**Proceedings of the ASME 2022 41st International Conference on Ocean, Offshore and Arctic Engineering
OMAE2022
June 5-10, 2022, Hamburg, Germany**

OMAE2022-81505

APPLYING OPEN WEB ARCHITECTURES TOWARDS COLLABORATIVE MARITIME DESIGN AND SIMULATION

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

ABSTRACT

This work investigates the use of open architectures to support the development of flexible and scalable maritime design web applications, giving stakeholders shared access to data. It turns to the popular full-stack MERN architecture (MongoDB, Node.js, Express, and React), which is is modular and mostly open source. A prototype web application providing features for ship design and operation was developed. The app stores a ship model which can be linked to different analyses and simulations. During design, users might opt to visualize the model of the ship with a spatial view; during operation, they can resort to a detailed visualization displaying the vessel as built. Three examples are provided to illustrate the potential of these features. First, a dashboard displaying results for hydrostatics, stability, resistance, and motion response. The second use case hypothesizes a vessel is set to undergo a jumboization procedure and compares the analyses results for the vessel after elongation with the current ones. The third exemplifies how a preliminary maneuvering model can be confronted with results from a sea trial by linking the app to operational data, a step towards digital-twin concepts. The discussion addresses the potential of the approach and challenges that need to be considered before extending it to an application that can be used outside the academia.

1 INTRODUCTION

The sustained advances in technologies for data storage, computing processors, and sensors are leading to significant transformations across industries. One of the central aspects of this transformation is the emergence of digital platforms connecting distributed systems, both virtual and physical, to provide a certain service [1]. In the maritime sector, there is an increasing amount of data generated during the life cycle of systems and a considerable unexplored horizon to use that data more effectively. From one side, these developments bring a great potential toward generating value to stakeholders, e.g., in the form of cyber-physical systems and digital twins. On the other, they require a significant reshaping of existing tools and practices to be exploited successfully.

One of the concepts to achieve this in practice is the digital thread. It describes the use and management of data in a way that effectively supports the life cycle of complex engineering systems [2]. By establishing a single "source of truth" for system data, the digital thread could bring advantages on several fronts. One of them is in organizing and contextualizing data according to the analyzed subsystem or scenario. As the life cycle progresses, the digital thread maintains information about the system, supporting downstream of data to subsequent phases, such as reusing design simulations during system operation. Another advantage is in the coordination of system design and construction. By sharing and updating the digital model of the system, stakeholders will be able to reduce mistakes and rework related to miscommunication and incorrect versioning of production drawings.

In face of the complexity of maritime systems [3], these functionalities can increase traceability of information and, in the long-term, enable iterative learning based on the data stored for a previous tender. In practice, there are significant obstacles to the implementation of that vision. Two of the most important ones are the lack of convergence toward standard formats and the lack of interfaces for data exchange among different tools.

[*]Contact author: henrique.gaspar@ntnu.no

They might be illustrated with two examples from the literature. In 2011, when attempting to accomplish effective data exchange among multiple ship design tools, Whitfield *et al.* first turned toward the ISO 10303 standard, informally known as STEP [4]. However, during the study, they found out the standard was not sufficient to cover the scope of the data to be exchanged and ended up creating a custom ship model instead [5]. More recently, in 2019, El-khoury *et al.* reported on the challenges encountered when attempting to expose and manipulate data in software for automotive engineering [6]. They explain that even when tools provide Application Programming Interfaces (API) for data exchange, they impose limitations, e.g., by not exposing relevant internal logic that is used to manage the data, or by limiting API access level to read-only.

Currently, the lack of established methods for interoperability cause hindrances that range from burdening stakeholders with data rework when exchanging it among tools, at least, to loss of important information, at most [7]. In the future, it might become an impediment to the integration of digital engineering tools in platforms and heterogeneous systems. For such reasons, it is necessary to develop methods capable of supporting the digital thread.

2 IMPLEMENTATION WITH THE VESSEL.JS LIBRARY

This work turns to web technologies as an enabler to that vision. More specifically, it investigates the use of modern open web architectures to aggregate the information and simulations of a maritime system into a digital hub accessible to various stakeholders. By relying on web-driven standards and infrastructure, it should become easier to access and share data. E.g., when a simulation is executed, its results would be stored in a database to which other stakeholders will have access, allowing them to read and download the required information in a convenient format such as JSON. While the architecture aims to centralize engineering information, its modular characteristic should leave open the opportunity for customization with proprietary components and for data exchange with additional tools. E.g., it might enable access to a given 3D model or simulation result through a Universal Resource Locator (URL) or an API.

