simplified hull shape, which is automatically generated from a table of offsets stored in the specification. The ocean is governed by a regular wave with amplitude, period, and heading direction configurable by the user. The wavelength is derived automatically from the period with the dispersion relation for deep waters.

Figure 7 shows a screenshot of the simulator. The user can navigate through different simulation scenarios by adjusting the sliders on the GUI. Besides modifying the wave parameters, the sliders also allow the user to scale the main ship dimensions on the three coordinate axes, i.e., length, beam, and depth. Doing so automatically calculates the new vessel weight distribution, hull shape, and updates the motion response accordingly. During the early design stage, this functionality can be used to quickly visualize the effect of variations of main dimensions on the motion response.

5.3 Motion of various vessels with closed-form expressions

The previous case can be expanded to account for the motion of multiple vessels simultaneously to compare their responses. As the motion of individual vessels is governed by closed-form expressions, the simulation does not account for interactions such as the shadow effect between them. Similar to the previous example, the user can adjust the simulation scenario using the GUI sliders, but now the option to control the number of vessels in the scenario is also provided. The application supports simulation of dozens of vessels simultaneously, as Fig. 8 illustrates. The user can also upload external vessel specifications to be included in the simulation case.

5.4 Simulation of the radiation problem

This case models the radiation problem for a barge floating on still water. It considers that the barge is stationary until an initial motion disturbance applied by the user will start a movement that is dampened by the interaction with the water[15]. The entities included in this simulation are a box-shaped barge and a calm ocean. The analysis is based on the differential equations of motion with 6 DOF.

The simulation view includes the box-shaped barge placed on a flat plane representing the ocean entity for simplification of the visualization model, and the ocean does not display the excitation waves generated by the barge motion. Figure 9 shows a screenshot of the simulation. The initial conditions for heave, roll, and pitch, i.e., the modes which

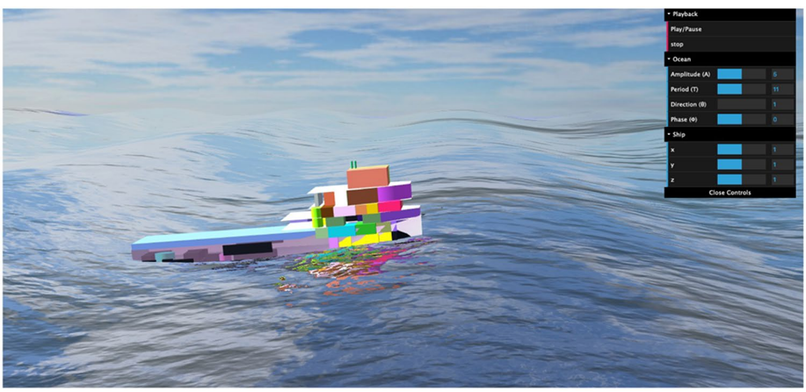


Fig. 7 Simulation of motion response for a single vessel with closed-form expressions

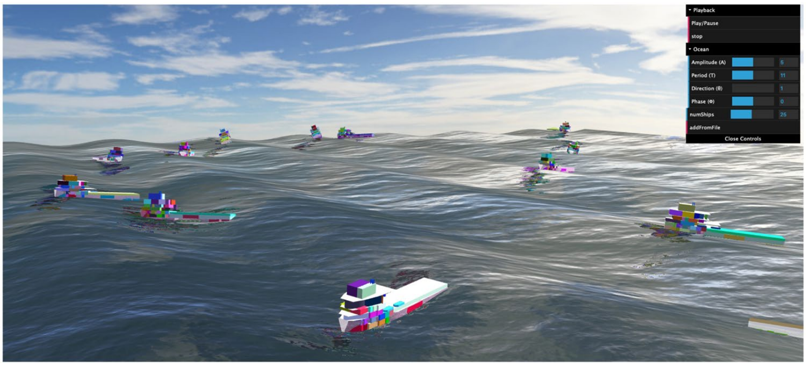


Fig. 8 Motion response scenario with 24 vessels

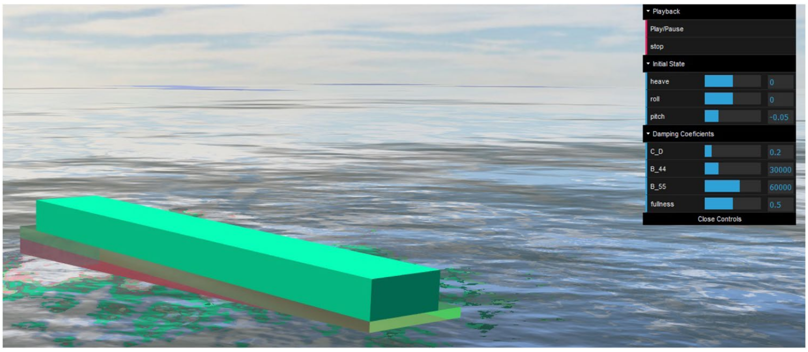


Fig. 9 Simulation of the radiation problem based on differential equations, with the barge subject to an initial pitch motion

have restoring effects, can be modified by the user. The motion starts after the initial state is set with the sliders and then is dampened as fast as the coefficients allow.

Alternatively, the case can be assembled with hydrodynamic coefficients obtained from external software. For this, an analysis resembling the radiation problem needs to be prepared. The diffraction problem is set to null, so the BEM software can identify the intention to solve only the radiation problem. The number of periods and wave headings is set to zero to ensure a calm sea and a seabed depth is specified. The resulting hydrodynamic coefficients are written on the corresponding arrays in the simulation source code. The final simulation presents similar functionalities, but with more accurate added mass, damping, and restoring matrices.

5.5 Interaction with mooring lines

By expanding the previous case, the simulation can include mooring lines anchored to the barge to restrain its motions. The mooring lines are new entities added to the 3D visualization and the barge's equations of motion. The resulting simulation behaves similarly to the previous case, but now accounting for a simplified mooring interaction. Figure 10 shows a screenshot of the web application.

5.6 Motion response with BEM and verification of results

The following simulations in this section are carried with an FPSO (Floating, Production, Storage and Offloading Platform), and eventually, an additional Suezmax tanker simulating a side-by-side offloading operation. The characteristics of such vessels were defined based on typical dimensions and are listed in Table 1, while Table 2 shows the physical characteristics described for the mooring line. The FPSO is considered as being moored to the seabed, while the Suezmax vessels are only attached to the FPSO with hawsers. These lines exert motion interactions that need to be considered with adequate restoring matrices.

Four different models were evaluated in WAMIT:

1. FPSO only.
2. Suezmax only.
3. FPSO and Suezmax in Side-by-side.
4. FPSO and Suezmax in Side-by-side with mooring.

The RAO results in heave, roll, and pitch for the FPSO and the Suezmax are presented in Figs. 11 and 12, respectively. These figures show the results for two wave incidence

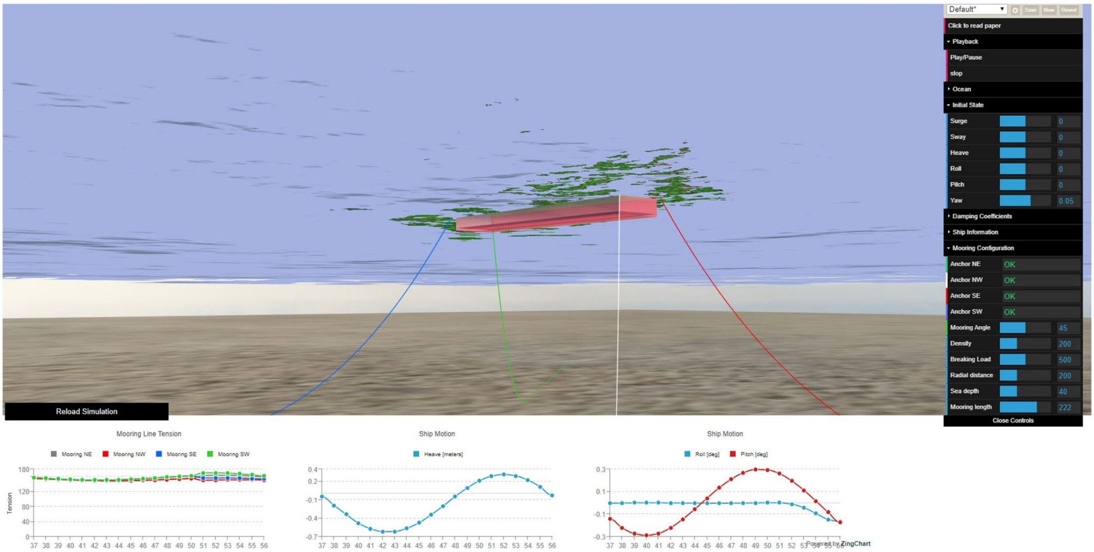


Fig. 10 Bottom view of the barge in the simulation with mooring line interaction calculated with differential equations