One of the research projects the authors have been involved in the past few years is the development of the Vessel.js library, which applies an object-oriented approach to vessel design and simulation [8]. Vessel.js is one of the central tools used in this report, being incorporated as a central element to the development of the architecture. Previous research used Vessel.js to model a ship and calculate its states during simulations, for example, mooring systems with six degrees of freedom described with ordinary differential equations, solved in real-time [9]. The library is developed in JavaScript [10], allowing it to be executed on web browsers natively, thus obtaining compatibility benefits with different systems and devices.

While the existing capacity for numerical solving is restricted to closed-form equations of motion, the library might be used as a visualization tool for results obtained with more resource-intensive software. For instance, examples of simulations with a higher level of mathematical complexity such as those involving the hydrodynamic calculation of multiple floating structures, or multibody simulations [11]. This is especially true considering interactions among multiple floating vessels, i.e., the shadow effect. In this case, the evaluation of hydrodynamic coefficients and responses require a high processing capacity, discarding the possibility for real-time calculation on the browser.

To overcome this drawback, the response operators can be calculated beforehand using a commercial boundary element software (such as WAMIT [12]) and saved to a file. Later when simulating the motion on a web browser, the application fetches the response amplitudes and uses them to calculate a visualization of the ship and mooring responses over time. The work discussed the opportunity of further integrating data during design and operation, allowing operators to access and use that data in databases with APIs for data exchange among the computers executing the simulations and then visualizing simulation results on the front end.

In practice, the creation of environments offering this level of integration was hindered by the fact that simulations were developed as standalone applications, without a server-side logic connecting them among each other or to a common database. In the absence of those features, data needed to be exchanged repetitively between different agents, sometimes through a manual process. While this approach is enough for the development of isolated apps or case studies, it precludes fulfillment of a digital thread because it lacks scalable methods to serialize data and exchange it among individual apps.

A web architecture is a fitting tool to implement such kind of integration and compatibility in practice. This work proposes the use of existing web frameworks as an initial step towards achieving this level of data exchange, with the longer-term goal of providing stakeholders collaborative access to a common repository storing data and models about a maritime system.

3 THE MERN ARCHITECTURE

The web architecture relies on the development stack popularly called MERN, comprising the tools MongoDB, Express, React, and Node.js. Those tools are a set of open source components that together provide an end-to-end framework for building dynamic web applications; starting from the top (code running in the browser) to the bottom (database).

Among several advantages, this stack applies the Javascript language and the JSON data notation from the front end (user side) to the back end (server side). From one point of view, this simplifies development for not requiring use of multiple programming languages.

Figure 1 shows MERN as a three-tier architecture, categorizing the most important tools for this project according to the respective domain. Among its advantages, the architecture provides opportunity for upgrades or changes in one of its tiers without requiring modifications to the entire system.

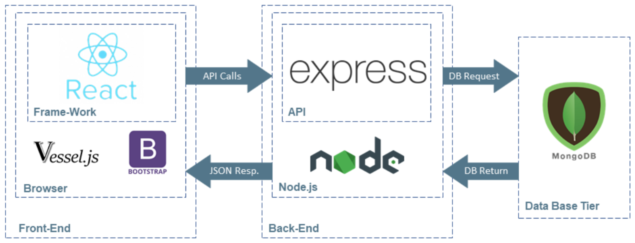


FIGURE 1. Scheme of three-tier MERN web architecture.

Each of the web frameworks used in this work can be explained according to its role in development and its tier in the architecture, for instance:

1. Front-End: the user interface layer. It establishes the logic and graphical interface according to which the user interacts with the web tool.
   - A React: a declarative and flexible JavaScript library for building user interfaces. React modifies the front-end according to the state, rendering just the right components for each change. The building process is encapsulated in components, thus enhancing modularization.
   - B Vessel.js: is a JavaScript library for conceptual ship design with an object-oriented approach. Vessel.js represents the vessel as an object, which is used to simulate different functionalities and behaviors. Currently, the library includes methods for resistance, seakeeping, hydrostatic, and stability calculations.
   - C Bootstrap: open source toolkit for the development of adaptable graphical user interfaces on the front-end, with responsive grid system based on prebuilt interface components.
2. Back-End: server-side layer to handle data storage and programming logic.
   - A Express.js: framework for development of web applications with Node.js. It provides a robust set of features, including methods and middle-ware for creation of robust APIs quickly and easily.
   - B Node.js: a cross-platform back-end runtime environment. It runs JavaScript, the same programming executed in web browsers.
3. Database: a logical organization of collected information.
   - A MongoDB: a document-oriented database that stores data in flexible, JSON-like documents. This allows variation of fields from document to document and changes to the data structure over time.

In addition, the Node Package Manager (NPM) is an important tool for maintaining control over the versions of the packages. NPM includes a command line interface that allows scalable publishing, download and update of packages with Node.js.

4 WEB-CENTERED COLLABORATIVE APPROACH

An open web architecture is expected to foster different kinds of collaboration. Some of these collaboration methods can be correlated with the three architecture tiers depicted in Fig. 1. Figure 2 exemplifies some of the approaches a user or developer can take when interacting with the web platform, according to the tier in which that interaction occurs.

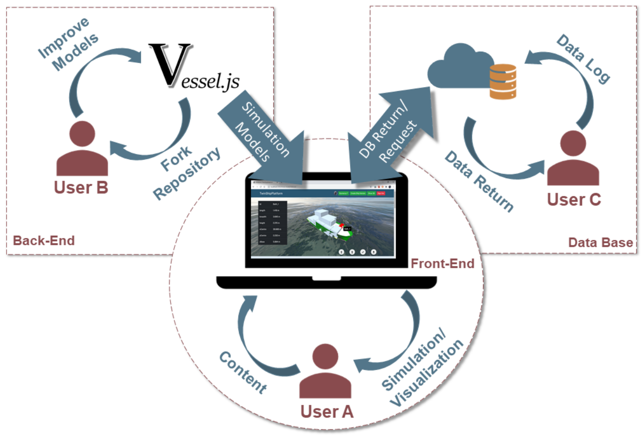


FIGURE 2. Three possible collaboration modes.

One of the ways for collaboration through the platform is via the insertion of new content in the front end by the upload and refinement of new ship models. In this example, User A improves the web platform by inputting a new ship version or refining an existing one. Since the modifications can be saved and stored in the database, an other user would be able to pick-up on the latest version of the vessel model and continue work from there. In addition, a developer could take the initiative to modify the open source code for the front-end user interface, which is publicly hosted in the GitHub repository.

The second form to collaborate with a web platform based on the architecture proposed here is by improving the source code used in the applications. For instance, from the development side or back-end layer, it is possible to update existing open source libraries used in the project, such as Vessel.js, and incorporate new ones. This type of expansion is classified as one of the forms of collaboration the platform enables, as represented by User B in Fig. 2. It can be used to improve the mathematical models used in simulations by updating them with new versions right into the back end layer. As matter of fact, this type of collaboration is not directly aimed at the platform, since in this case it is the library upon which the applications are built that is being improved. Thus, this form of collaboration assumes a more indirect approach.

Finally, the last way to improve the web platform is by exchanging data with the database, such as shown with User C. This data could be acquired through sensors or highly expensive computation results that are impractical to be calculated inside a web environment. While the figure illustrates this process occurring through manipulation by a user, this can be implemented with alternative approaches which offer more robustness and scalability. For instance, it is becoming recurring to use APIs in web services to acquire third party functions and data. In such manner, APIs could be used to reinforce the collaborative aspect of the web platform by accessing resources hosted in external services and, conversely, by providing its own resources for use in third party applications. From the data acquisition perspective, this means that the platform does not need to be restricted to a single data source. Instead, it can be connected to a composition of databases from multiple stakeholders, as long as there are commercial and IP arrangements among them to share their data toward a common purpose.

5 SHIP MODELS FOR DESIGN AND OPERATION

Two ship models were chosen for the case studies to apply the capabilities of the web platform: a PSV design [8] and the research vessel (R/V) Gunnerus with detailed 3D models, provided by NTNU.

The PSV design (Fig. 3) contains 106 objects representing compartments and tanks, being the most extensive one used in the Vessel.js library in terms of quantity of objects.