Table 1 Vessels main characteristics for FPSO and Suezmax for the simulated loading conditions

Characteristic	FPSO	Suezmax
Loading condition	Ballast	Loaded
Length overall (m)	316.5	264.8
Beam (m)	56.0	48.0
Depth (m)	23.0	18.0
Draft (m)	10.0	16.4
KG (m)	9.9	10.0
GM (m)	21.6	10.4
Displacement (t)	140000	175000
Ixx (t.m^2)	5.93E+07	4.97E+07
Iyy (t.m^2)	9.97E+08	7.68E+08
Izz (t.m^2)	1.05E+09	8.17E+08

Table 2 Characteristics of the mooring line used to calculate the restoring matrix

Property	Value
Material	Polyester
Density	0.8 kg/m
Elastic modulus	126 kN
Axial stiffness	$1{,}090{,}000{\cdot}D^2$

angles, 150 and 210 degrees, where the angle of 150 corresponds to the case in which the FPSO is protected from the waves by the Suezmax and the angle of 210 to the situation in which the Suezmax is protected by the FPSO. The angle

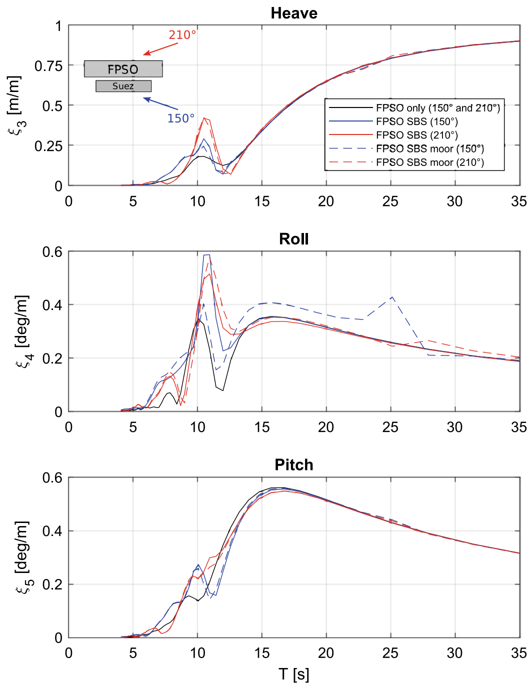


Fig. 11 Comparison of heave, roll, and pitch RAOs considering the FPSO isolated and in a side-by-side configuration without and with the linearized mooring effects

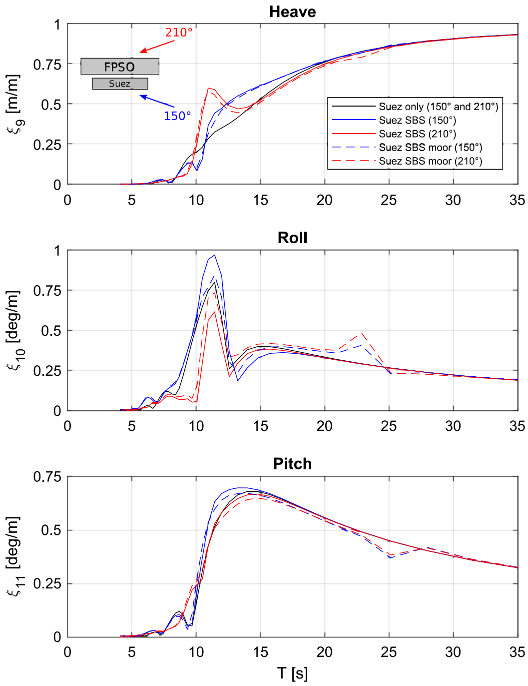


Fig. 12 Comparison of heave, roll, and pitch RAOs considering the Suezmax isolated and in side-by-side configuration without and with the linearized mooring effects

of incidence is measured from the stern (0°) up to the bow (180°) in a counterclockwise direction, according to[17].

In both figures, it is possible to observe that all coupled RAOs (blue and red solid lines) differ from the results obtained for the models in which vessels are isolated (black solid lines). These differences are found mainly in the region close to the natural periods of the ships, that is, for periods between 5 and 15 s. For both vessels, the most considerable differences were obtained for heave and roll motions. The pitch motion showed no essential differences.

In the case of the FPSO considering only the hydrodynamic coupling, it was observed that the heave motion is more significant for waves incident by 210°. The mooring effect, represented by dashed lines, is only evident in the roll motion, for which case a significant difference is observed for waves of 150°. Also, a peak in 25 s is found related to mechanical coupling due to the mooring system.

Considering the heave motions of Suezmax, it was observed that for periods up to 12 s, the movement for an incident wave of 210° is greater. This behavior probably occurs due to the waves radiated by the FPSO resonant heave motion. For periods over 12 s, the shadow effect is observed, and the motion of 150° waves is greater. The shadow effect is more pronounced for the roll motion throughout the region between 5 and 15 s. However, when mooring is considered, the difference between motions decreases. A peak in the region of 22 s is also observed due to the mooring coupling.

Similar results were obtained, for example, by Hu et al. [30]. Their work performed experimental research on the motions of an FLNG (Floating Liquefied Natural Gas platform) with similar dimensions and conditions to the ones discussed in this work. The researchers also compared the experimental results with the motions obtained with the software SESAM[31]. It was found that the model from the experiment produces similar results. However, the roll motion for the simulation in the current case is more pronounced for lower periods in comparison to the referenced study. The same applies, though with less severity, to the pitch motion. As for the other degrees of freedom, the amplitudes are quite close to the values obtained in this study.

5.7 FPSO motion response with BEM

This case presents the FPSO floating on an ocean with an incoming regular wave. A simplified barge-like entity was used to visualize the FPSO motion response. The case considers 5 wave amplitudes, 30 periods, and 25 different wave headings, being one heading every 15° until a 360° circle is completed. To simulate the FPSO motion, the RAOs in 6 DOFs obtained from the BEM software are imported to the web application. Once the platform starts moving, the results are fetched and converted to the corresponding time series.

This case is also taken as an opportunity to verify that the external RAO results are correctly interpreted by the web application. This was done by comparing the displayed motion amplitude to the values in the RAOs from the BEM software. For example, Fig. 13 shows the FPSO without mooring floating on a wave with heading of 150°, period of 10 s, and amplitude of 1 m. By comparing the movement amplitudes on the plot to the black RAO line in Fig. 11, it is possible to note that the amplitudes are correctly retrieved and displayed in the simulation.

5.8 Multiple vessels: side-by-side offloading operation

The last case models multiple vessels, including interaction with mooring lines and hawsers. An interface was developed where the user is able to visualize the motion of the moored FPSO only or of the FPSO with a Suezmax ship, as in a side-by-side operation. The user can adjust the desired case by scrolling sliders to add the Suezmax ship or the FPSO mooring to the seabed. For every user choice, the web application fetches the results from the corresponding hydrodynamic models, whether they are for one or two vessels, with or

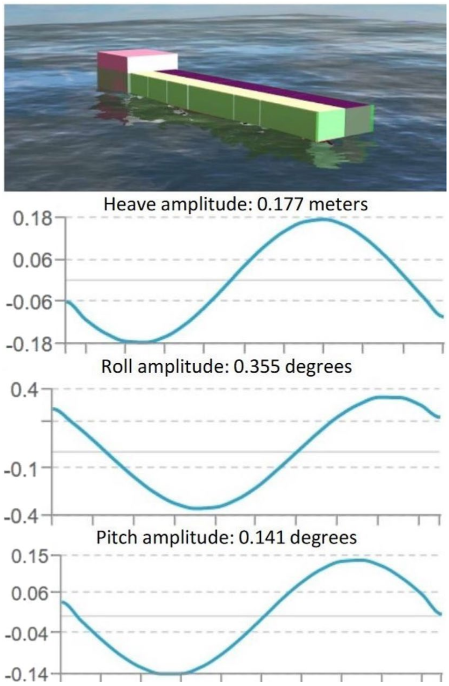


Fig. 13 FPSO motion as retrieved for a wave with 150° heading, 10 s period, and 1 m amplitude

without mooring line. The final interface is illustrated in Fig. 14.