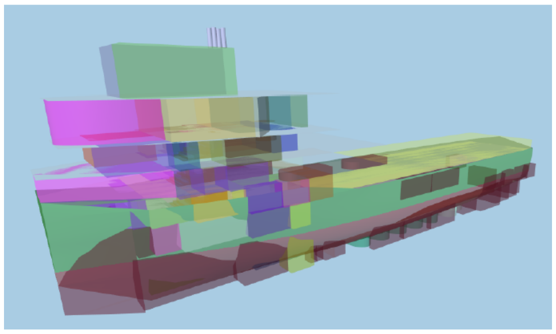


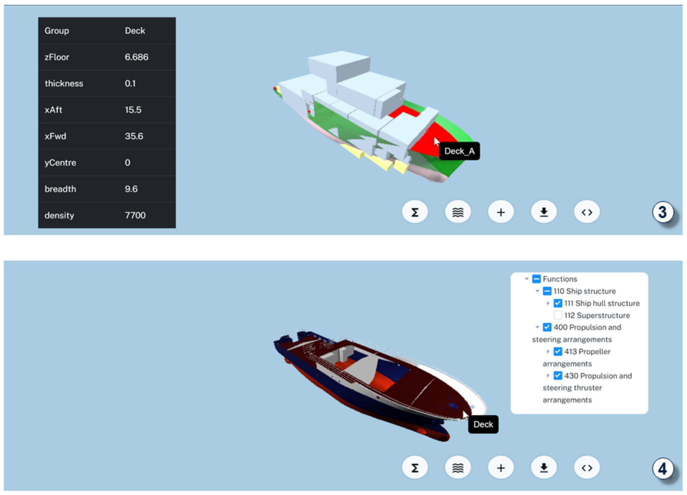
FIGURE 3. Adopted PSV geometry [8].

The second ship, R/V Gunnerus (Fig. 4), is owned by NTNU and doted with sensor infrastructure for remote monitoring of operational states. In practice this brings the possibility to use operational data in simulations, opening doors for future implementation of a digital twin.

The web apps provide a visualization page that allows browsing R/V Gunnerus from different perspectives (taxonomies), as contrasted in Fig. 5. In the preliminary analysis, the ship follows a simplified visualization, closer to the preliminary design methods that aim to outline the vessel spaces. Thus, the top image shows a spatial arrangement, with the definition of system locations. In the detailing phase, the visualization page shows a more accurate ship model, containing detailed shape and colors in a glTF (Graphics Library Transmission Format) file.

FIGURE 4. Detailed three dimensional model of R/V Gunnerus, NTNU.

The vessel elements are organized according to a functional hierarchy [13]. The user is allowed to toggle different vessel functions to show or hide the vessel elements related to it, as observed in the bottom part of the figure.



Group	Deck
zFloor	6.686
thickness	0.1
xAft	15.5
xFwd	35.6
yCentre	0
breadth	9.6
density	7700

- [−] Functions
  - [−] 110 Ship structure
    - [✓] 111 Ship hull structure
    - [ ] 112 Superstructure
  - [✓] 400 Propulsion and steering arrangements
    - [✓] 413 Propeller arrangements
    - [✓] 430 Propulsion and steering thruster arrangements

FIGURE 5. Comparison between the R/V Gunnerus visualizations for initial design (physical taxonomy) and operation (functional taxonomy).

Table 1 summarises the main dimensions for both vessels used in the case studies. Some of the hydrostatic and stability methods in Vessel.js have been previously validated with a simplified barge geometry. However, others, such as the ones for resistance and seakeeping, offer a limited range of validity. This affects the quality of results obtained when applying them to small vessels such as the ones in the case studies, as they do not always attain the indicated ranges for those methods.

From a research perspective, this happened due to a few reasons. In short, previous works by the authors focused on incorporating classical naval architecture formulas [14] and other useful expressions found in the literature [15] into Vessel.js. However, these models are calibrated for large cargo vessels, and not for smaller offshore support vessels or research vessels.

Since this paper's scope is data integration in the maritime

TABLE 1. Summary of ships' main dimensions.

	Gunnerus	PSV
LOA (m)	36.25	82.00
BOA (m)	9.60	18.00
Depth (m)	6.60	8.00
Draft (m)	2.79	6.50
LWL (m)	34.90	80.00
Cb	0.63	0.68

digital thread, simulation results are mostly used as preliminary examples of the approach's potential. Furthermore, the development of accurate maneuvering models validated based on operational data is a valuable research topic by itself [16]. The simulation models could certainly be substituted with adequate alternatives provided by other authors in the future. In fact, the possibility to incorporate code from other projects is exactly one of the advantages of an open source approach.

6 CASE STUDIES

6.1 Design Evaluation Dashboard

One of the web tool's functionalities is to display analysis results with the objective of evaluating a vessel design. This can be exemplified with a qualitative chart comparison based on the conventional Holtrop method for resistance analysis [14]. Figure 6 exposes the resistance as a function of ship velocity for Gunnerus and the PSV, decomposing the total resistance in contributions from different sources. It is possible to notice, for example, that at higher speeds Gunnerus demands more energy due wave generation, while the PSV presents a higher energy demand due to viscous sources. This information could be used in the ship design phase to identify which solutions are better suitable for resistance reduction. The graphical interface provides options to present resistance results in other manners as well, such as a table summary with key resistance parameters, or as a pie chart showing different resistance components for a given speed.