6 Vessel.js as a tool to develop open and collaborative simulations

The cases in the previous section illustrate the ability to perform multibody simulations by following the proposed procedures. Furthermore, the taxonomy applied through the work establishes a common framework to organize the elements in different simulations. As Vessel.js is open and web-based, all the simulations and their source code are available for online access and can be downloaded, modified, and reused by any interested party. This allows users to take the case studies and build upon them for further applications using the available models.

To create a new multibody operation, the steps shown in Fig. 15 should be followed. First, the last version of Vessel.js library needs to be downloaded, where one will find the examples shown in the previous sections. Then, it is possible to choose the case that better suits the desired application and adapt it with new entities or setups by reusing the source code needed for simulating the marine operation.

In the process, the user should reason about which of the analysis models is suitable for the intended simulation. This may require an external hydrodynamic analysis to obtain accurate motion results, especially if new interacting bodies such as mooring lines or vessels in proximity are added to the simulation.

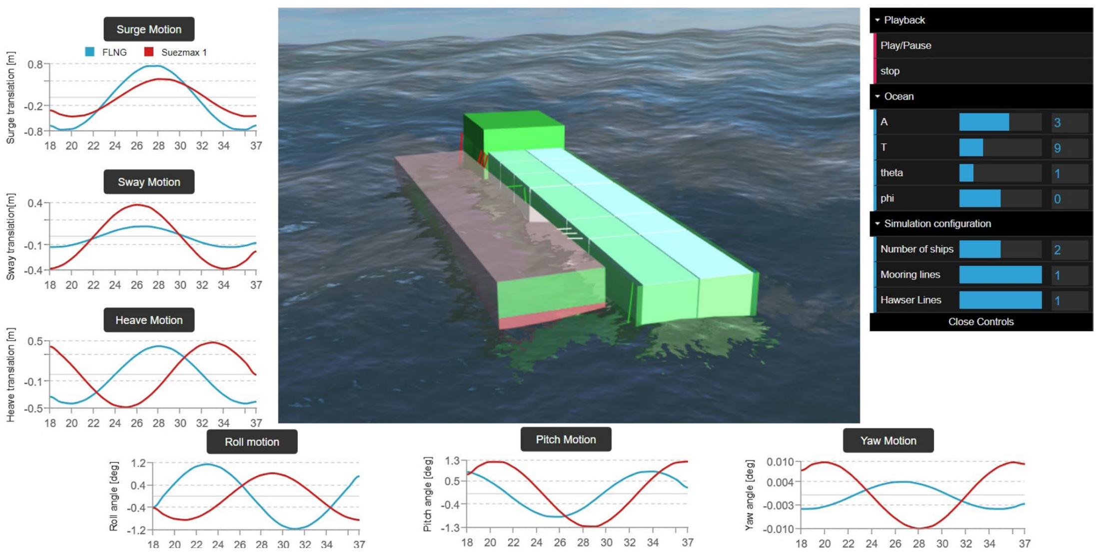


Fig. 14 Visualization of operation with hawsers showing plots for the motion of the two vessels in real time

Fig. 15 Process to create a multibody marine operation using the Vessel.js library

The simulation methods can have different applications depending on the stage of the life cycle when they are used. For example, during system design, they can be used to visualize the expected motion response in a given wave condition. During operation, it can be used to plan and discuss an operation before it is executed.

7 Conclusion and future work-closing the gap between design and operation

This paper presents functionalities for simulations of multibody operations developed into the Vessel.js library. The work uses a taxonomy that organizes the simulations by entities, analysis, and scenarios, which allow treating each entity separately so they can later be linked together to simulate an operation.

The simulations are open-source and developed with a web-based approach, making use of GUIs, 3D animations, plots, and other functionalities. The results are particularly promising because they were able to reconcile usage of well-established tools, e.g., WAMIT and MAP++, with an approach based on open standards ensuring that the simulations and visualizations can be accessed across devices and geographic locations. The cases were developed with the idea that other users can adapt and reuse the code freely, improving and creating new operations, and this is encouraged by the authors.

Previous sections discussed how the simulations can be used to support activities during different stages of the life cycle. Future work will focus on closing the gap between the design and operation of floating systems. In one front, the source code is being adapted to the development of digital twins. Digital twins are simulations that replicate the behavior of a physical asset during operation to provide decision support. A recent work was able to successfully implement a digital twin which mirrored an experiment with a scale model ship in real time. Further work will attempt to develop digital twins of real maritime systems by applying the same principles.

As it was not the objective of this work to validate computational results for any particular operation, it discussed the hydrodynamic behaviors based only on their feasibility and on similar works found in the literature. Another front of future work focuses on validating the side-by-side simulations based on experiments such as those presented in[22]. The study should help establish the usage of simulations to plan complex maritime operations, once the use of a virtual prototype is very necessary for the presentation of hydrodynamic behavior to other professionals involved in those operations as commanders, pilots, crane operators etc.

8 Source code and examples

The simulations presented in this work and other examples are available on the address http://vesseljs.org.

Acknowledgements Open Access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). This work was partly supported by the INTPART project for subsea development grant number 261824, funded by the Research Council of Norway.

References

1. H.M. Gaspar, in *Marine Design Conference (IMDC'18)* (2018). https://doi.org/10.1201/9780429440533
2. I.A. Fonseca, H.M. Gaspar, in *ECMS 2019 Proceedings edited by Mauro Iacono, Francesco Palmieri, Marco Gribaudo, Massimo Ficco* (ECMS, 2019). https://doi.org/10.7148/2019-0023
3. H.M. Gaspar, in *16th Conference on Computer and IT Applications in the Maritime Industries* (2017), pp. 253–269
4. Three.js. Available at https://threejs.org/. Retrieved May 4, 2020
5. S. Loisel. Numeric.js. Available at https://github.com/sloisel/numeric. Retrieved May 4, 2020
6. G. Wagner, in *2016 Winter Simulation Conference (WSC)* (IEEE, 2016). https://doi.org/10.1109/wsc.2016.7822086

7. D.G. Watson, *Practical Ship Design*, vol. 1 (Elsevier, Amsterdam, 2002)
8. D. Andrews, R. Pawling, *Proceedings of RINA Warship 2007: The Affordable Warship* (Bath 2007)
9. Í.A. Fonseca, H.M. Gaspar, C.F. Ryan, G.A. Thomas, in *17th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'18)* (2018), pp. 412–425
10. Í.A. Fonseca, F.F. de Oliveira, H.M. Gaspar, in *Proceedings of the 2019 38th International Conference on Offshore Mechanics and Arctic Engineering, OMAE* (American Society of Mechanical Engineers, 2019). https://doi.org/10.1115/omae2019-96051
11. B. He, Y. Wang, W. Song, W. Tang, Proc.e Inst. Mech. Eng. **229**(12), 2284 (2014). https://doi.org/10.1177/0954405414551106
12. Í.A. Fonseca, H.M. Gaspar, ECMS 171–177 (2015). https://doi.org/10.7148/2015-0171
13. J.J. Jensen, A.E. Mansour, A.S. Olsen, Ocean Eng. **31**(1), 61 (2004). https://doi.org/10.1016/S0029-8018(03)00108-2
14. T.I. Fossen, O.E. Fjellstad, Math Model. Syst. **1**(1), 1 (1995)
15. F.F. de Oliveira, *Implementation of open source code for 6 degrees of freedom simulations in maritime applications* (Tech. rep, Ship Design and Operation Lab, 2019)
16. B. Andrade, H. Brinati, O. Augusto, M. Conti, *Applied Topics in Marine Hydrodynamics* (Escola Politécnica da Universidade de São Paulo, 2016), chap. 7, pp. 262–297
17. WAMIT Inc. WAMIT user manual. Version 7.2
18. J.N. Newman, *Marine Hydrodynamics* (MIT press, London, 1977)
19. J.N. Newman, in *Proceeding of the Conference, Eleventh Australasian Fluid Mechanics* (1992)
20. H.W. Lee, M.I. Roh, Ocean Eng. **167**, 65 (2018). https://doi.org/10.1016/j.oceaneng.2018.08.022
21. S. Chakrabarti, Ocean Eng. **27**(10), 1037 (2000). https://doi.org/10.1016/S0029-8018(99)00034-7
22. D.P. Vieira, P.C. de Mello, R. Dotta, K. Nishimoto, Appl. Ocean Res. **74**, 28 (2018). https://doi.org/10.1016/j.apor.2018.02.019
23. H. Ormberg, K. Larsen, Appl. Ocean Res. **20**(1–2), 55 (1998). https://doi.org/10.1016/s0141-1187(98)00012-1
24. A.H. Peyrot, A.M. Goulois, J. Struct. Division **104**(5), 763 (1978)
25. H.M. Irvine, *Cable Structures* (Dover Publications, New York, 1992)
26. K. Nishimoto, C.H. Fucatu, I.Q. Masetti, J. Offshore Mech. Arctic Eng. **124**(4), 203 (2002). https://doi.org/10.1115/1.1513176
27. B. Molin, J. Fluid Mech. **430**, 27 (2001). https://doi.org/10.1017/S0022112000002871
28. J.N. Newman, *Application of generalized modes for the simulation of free surface patches in multiple body interactions* (Tech. rep, WAMIT Consortium report, 2003)
29. W.H. Pauw, R.H. Huijsmans, A. Voogt, in *International Conference on Offshore Mechanics and Arctic Engineering*, vol. 42703, pp. 597–603 (2007)
30. Z.Q. Hu, S.Y. Wang, G. Chen, S.H. Chai, Y.T. Jin, Int. J. Naval Archit. Ocean Eng. **9**(1), 114 (2017). https://doi.org/10.1016/j.ijnaoe.2016.09.007
31. DNV-GL, SESAM - Software suite for hydrodynamic and structural analysis of ships and offshore structures. Tech. rep., DNV GL AS (2019)