6.2 Assessment of Vessel Modification

During their operational span, vessels might undergo an enlargement procedure called jumboization. In it, the ship is elongated by the introduction of a middle body section, usually with the objective of attending to an increased demand arising after its construction and delivery. R/V Gunnerus recently went through a jumboization procedure that increased its length overall in five meters.

The planning and execution of this sort of modification requires re-calculation of several factors, e.g., the increase in resistance, modification of the motion response curves and variation of hydrostatic factors. This brings a special degree of complexity due to the requirement of managing trade-off between different functional aspects of the ship. This past scenario is taken as motivation to illustrate usage of the web platform as a tool for comparisons between alternative ship versions.

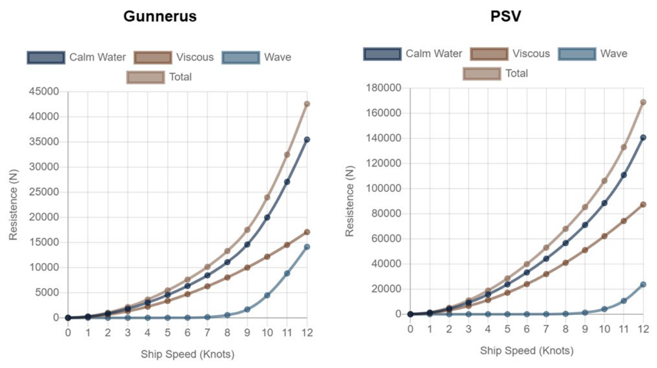
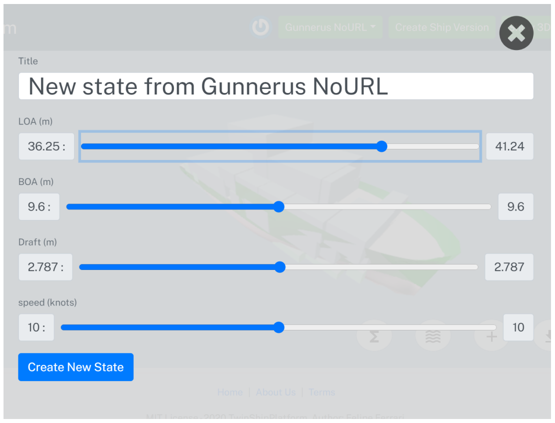


FIGURE 6. Resistance as function of speed for Gunnerus and the PSV.

With this concept in mind, the web platform was designed to give the ability for users to create new scenarios for a specific ship version easily, allowing quick assessment of multiple parameters. It is expected that the chances of version mismatch in such cases are reduced by using the web platform as the central source of information during the process of evaluating different extension lengths. This is especially important since the digital thread aims to become a bridge between several stakeholders who must be aware of the circumstances of a project.

In the web interface, the user is able to press a "create new state" button in the navigation bar, then a set of sliders allowing vessel modification appears (Fig. 7). In the current application version, the creation of new ship version allows only modifications of main dimensions. However, in future versions it could include modification of other variables, such as form coefficients.

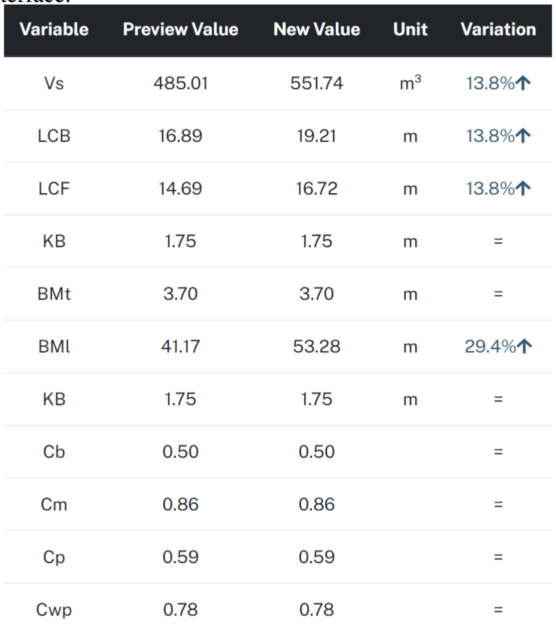


Title
New state from Gunnerus NoURL
LOA (m) 36.25 : 41.24
BOA (m) 9.6 : 9.6
Draft (m) 2.787 : 2.787
speed (knots) 10 : 10

FIGURE 7. Example of a new state page with the jumboization value for the length overall.