Paper V

Ícaro A. Fonseca, Felipe F. de Oliveira and Henrique M. Gaspar (June 2019). 'Virtual Prototyping and Simulation of Multibody Marine Operations Using Web-Based Technologies'. In: *Proceedings of the 38th International Conference on Ocean, Offshore & Arctic Engineering, OMAE*. Glasgow, UK: American Society of Mechanical Engineers. DOI: 10.1115/omae2019-96051

**Proceedings of the ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering OMAE2019
June 9-14, 2019, Glasgow, Scotland, UK**

OMAE2019-96051

VIRTUAL PROTOTYPING AND SIMULATION OF MULTIBODY MARINE OPERATIONS USING WEB-BASED TECHNOLOGIES

Ícaro A. Fonseca,[*] Felipe F. de Oliveira, Henrique M. Gaspar
Department of Ocean Operations and Civil Engineering
Norwegian University of Science and Technology
Ålesund, Norway

ABSTRACT

This paper focuses on virtual prototyping and simulation of marine operations based on web technologies. The ship is represented as a digital object, which can be used to perform different types of analyses and simulations. The presented simulations are: motion of a single hull and of multiple hulls in regular waves calculated with closed-form expressions, induced pendulum motion response to a lifted load, and motion of a barge with initial movements in still water calculated with equations of motion.

The simulations are developed as web applications in JavaScript and HTML, with graphical user interfaces and 3D renders of the operations. Relevant parameters of the simulations such as wave characteristics and design dimensions are linked to interactive dashboards, allowing the user to modify them and visualize the results in real-time. The applications are lightweight enough to be executed locally in the web browser of most modern devices.

The work employs an open source approach, relying most notably on the Vessel.js library. This aims to foster reuse of models and collaboration with external contributors.

NOMENCLATURE

t Time.
j Vessel motion mode (from 1 to 6 for surge, sway, heave, roll, pitch and yaw respectively).
$\eta_j(t)$ Vessel motion.
ϕ_j Amplitude of the vessel motion.
ω Angular wave frequency.
k Wave number.
$dist$ Orthogonal distance between the vessel's position and the origin plane of the regular wave train.
θ_j Phase angle of the motion mode.
J Transformation matrix from body-fixed to world coordinate system.
F External forces, including gravitational force.
$C_{RB}(\dot{\eta})$ Rigid body Coriolis and centripetal matrix.
$C_A(\dot{\eta})$ Hydrodynamic added Coriolis and centripetal Matrix.
B Damping matrix.
$C(\eta)$ Restoring forces.
M_{RB} Rigid body matrix of inertia.
M_A Added mass matrix of inertia.

INTRODUCTION

Simulations and virtual prototyping (VP) have been important in marine engineering design for years, and the overall usage of simulations and VP through the marine life cycle has been recently increasing. Virtual prototypes allow testing of engineering systems for different purposes in the life cycle, for instance: evaluation of proposals during conceptual design, virtual commissioning of the system and planning of operations. These features are quite desirable in the context of the marine operations, where the high risk, complexity and cost of the systems is prohibitive to the usage of physical prototypes in general.

Virtual prototyping also poses the advantage of allowing sharing of models among distributed agents for usage, verification and validation. Given the high number of stakeholders involved in the vessel's life cycle, it becomes important to share data among distributed agents as efficiently as possible, allowing them to easily access the data that is relevant to their activities.

[*]Contact author: icaro.a.fonseca@ntnu.no

In this context, the web-based approach brings useful features to make vessel data accessible to a great number of users while reducing complications usually associated with the management of digital engineering tools.

For instance, web-applications are compatible with any modern device that has a web browser, avoiding compatibility issues from multiple sources. This ubiquity was attainable in great part due to the reliance of web technologies in open standards, allowing developers to freely use and implement such standards in the development of web-based applications. In fact, two of the three core technologies of the web, HTML and CSS, are open standards, while the third one, the JavaScript programming language, is an implementation of an open standard, the ECMAScript.

In practice, this implies that it is not necessary for the developer to target a specific operational system or device configuration. On the other end, the user is not required to install any software or environment in order to execute the application, and they always have access to the latest version of the app without being required to install updates.

When applied to simulate marine operations, the web-based approach allows the creation of interactive visualizations with realistic 3D graphics including textures and lighting. The applications can be useful in different stages of the life cycle: during early design phase, they may give the user a better perception of the physical meaning of the results; during operation, they may be used for training of personnel or planning of activities.

WEB-BASED VIRTUAL PROTOTYPING AND SIMULATION OF MARINE OPERATIONS

Web-based development is supported by a wide variety of open source libraries for different purposes: they can be applied not only for solving mathematical models such as differential equations, but also for creating elaborated graphical user interfaces, 2D and 3D visualizations, and so on. Gaspar [1] gives an overview of JavaScript development in the context of maritime design and engineering, listing some useful open source libraries.

WebGL is one of the most relevant JavaScript APIs for rendering graphics in a web browser. It supports GPU acceleration for physics and image processing. The Three.js library can be used to draw and load 3D shapes in a canvas using WebGL, making it easier to create animations with lights, textures and other graphical features. All the simulators described in the following paragraphs use visualizations created with Three.js.

There are already some web-applications related to the scope of this work. In terms of virtual prototyping, the CAD platform CAESES released a generator of Wageningen B-series propeller geometries [2]. The user is allowed to configure all relevant propeller characteristics (e.g., diameter, expanded area, pitch, thickness) and the propeller geometry is automatically created in a remote server running the CAD environment. When satisfied, the user can download the final model as a file in STEP or STL format for posterior use. STL is suitable for 3D printing (in fact, the format's name is an abbreviation of "stereolithograph"). A STL model is defined with triangular facets forming a 3D shape. STEP is a CAD format which can be used for engineering analyses. It is an open standard for CAD model exchange developed by ISO, being supported by various engineering software.

Hatledal et al. [3] present an architecture for simulations based on web technologies and the Functional Mock-up Interface (FMI). FMI is an open standard for dynamic simulation models. It is widely used in the automotive industry, but can be applied to other domains as well. FMI allows development of modules that can be exchanged and assembled into complex simulations. It is adequate for distributed co-simulations, where multiple geographically disperse users interact with different aspects of the operation in the same simulation environment simultaneously. The architecture presented in the work executes the simulation modules in the server and synchronizes the results with the client browser, where the visualization layer renders the graphics. The architecture was applied to virtual prototyping and operation of maritime cranes.

The research group with which this work is involved has been consistently developing web applications for marine design and engineering, including some simulations. Chaves and Gaspar [4] presented a 3D simulator for ship virtual prototype and motion prediction in regular waves. It allows configuration of design characteristics for visualization purposes (i.e., propulsion type, bow shape, size of superstructure) and variation of vessel main dimensions, which directly influence the predicted motion response.