The web interface allows users to assess changes in resistance, hydrostatics, and seakeeping parameters. For instance, Table 2 summarizes the comparison of hydrostatic parameters for the ship after an elongation of 5.0 meters, considering both versions, previous and elongated, at the same draft of 2.79 meters.

TABLE 2. Screenshot of a hydrostatic comparison table within the web interface.



Variable	Preview Value	New Value	Unit	Variation
Vs	485.01	551.74	m^3	13.8%↑
LCB	16.89	19.21	m	13.8%↑
LCF	14.69	16.72	m	13.8%↑
KB	1.75	1.75	m	=
BMt	3.70	3.70	m	=
BMl	41.17	53.28	m	29.4%↑
KB	1.75	1.75	m	=
Cb	0.50	0.50		=
Cm	0.86	0.86		=
Cp	0.59	0.59		=
Cwp	0.78	0.78		=

The clearest outcome is the increase in the volume displacement with the increase of the length, meaning that in this new configuration the ship can carry more weight than before without a drastic increase in energy demand. LCB and LCF are changed proportionally to the increment in length. This may affect the weight distribution and consequently the trim. The major difference is the increase in the longitudinal metacentric distance, which in this case represented almost 30% in gains that can be translated into a more stable ship trim angle. In short, a web platform aggregating several methods in a single interface allows quick comparison of vessel alternatives from different performance perspectives.

6.3 Linking to Operational Data

The last case study presents a preliminary attempt to perform and verify maneuvering simulations with the developed tools. The results from the simulation are compared to navigation data from Gunnerus acquired through monitoring systems. From the data exchange perspective, the application presents an example of data usage at the database tier, since the information is incorporated into the database from an external system. From the front-end perspective, the example illustrates how simulation verification can be presented in an intuitive manner to the end-user.

One of the main reasons to use digital twins when confronting the information with the adopted maneuvering models [17] is to verify how well they predict the real behavior. For this purpose, a specific module was developed to read the sensor data from the database, calculate the modeled output at each time step, and stream the results together with the original data. The diagram in Fig. 8 helps to understand the process workflow. It combines fundamental digital twin elements (external data from sensors and internal data from feeedack) to the analyses loop. More about principles of digital twin using the same platform as backbone can be found in Fonseca and Gaspar [18] [19].

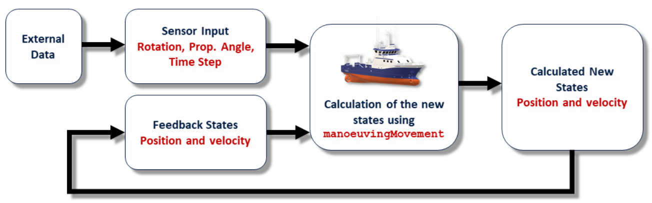
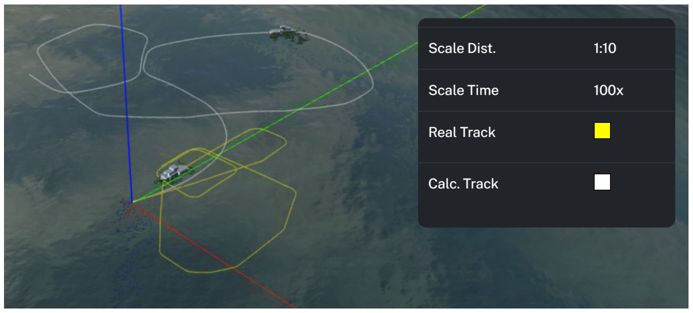


FIGURE 8. Algorithm loop for the twin ship application.

The results from the real Gunnerus data and from the simulation model a linear damping condition can be visualized in Fig. 9. The white track represents the route calculated using the model in Vessel.js, having the real data of the propeller rotation and angle as input; the yellow track is the real data ship route during the sea trial.



Scale Dist.	1:10
Scale Time	100x
Real Track	🟨
Calc. Track	⬜

FIGURE 9. Digital twin ship application with real trajectory in yellow and prediction in white.