VESSEL.JS FOR SIMULATIONS

Vessel.js is a JavaScript library for investigation of common issues in conceptual ship design currently developed by the Ship Design and Operation Lab at NTNU in Ålesund [5]. The library follows a web-based and object-oriented approach. It is open source and collaborative, welcoming reuse of code and input from external contributors.

Vessel.js supports the simulations presented in the following sections, from virtual prototyping of a vessel to simulation of vessel behavior in operation. The simulations are based on a taxonomy comprising three sub-models: entities, states and processes [6, 7]. The entity model collects data about the simulated system. It may represent an actual vessel or a design concept during the design stage. The state model defines static constraints to which the vessel is subjected. It is a static simulation or analysis, e.g., calculation of floating condition or resistance for a given speed. Finally, the process model is a succession of states, which may be arranged to create a dynamic simulation, e.g., a simulation of an operation. In the Vessel.js library, the entity model translates to a ship object, possibly complemented by other objects representing additional systems, the state model translates to modules that receive the ship object and other arguments to calculate the states, and the process model to simulation scripts where the states are combined to simulate the ship behavior.

The next section explains how the ship virtual prototype is defined with Vessel.js, and the following one explains how the library calculates states based on the ship definition and on the simulation constraints. These principles are used to perform the time-domain simulations presented later in this work.

VIRTUAL PROTOTYPING WITH VESSEL.JS

A ship design is described with objects for compartments, structure and additional systems. The compartments are created with "base" and "derived" objects. The base objects define weight data, dimensions and link to 3D files. A given base object functions as a "template" of a compartment, which can be replicated in different positions inside the vessel. This is done with the derived objects, which contain the coordinates where the element will be placed inside the vessel. The ship's structure comprises hull, decks and bulkheads. The decks and bulkheads are defined with geometric dimensions, the spans and equivalent thickness, and material density. The hull is defined with a table of offsets. The weights of the decks and bulkheads are derived directly from the physical dimensions of the elements, while the hull weight is estimated with empirical formulas in order to overcome the lack of structural detailing during conceptual design stage. Finally, additional subsystems (e.g., propulsion, lifting equipment) are modeled in the library with specific approaches depending on the intended purpose of the model and the requirements of the simulation.

Once a ship object is defined with Vessel.js, it can be visualized in WebGL. A function was specifically developed to create a 3D visualization in Three.js from the ship object. The function automatically generates the hull visualization from its table of offsets. The base objects are represented with STL files provided by the user. If no file is provided for a given base object, it will be represented in the visualization with a cuboid of equivalent dimensions. The function returns an object ready to be loaded to a scene in the web browser, where the user can visualize it as pictured in Fig. 1.

A ship object created with Vessel.js can be serialized as a specification and stored for posterior use in various applications developed with the library. Vessel.js uses the JavaScript Object Notation (JSON) as the standard for serialization. Besides being ubiquitous across programming languages and libraries, JSON is also human-readable. This is a crucial feature to allow semantic interpretation of data, facilitating inspection and modification of the specifications.

CALCULATION OF STATES

Vessel.js provides methods to calculate various types of ship states, which can be used to perform a design analysis or to assemble a dynamic simulation. The handling of states follows a certain degree of modularization, being calculated independently from each other when possible.

The Vessel.js library includes an object prototype to handle all the states calculated during a simulation. The object is able to handle both discrete states which do not need to be constantly updated, e.g., the filling ratios of tanks; and continuous states which are constantly reevaluated during the simulation, most notably the vessel's position in the six degrees of freedom. The positional states can be directly linked to the Three.js scene in order to visualize the vessel's motion. The following paragraphs detail the simulation models used in the simulations presented in this work.

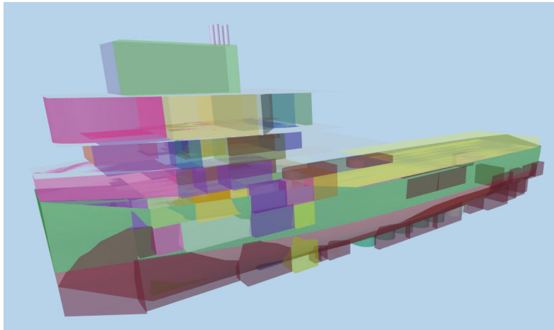


FIGURE 1. VISUALIZATION OF A PSV SPECIFICATION GENERATED WITH THE VESSEL.JS LIBRARY. THE WEB INTERFACE ALLOWS THE USER TO INTERACT WITH THE VISUALIZATION USING MOUSE OR TOUCH COMMANDS.

Loading Condition

The vessel's loading condition can be defined by assigning filling ratios and positions to its derived objects, which are intended to represent its tanks and compartments. It is possible to define the filling ratio of each tank individually or in groups (e.g., group of ballast tanks, group of fuel tanks). When the user requests the library to calculate the vessel's displacement and center of gravity, the library combines the vessel's lightweight with the current loading condition to assess the resulting values.

Floating Condition

The vessel's floating condition is defined by confronting the vessel's current displacement and center of gravity with the hull table of offsets to calculate its floating dimensions, hydrostatic and stability coefficients numerically. This includes calculation of draft, water plane dimensions and coefficients, form coefficients and position of metacenters, among others. Trim is also calculated for small angles (that is, inclining angles small enough for the metacenter position to remain approximately the same). The scheme in Fig. 2 illustrates how a ship can be associated to states describing different loading conditions, which lead to different floating conditions.

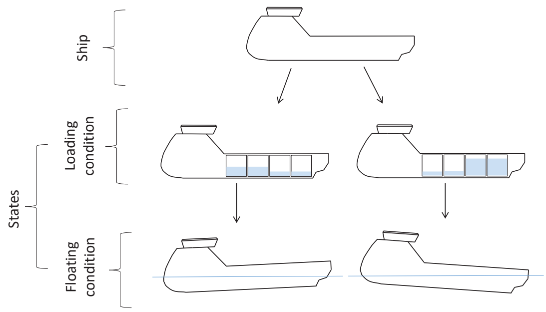


FIGURE 2. SHIP SUBJECTED TO DIFFERENT LOADING STATES, LEADING TO DIFFERENT FLOATING CONDITIONS.

Wave Motion Response Amplitude with Closed-Form Expressions

The amplitude of wave motion response is estimated with closed-form expressions by Jensen et al. [8]. The method was developed motivated by simplicity and suitability for use at early stages of design. It estimates amplitude response for heave, roll and pitch in regular waves based on the hull's main dimensions and its form parameters. The hull is modeled as a box-shaped barge, for heave and pitch, and as a combination of two box shapes, for roll. It neglects coupling between heave and pitch so that the total vertical motion amplitude is estimated by assuming a 90° phase difference between both movements. The authors of the method validated the formulas by comparing its estimates with results from model tests and strip theory calculations for different types of vessel. In general, the formulas were found to predict the motions and accelerations fairly accurately, with some exceptions identified for each motion mode.

In the Vessel.js library, the regular wave characteristics are handled by an object with angular frequency, amplitude and direction in relation to the environment. The ship state should also includes the ship direction in relation to the environment. When the response amplitude is calculated, the wave and ship directions are compared in order to derive the ship heading in relation to waves. The scheme in Fig.3 below illustrates how the wave motion response is calculated based on a given floating condition excited by an incident wave.

Time-Domain Response of Hull with Closed-Form Expressions

The response amplitudes for heave, pitch and roll calculated with the formulation in the previous section can be converted to sinusoidal series with Eqn. (1) and then synchronized with an incident regular wave in a 3D visualization in order to represent hull motion over time. If the vessel is not positioned in the wave origin, the motion phase may need to be corrected for the orthogonal distance in relation to the wave train's origin in order to keep

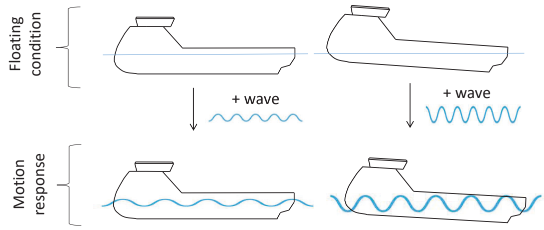


FIGURE 3. MOTION RESPONSE STATES CALCULATED FROM A THE SHIP ON A GIVEN FLOATING CONDITION EXCITED BY AN INCIDENT WAVE.

the hull and wave motions in synchronization.

$$\eta_j(t) = \phi_j cos(\omega \cdot t - k \cdot dist + \theta_j) \qquad j = 3,4,5. \qquad (1)$$

Pendulum Response of Load During Lifting Operation

A module was created to simulate the motion of a lifted load, where the load hangs from an A-frame and has a pendulum motion induced by the vessel's response to incident regular waves. The mathematical formulation models the hanging load as a spherical pendulum with a moving pivot. The equations of motion are derived from the Lagrangian formulation describing the pendulum motion with Euler angles [9, 10], a system of coordinates that can be easily represented in a Three.js visualization.

The accelerations of the pivot, i.e., the load's hanging point, are derived from the motion response calculated with the closed-form expressions, as presented in the previous section. The motion on the hanging point of the load is calculated and substituted on the pendulum equations for each time step of the simulation. The system of equations is solved with the Dormand-Prince method from the Runge-Kutta family of solvers (RKDP), implemented in the Numeric.js library [11], yielding the angular position and velocity of the pendulum over time. As the RKDP method uses adaptive time step, no standard time step duration is specified and the equations are solved in synchronization with the refresh rate of the 3D visualization. The scheme in Fig. 4 illustrates the calculation approach, where the ship motion is combined with the lifting equipment to derive the pendulum response of the lifted load. The pendulum model is purely kinematic, not taking into consideration the forces induced by the load or the motion interaction between load and ship.

Time-Domain Response of Hull with Equations of Motion

Fossen and Fjellstad [12] derived the following equation of motion for the ship by applying Newton's second law to its six degrees of freedom:

$$\ddot{\eta} = \frac{[J^{-1} \cdot F - (C_{RB}(\dot{\eta}) + C_A(\dot{\eta})) \cdot \dot{\eta} - B \cdot \dot{\eta} - C(\eta)]}{M_{RB} + M_A} \qquad (2)$$

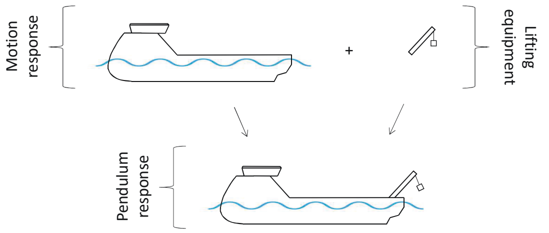


FIGURE 4. SHIP MOTION RESPONSE INDUCING A PENDULUM MOTION TO A LOAD LIFTED FROM ITS A-FRAME.

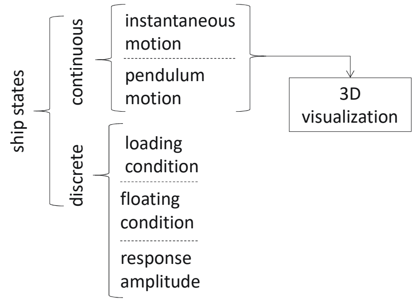


FIGURE 5. SCHEME OF THE VESSEL.JS SHIP STATE OBJECT.

Assuming that the initial states $\dot{\eta}$ and η are known, it is possible to solve Eqn. (2) to calculate the acceleration of the rigid body. The force F represents the sum of external forces applied to the rigid body. This work simplifies the equation by considering the hull floating freely on still water, so the only external force acting on the body is the gravity. However, the formulation can be adapted to account for waves, current or mooring forces, thus being suitable for a wide range of marine operations.

Furthermore, to allow a simple use of Eqn. (2), we choose to focus on the case of a barge with small movement responses. For such case, it is possible to estimate the added-mass and restoring coefficients with closed-form expressions found in the literature, particularly the ones presented by Bergdahl [13]. Oliveira [14] details the formulas used for each coefficient. On the other hand, the damping coefficients are highly dependent on non-linear viscous effects, and thus are not easily calculated. For that reason, the developed applications give the user the option to configure them as inputs.

The system of equations is solved with the same RKDP method as in the previous case, which allows calculation of the position and velocity components of the rigid body for each time step, thus simulating the ship motion over time. Part of the code for the equations of motion was adapted from an open source application previously developed by Monteiro et al. [15].

State Handling in Simulations

The states simulated with the Vessel.js library are stored in a ship state object. Fig. 5 illustrates the two categories in which the object organizes the states: discrete and continuous. Discrete states are assumed to remain constant for longer intervals during the simulation, such as the loading condition, floating condition and the response amplitude calculated with the closed-form expressions for regular waves. They are stored in groups and are marked with cache systems in order to identify when the stored results need to be recalculated. The continuous states experience continuous variation during the simulation, and thus need to be stored and modified constantly, such as the instantaneous positions of the ship and of the lifted load. They are directly linked to the 3D scene and are updated at the visualization's frame rate.

TIME-DOMAIN SIMULATIONS

Dynamic simulations are performed by combining the states as described in the previous section to simulate vessel behavior over time. The simulations presented in the next section grow in scope from motion response of one hull to response of several vessels and accompanying subsystems. Continuous states are calculated in synchronization with the visualization. Discrete states are calculated at the beginning of the simulation and are only updated in case there is a significant change which requires this, e.g., recalculation of stability coefficients due to rearrangement of weights inside the vessel.

The web applications use the Vessel.js features to allow virtual prototyping of vessel and subsystems behavior. Relevant parameters of the simulations such as wave characteristics and design dimensions are linked to interactive dashboards, allowing interaction of the user with the simulations to evaluate performance of different design proposals under different sea conditions. Every time the user modifies a simulation parameter such as wave period, wave height, vessel main dimensions or lifting equipment dimensions, the application recalculates the results and updates the visualization accordingly. The mathematical models are lightweight enough to allow the web browser to execute all operations locally in real-time.

Single Hull Motion Response

The first simulation assesses motion response of a single hull subjected to regular waves, as shown in Fig. 6. By default, the simulation loads with a PSV model. The main dimensions of the model (length, beam and draft) can be scaled by the user, and the simulation automatically updates with the results for the scaled ship. When one dimension is modified by the user, the entire design is scaled, which includes recalculation of tank capacities, structural weight and weight distribution, which in turn influences the floating condition. The user can also configure the amplitude, period and direction of the incident wave. The wave length is automatically adjusted based on the dispersion relation for deep waters considering the chosen period.

The flowchart in Fig. 7 shows the main components of this simulator grouped in three categories: input, calculation (pro-

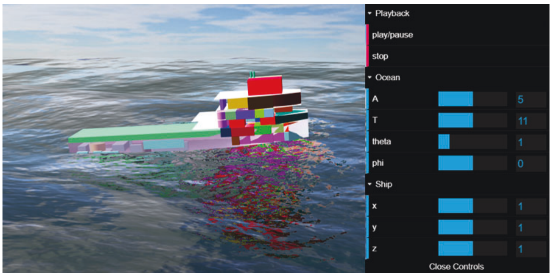


FIGURE 6. SCREENSHOT OF THE SINGLE HULL MOTION RESPONSE SIMULATOR. SLIDERS X, Y AND Z ALLOW THE USER TO SCALE THE SHIP LENGTH, BEAM AND DEPTH, RESPECTIVELY. GUI ZOOMED FOR READABILITY.

cess) and output. The following paragraphs explain each component following the numbering convention in the figure:

0. GUI: a graphical user interface with simple sliders allows the user to control the ship main dimensions and the wave parameters in the simulation (items 1.2 and 1.3, respectively).
1. Input: the 3D files and ship specification define the ship object and 3D model. The ship dimensions and wave parameters are simulation inputs that can be modified while the application is being executed.
   1.1. Ship specification (.json): a JSON ship specification as previously described in the section Virtual Prototyping with Vessel.js.
   1.2. Ship dimensions: the user can scale the main dimensions (length, beam and depth) of the ship specification. This redefines the hull, structure and compartments by proportionally applying the scaling coefficient. Once the user modifies a scaling coefficient, the scaled 3D model is automatically displayed with the recalculated motion response.
   1.3. Wave parameters: the user can modify the wave period, amplitude and direction angle. The wave length is defined based on the period, by applying the dispersion relation for deep waters. The wave geometry is automatically adjusted in the visualization as the user varies its defining parameters.
   1.4. 3D files (.stl): it is possible to display stored STL files in the 3D model of the vessel. They need to be referred in the ship specification in order to be included.
2. Calculation: calculation is handled with objects encapsulating relevant parts of the simulation which can be reused in other applications.
   2.1. Ship object: the Vessel.js ship object, created with the JSON specification.
   2.2. Motion state calculation object: a module containing the closed-form expressions for estimation of ship motion response amplitude.
   2.3. Ocean rendering library: an open source Three.js water shader library [16] used to render an ocean with a single regular wave.
   2.4. Ship 3D model: a Three.js ship 3D model generated from the JSON specification and the 3D files, as shown in Fig. 1.
3. Output: the output is the rendered scene, which can be decomposed in two main components that are reproduced in synchronization.
   3.1. Ship motion visualization: the ship motion is visualized by moving the ship 3D model in the scene according to a sinusoidal function with the motion response amplitude (Eqn. (1)).
   3.2. Wave visualization: the wave is rendered with the water shader library according to the parameters defined in the GUI sliders.