As it may be seen in the figure, the chosen model is far from the predicted condition; this happens because of several modeling reasons. One of the modeling restrictions is the absence of the bow thrust effect in the group of forces acting on the ship. Another factor that affects the prediction accuracy is that it does not account for environmental forces such as the wind, wave, and current in the ship's course. The third reason is the hydrodynamic coefficients matrix, which was chosen with a certain arbitrariness in the values and does not faithfully represent real coefficients for the vessel. Despite those restrictions, this application gave a step forward toward applying the Vessel.js library to a digital twin. The modeling problems can be assessed in the future by using more realistic models and hydrodynamic coefficients for the ship.

7 DISCUSSION

This study aimed to investigate the feasibility of using open web architectures as an enabler to the digital thread in maritime. In the context of digital support to life cycle activities, the case studies illustrated how a web platform could aid ship design and operation, focusing specifically on conceptual studies during design, assessment of modifications to an operating vessel and verification of simulations during operation. While these functionalities are relevant, there is also room for using such a tool during different life cycle stages. This includes the notably costly task of coordinating detailing and construction between design office and yard. In fact, this expanded scope is necessary in case any architecture wants to enable a digital thread, since it should by definition extend to all life cycle phases. Thus, the architecture outlines an approach toward establishing a common data source in maritime projects, with the aim of reducing conflicts between model versions and simplifying information sharing among collaborators.

Although the web-based approach was applied here to case studies in ship design and operation, it could also be applied to other maritime or engineering systems. In that sense, the use of such a tool is better justified in cases when the asset being managed is relatively complex. This is due to the higher amount of more components and behaviours to track, allied to the upfront investment in developing or acquiring a novel, full-fledged solution for data management in engineering. This scenario might change in the long term if it possible to find ways to drive these costs down, to the point in which it becomes competitive to have a sophisticated digital solution even for simpler systems.

One of the points where web architecture brings most value compared to traditional data management methods in maritime is in its combination of scalability and customization. In practice this means the architecture is able to accommodate robust methods for data handling and content distribution with the opportunity for different stakeholders to adopt tools from different vendors.

From the robustness perspective, web services are quite mature, being able to support projects of large scope. The accessibility of web technologies to distributed users makes the data and simulations scalable in terms of potential users and sharing, as they can be accessed from whatever part of the globe at any time. For commercial implementations, there is now a significant offer of comprehensive cloud services for subscription, such as Amazon Web Services, Microsoft Azure, and Google Cloud, besides specialized tools from smaller providers.

From the customization side, the adopted frameworks support various functionalities for interoperability among systems, such as APIs, protocols for data exchange, package managers, and so on. This potential for interoperability can be illustrated with the case of package managers. For instance, NPM provides functionalities for managing JavaScript software from publishing to installation and version control. Inside an engineering context, a solution in such molds could allow on-demand download of simulation modules that can be combined and used to evaluate behaviour of a given maritime system, such as in the Vessel.js case.

It must be noted however that while these resources provide the necessary technologies to enable interoperability among system infrastructure, they are not sufficient to ensure compatibility among product and simulation models in engineering domains. To achieve that, it is necessary that stakeholders such as software vendors, customers and standardization committees are able to establish agreements upon implementation of data schemas and methods covering the maritime sector specifically.

8 CONCLUSION AND FURTHER WORK

This study investigates use of web architectures to foster digital collaboration in the maritime sector and as a way to consolidate data sources and enhance data exchange between partners. The use of web technologies is a natural path for bridging the gap between those different partners, reducing friction for data exchange and establishing single source of truth for stakeholders involved in complex maritime projects.

In relation to previous work in the Vessel.js library, the web platform stands out from the previous projects with the Vessel.js library by being the first one to use a comprehensive three-tier architecture, including a database system for robust data management. The architecture also brings some modularization to the web platform, in the sense that an upgrade to one of its components, e.g., the database technology, would not affect the functions developed by the other two tiers. In addition to improving support, this factor gives some flexibility when customizing the adopted toolbox toward different objectives.

The work shows case studies illustrating how the web platform can enable collaboration according to each architecture tier. The collaboration in the front-end consists in the input of new information through the user interface. In the back-end, the collaboration is given by the upgrade of the software packages upon which the web platform relies. Finally, the last form of collaboration is through database information exchange, in this case, an external application was used to filter the experimental data and upload the information in the database without modifying other components in the web platform. As a consequence, the inserted data can be then streamed and evaluated in the front-end interface by the user.