While the motion visualization in this simulation is similar to a previous work [4], there are important differences between

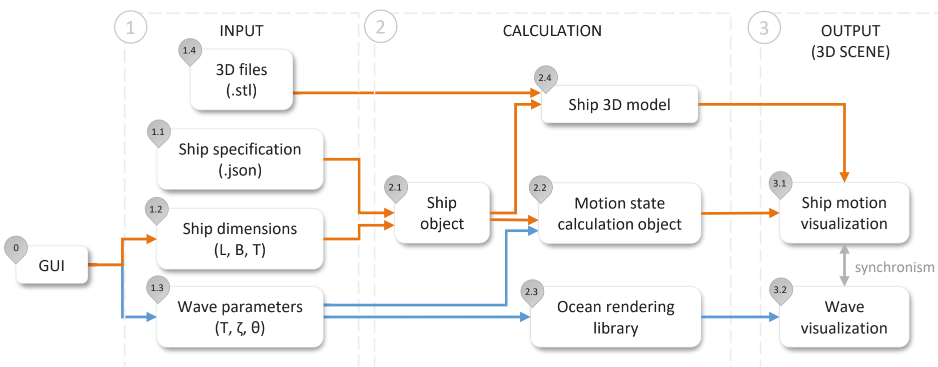


FIGURE 7. FLOWCHART OF THE SINGLE HULL MOTION SIMULATOR. INSPIRED BY CHAVES AND GASPAR [4].

the approaches of both applications. The previous simulator performed all the calculations based on the minimum set of design characteristics required to estimate the motion response with the closed-form model and rendered the visualization with a simplified 3D model of the vessel. The new simulator uses a ship design defined with the Vessel.js library and estimates the motion response based on the characteristics derived for a certain state of that design. While the first version of the simulator is a 3D visualization of the wave motion response in isolation, the new one works as an extension of the Vessel.js library, providing the same 3D motion visualization for a design defined by the user.

Multibody Motion Response

The second simulation calculates the motion responses of multiple hulls simultaneously subjected to regular waves, as shown in Fig. 8. It is very similar to the previous simulation, but adapted to handle the motion response of several hulls.

The flowchart in Fig. 9 illustrates how this is done with an object-oriented approach. The new flowchart is similar to the previous one (Fig. 7), but with the components related to the ship motion reproduced to account for multiple hulls floating simultaneously. The GUI allows the user to vary the number of hulls in the simulation. The flowchart schematizes a simulation with two hulls, but the same structure can be expanded to include more ship instances.

These ship instances are encapsulated and handled independently, which is a suitable approach for the evaluation of multiple motion response amplitudes with different calculation parameters. This way, the main script can perform the required calculations for each ship by invoking a method in the corresponding object, then access the results to move the corresponding ship 3D model in the visualization. Given the mathematical model previously described, the simulation does not consider the effects of wave interaction due to the presence of multiple vessels.

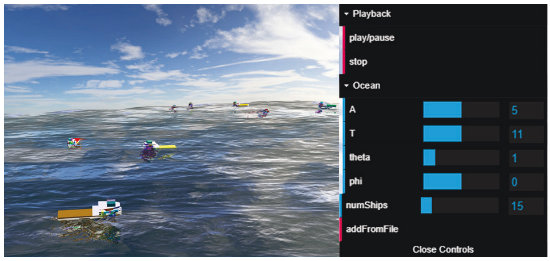


FIGURE 8. SCREENSHOT OF THE MULTIBODY MOTION RESPONSE SIMULATOR.

Pendulum Motion of Lifted Load

The pendulum application is similar to the Single Hull Motion Response simulator, but with the addition of an A-frame with a hanging load, as shown in Fig. 10. The pendulum motion responds in real-time to the ship motion, which in turn is influenced by the wave parameters set by the user (i.e., wave amplitude, period and direction).

The organization of the simulator is very similar to the flowchart in Fig. 7, but adapted to include a geometric definition of the A-frame, contained in an object, and a 3D model of the A-frame generated automatically from that definition. Furthermore, a new module is also necessary to calculate the pendulum motion induced by the hull response to waves.

Motion of Free Floating Hull

Differently from the previous examples, this simulation does not evaluate hull motion with predefined equations based on ex-

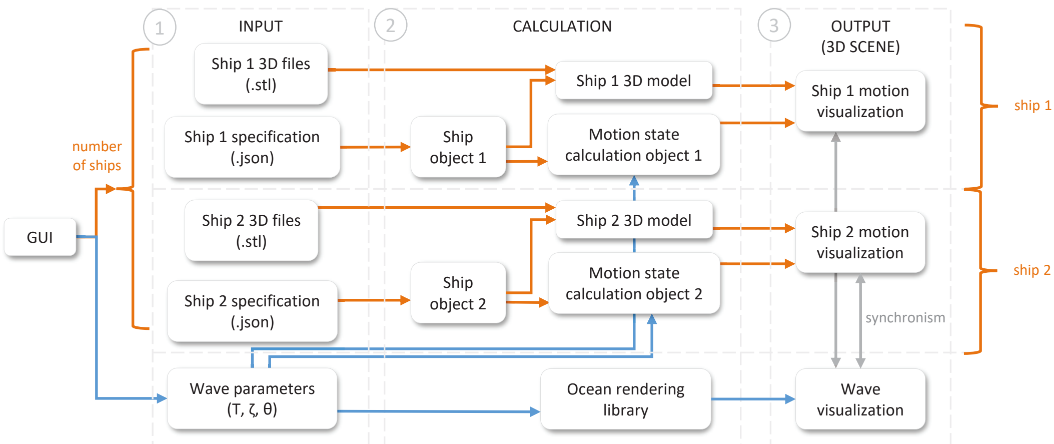


FIGURE 9. FLOWCHART OF THE MULTIBODY SIMULATOR.

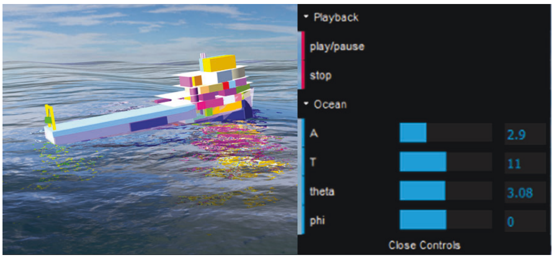


FIGURE 10. SCREENSHOT OF THE PENDULUM MOTION SIMULATOR.

perimental methods. Instead, it uses Eqn. (2) to calculate hull position over time. Fig. 11 shows a barge with initial heave and roll conditions, set by the user to be different from zero. As the simulation advances, the barge will oscillate until virtually all energy dissipates due to damping. Despite the fact that the equation accounts for the motion's six degrees of freedom, the web-interface only allows the user to set initial conditions in the modes with restoring components, i.e., heave, roll and pitch, in order for the hull oscillation to be observed. The box-shaped barge geometry was chosen due to the simplicity of its motion coefficients, particularly the added mass coefficients. However, in the future it is possible to use more complex formulations in order to represent added masses for distinct geometry types.

The flowchart in Fig. 12 shows the components of the simulator. Note that this simulation does not use any parameter to configure the ocean, because it is always considered to be in the calm water condition. It is worthwhile to have a deeper look into two items from the chart inputs, because they differ from the other simulations:

1.3. Initial state: the user can change the vessel's heave, roll, and pitch in order to simulate its movement trough time. The initial state will be changed in the ship state object, which is translated to the movement of the ship 3D model in the visualization.

1.4. Damping coefficients: these are the coefficients responsible for the movement decay. C_D is used to calculate the damping in the three linear directions (surge, sway and heave). B_{44} and B_{55} account for the damping in roll and pitch, respectively.