The web platform was proposed as a way to manage different data in different contexts and life cycle phases. In this specific case, the web platform was applied to design and operation phases. In the future, it could be extended to accommodate digital twins and different life cycle phases until it is able to cover the entire project life cycle, thus achieving the digital thread. In this regard, there should be further development of integrated interfaces for exchanging product information among the different phases, with the goal of helping decision-makers migrate information from one stage of the life cycle to the following one in a structured and systematic manner.

The last suggestion for future work would be to work on improving simulation accuracy, possibly by including machine learning techniques. A short path to this integration would be the application of the TensorFlow.js library due to its extensive community, which provides a consolidated ecosystem for learning and support. As TensorFlow is open source, it eliminates legal restrictions for the utilization even in commercial solutions. The library implementation in JavaScript also prevents possible drawbacks related to the integration of a new programming language with the existing architecture.

REFERENCES

[1] de Reuver, M., Sørensen, C., and Basole, R. C., 2018. "The digital platform: A research agenda". *Journal of Information Technology,* ***33***(2), jun, pp. 124–135.

[2] Defense Acquisition University. Glossary of defense acquisition acronyms and terms. `https://www.dau.edu/glossary/Pages/Glossary.aspx`. Accessed: 11-June-2021.

[3] Gaspar, H. M., Rhodes, D. H., Ross, A. M., and Ove Erikstad, S., 2012. "Addressing Complexity Aspects in Conceptual Ship Design: A Systems Engineering Approach". *Journal of Ship Production and Design,* ***28***(04), 11, pp. 145–159.

[4] ISO, 2004. 10303-218 industrial automation systems and integration - product data representation and exchange - part 218: Application protocol: Ship structures.

[5] Whitfield, R., Duffy, A., York, P., Vassalos, D., and Kaklis, P., 2011. "Managing the exchange of engineering product data to support through life ship design". *Computer-Aided Design,* ***43***(5), may, pp. 516–532.

[6] El-khoury, J., Berezovskyi, A., and Nyberg, M., 2019. "An industrial evaluation of data access techniques for the interoperability of engineering software tools". *Journal of Industrial Information Integration,* ***15***, sep, pp. 58–68.

[7] Gaspar, H. M., 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), pp. 485–499.

[8] Gaspar, H. M., 2018. "Vessel.js: an open and collaborative ship design object-oriented library". In Marine Design Conference (IMDC'18).

[9] Fonseca, Í. A., de Oliveira, F. F., and Gaspar, H. M., 2019. "Virtual prototyping and simulation of multibody marine operations using web-based technologies". In Proceedings of the 2019 38th International Conference on Offshore Mechanics and Arctic Engineering, OMAE, American Society of Mechanical Engineers.

[10] Gaspar, H. M., 2017. "Javascript applied to maritime design and engineering". In 16th Conference on Computer and IT Applications in the Maritime Industries (COMPIT'19), pp. 428–443.

[11] i Miquel, S. E., Fonseca, Í. A., Gaspar, H. M., and Vieira, D. P., 2020. "An open-source library for hydrodynamic simulation of marine structures". *Marine Systems & Ocean Technology,* ***15***(3), 7, pp. 160–174.

[12] WAMIT. Wamit, inc. - the state of the art in wave interaction analysis. `https://www.wamit.com/`. Accessed: 17-December-2021.

[13] Låg, S., Vindøy, V., and Ramsrud, K., 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.

[14] Holtrop, J., 1984. "A statistical re-analysis of resistance and propulsion data". *Int Shipbuild Prog,* ***31***, pp. 272–276.

[15] Jensen, J. J., Mansour, A. E., and Olsen, A. S., 2004. "Estimation of ship motions using closed-form expressions". *Ocean Engineering,* ***31***(1), pp. 61–85.

[16] Wang, T., Li, G., Hatledal, L. I., Skulstad, R., AEsoy, V., and Zhang, H., 2022. "Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction". *IEEE Transactions on Industrial Informatics,* ***18***(3), mar, pp. 1781–1789.

[17] Lee, T., 2003. "On an empirical prediction of hydrodynamic coefficients for modern ship hulls". *Proceedings of MARSIM 2003*.

[18] Fonseca, Í. A., and Gaspar, H. M., 2019. "A prime on web-based simulation". In Proceedings of the 2019 33rd ECMS Conference on Modelling and Simulation, European Council of Modelling and Simulation.

[19] Fonseca, Í. A., and Gaspar, H. M., 2020. "Fundamentals of digital twins applied to a plastic toy boat and a ship scale model". In Proceedings of the 2020 34th ECMS Conference on Modelling and Simulation, European Council of Modelling and Simulation.