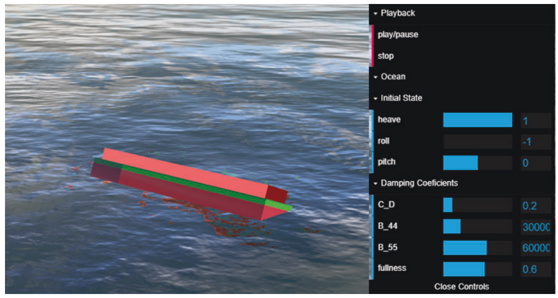


FIGURE 11. SCREENSHOT OF THE FREE FLOATING HULL MOTION SIMULATOR. THE SLIDERS ALLOW THE USER TO ADJUST THE DAMPING COEFFICIENTS AND VARY THE LOADING CONDITION OF THE BARGE.

DISCUSSION

The web applications presented in this work successful performed time-domain simulations of motion with 3D visualizations in real-time on the client-side (that is, purely on the browser, without relinquishing computation to a server). The framework for state handling, which had its development started

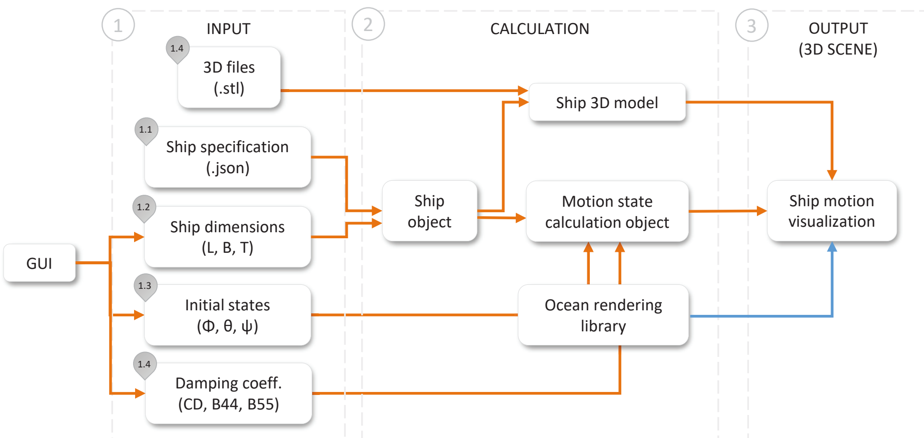


FIGURE 12. FLOWCHART OF THE FREE MOTION SIMULATOR.

in previous works [7], now has its foundations in place, providing capability to handle both discrete and continuous states during a simulation.

At this point, the web applications still present some limitations in scope and accuracy to account for the simulation of an entire marine operation. Heave and pitch responses calculated with the closed-form expressions are exaggerated in some cases, as already acknowledged in the source material [8]. Likewise, the motion simulation based on equations of motion is very incipient, and does not yet account for wave response.

However, the simulations put the potential of the web-based approach to test and serve as a starting point for the forthcoming work. The approach materializes the anticipated benefits in accessibility of simulation models, allowing one to configure them online and provide access to geographically distributed users with minor complications. The simulators demonstrated the potential of web technologies in supporting user interaction, by allowing creation of interfaces and visualizations, and in taking advantage of open source development, by applying various open libraries to engineering problems.

The simulations give the first step towards simulation of motion with differential equations for the Vessel.js library. Given the computational performance of the applications presented, the web-based approach still provides potential to accommodate more demanding mathematical models. In the future, they could be further developed to incorporate strip-theory methods.

CONCLUSION

This work presented a web-based approach to ship virtual prototyping and simulation of marine operations. The approach was applied to the development of web-applications with simulations of motion response of a single hull in regular waves, of multiple hulls in regular waves, of a load lifted from an A-frame and of a hull floating in still-water. The motion response is calculated with closed-form expressions for the hulls in regular waves, while the motion of the hanging load and of the hull in still water are calculated by solving the equations of motion numerically.

The web-based and open source approaches were also beneficial to the development of the applications by allowing interactive visual presentation, by assuring accessibility and compatibility of the simulators among devices, and by enabling the usage of various open source libraries during development, most notably Vessel.js.

FUTURE WORK

At the moment, the development of the Vessel.js library is focused towards simulations of subsea operations and motion interaction between vessel and mooring or towing lines.

Furthermore, the library may also be linked to FMI, which is now being proposed as a standard for exchange of simulation models in the maritime industry [17]. This would provide the benefits of the web-based approach to the FMI simulations, while allowing organization of the functional mock-up units in a more comprehensive framework supported by Vessel.js.

SOURCE CODE

The web page of the Vessel.js library can be accessed on the following address: https://vesseljs.org/. Besides the source code of all the examples developed for this paper, it includes documentation, other examples of applications and tutorials. The library and web page are currently under active development and should still undergo improvements after the publication of this work. As the project aims to be collaborative, anyone is welcome to use and contribute to the project.

ACKNOWLEDGMENT

This research is connected to the Ship Design and Operation Lab at NTNU in Ålesund. The research is partly supported by the EDIS project, in cooperation with Ulstein International AS (Norway) and the Research Council of Norway, and by the INTPART Subsea project in cooperation with the University of São Paulo (USP) and the Research Council of Norway.

REFERENCES

[1] Gaspar, H. M., 2017. "JavaScript applied to maritime design and engineering". In 16th Conference on Computer and IT Applications in the Maritime Industries, pp. 253–269.

[2] Harries, S., Lorentz, K., Palluch, J., and Praefke, E., 2018. "Appification of propeller modeling and design via CAESES". In 17th Conference on Computer and IT Applications in the Maritime Industries, pp. 292–307.

[3] Hatledal, L. I., Schaathun, H. G., and Zhang, H., 2015. "A software architecture for simulation and visualisation based on the functional mock-up interface and web technologies". In Proceedings of the 56th Conference on Simulation and Modelling, Linköping University Electronic Press.

[4] Chaves, O., and Gaspar, H., 2016. "A web based real-time 3D simulatorfor ship design virtual prototype and motion prediction". In 15th Conference on Computer and IT Applications in the Maritime Industries.

[5] Gaspar, H. M., 2018. "Vessel.js: an open and collaborative ship design object-oriented library". In Marine Design Conference (IMDC'18).

[6] He, B., Wang, Y., Song, W., and Tang, W., 2015. "Design resource management for virtual prototyping in product collaborative design". *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* ***229***(12), pp. 2284–2300.

[7] Fonseca, Í. A., 2018. "An open and collaborative object-oriented taxonomy for simulation of marine operations". Master's thesis, NTNU.

[8] Jensen, J. J., Mansour, A. E., and Olsen, A. S., 2004. "Es-

timation of ship motions using closed-form expressions". *Ocean Engineering,* ***31***(1), pp. 61–85.
[9] Myhre, T. A. Spherical pendulum dynamics. Available at `https://www.torsteinmyhre.name/snippets/spherical_pendulum.html`.
[10] Myhre, T. A., and Egeland, O., 2016. "Collision detection for visual tracking of crane loads using a particle filter". In IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE.
[11] Loisel, S. Numeric.js. Available at `https://github.com/sloisel/numeric`.
[12] Fossen, T. I., and Fjellstad, O.-E., 1995. "Nonlinear modeling of marine vehicles in 6 degrees of freedom". *Mathematical Modeling of Systems,* ***1***(1), pp. 1–11.
[13] Bergdahl, L., 2009. "Wave loads on and motions of a ship in regular waves". In Wave-induced loads and ship motion, pp. 65–112.
[14] de Oliveira, F. F., 2019. Implementation of open source code for 6 degrees of freedom simulations in maritime applications. Tech. rep., Ship Design and Operation Lab.
[15] Monteiro, T. G., Xu, J., and Gaspar, H. M. Animated linear roll + heave ship model (6dof model). Available at `http://www.shiplab.hials.org/app/6dof/`.
[16] Bouny, J. Ocean - realistic water shader for three.js. Available at `https://github.com/jbouny/ocean`.
[17] Chu, Y., Hatledal, L. I., Æsøy, V., Ehlers, S., and Zhang, H., 2017. "An object-oriented modeling approach to virtual prototyping of marine operation systems based on functional mock-up interface co-simulation". *Journal of Offshore Mechanics and Arctic Engineering,* ***140***(2), nov, p. 021601.