Data-Driven Ship Design

Henrique M. Gaspar, NTNU, Alesund/Norway, henrique.gaspar@ntnu.no

Abstract

This paper proposes data-driven methods thinking to the ship design practices, enabling effective data collection, quality, access, analysis and monitoring during the vessel lifecycle. Data is here understood as the key element to extract information and knowledge, and is represented in two complimentary categories, product (ship) and process (design). Ship data is studied via top-down and bottom-up approaches. The first comprises data from regressions, previous designs and parametric studies based on existing solutions, while the second is connected to specific key elements and subsystems that directly affects the mission of the ship. Data from the value-chain is studied under three distinct layers, namely overall standards (i), 3rd party choices (ii) and proprietary data (iii). Product lifecycle management tools advantages and drawbacks are discussed in light of the data challenges presented. Data-driven ship design aims to integrate product and process content according to modern DevOps practices, such as versioning, tracking, monitoring, testing, tuning and feedback. An open and collaborative ship design library is briefly introduced as an incipient initiative aiming to incorporate data-driven methods in ship design. Vessel.js is structured via JavaScript objects containing relevant information about the vessel systems and key components, as well as key performance analyses. A call for collaborative data-driven ship design using open standards closes the paper.

1. Data-Driven Methods for Ship Design

Data drivenness, in essence, is no stranger to maritime engineering and ship design. We are aware of the large amount of information required for a successful design, from properly requirements elucidation, *Andrews (2003)*, to feedback from captains and operators, *Brett et al. (2018)*. What makes then data-driven ship design worth to be studied is the gain when data can be converted to knowledge, and consequentially a more efficiency task-delivery and faster/accurate decision-making. In this context, data driven methods are connected to the build and development of tools, skills and processes that acts upon data. More important, to acknowledge the importance of such methods is to embrace a culture that properly handle data across the whole ship design value chain, *Anderson (2015)*.

Before digging on the implications that data-driven thinking can have in ship design, it is necessary to pay tribute to the large amount of new literature discussing the importance of data that appeared in the last decade. '*Big data', 'business intelligence', 'cloud storage', 'web services'* are terms that engineers from my and older generations did not hear during their studies, but that now any bachelor student is able to comprehend. The boundaries between data science, computer science, statistics and software engineering are blur, but it is clear that the terms and development in these areas are nowadays extended to (probably) every corner of the scientific community, from health care, *Murdoch and Detsky (2013),* to conceptual ship design, *Gaspar et al. (2014).*

The data-driven methods discussed in the rest of this paper are strongly influenced by the data-driven organization studies from *Anderson (2015)*, data science for business from *Provost and Fawcet (2013)* and data driven documents from *Bostock et al. (2011)*. The authors compile in their works a concise and coherent set of tools and terms that I found useful when using ship design field as object of study. A common point among the studies is the idea that good decision making comes from good data, via the data, information, knowledge and wisdom hierarchy (DIKW), knowledge discovery and data mining (KDD) and data-driving decision making (DDD). These concepts are not necessarily new, but have been heavily revisited in light of the large amount of computational power that we have today to process the data and (hopefully) extract knowledge.

Anderson (2015) explores the DIKW idea to deeply focus on the data itself, defining the pre-requisites for an efficient data-driven implementation, here exemplified for the ship design domain:

- Data Collection: the gathering itself of all data related to the business among all its sources, such as previous designs, supplier's information, rules, 2D and 3D drawings, sea trial results and operational logs.
- Data Quality: connected to the reliability of data. It can be observed in many facets, such as accessibility, accuracy, coherence, completeness, consistency and relevance. Extracting non-relevant (dirty) from relevant data is a key challenge when talking about good quality data, as well as handling missing data. A good conceptual design is strongly connected to the quality of data used to develop it, from the regressions and empirical formulas used to define main lines and dimensions; to the proper elucidation of requirements when defining how the key subsystems of the ship are able to perform the mission when neither the ship or the mission are clearly defined.
- Data Access: as important as having the data is fetching it, especially to connect one entry with another, query among all available datasets and share information with others. It can be exemplified as accessing previous designs drawings or being able to include a new system in the supplier's database. More important, the idea that efficient access to data is paramount to a proper data-driven organization, and that data should be as easy as possible available by everyone included in the design process, bounded by legal constraints, proprietary assets and level of risk.
- Data Analysis: concerns the transformation of data in the required layer (information, knowledge, wisdom). A report from the design office, for instance, saying that 2500hours were used in the project is pure data. Analysis is to compare it with other similar projects and get the information that previous designs used 10% less hours. Knowledge is understanding the consequences of these additional number of hours, e.g. higher cost for the designer but a more efficient ship to be constructed, saving hours later in the construction phase. Wisdom is connected to using this knowledge in the future, for instance trying to achieve the same amount of yard efficiency in less conceptual hours for future designs.
- Report/Alert: connected to the metrics we take from data, exemplified in ship design by the number of hours spent in a conceptual project or at the yard during construction, and how this data must be constantly reported and checked to assure profitability.

Data science in the context of engineering and processing of data is discussed by *Provost and Fawcet (2013)*. They defend that efficient DDD is connected to the merge between intuition (literally gut feeling) and analysis of data. Most of their work consists in exemplifying how to process and visualize information from data in order to gather knowledge. Traditional methods, such as regressions and other fitting models are discussed, as well as more modern techniques, such as clustering and text mining. *Bostock et al. (2011)* is more practical in his approach, proposing an open data-driven library for handle and visualize data, Data-Driven Documents (D3), which has been extremely successful for visualize data in a web environment. *Gaspar et al. (2014)* discuss how using D3 may assist the designer to extract knowledge from the data already available when describing and evaluating a design and presents several examples of this rich visual representation for ship design, Fig.1.

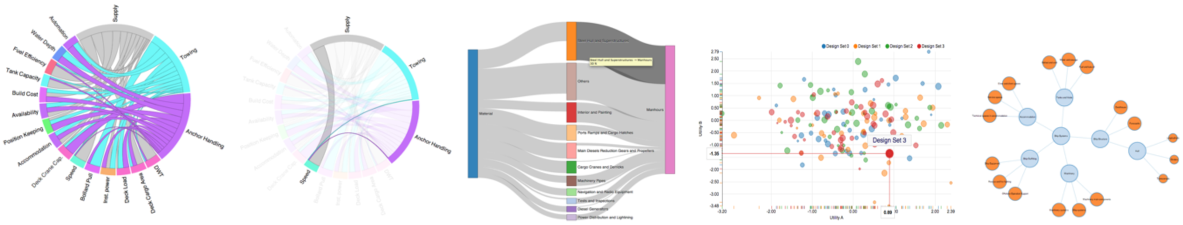


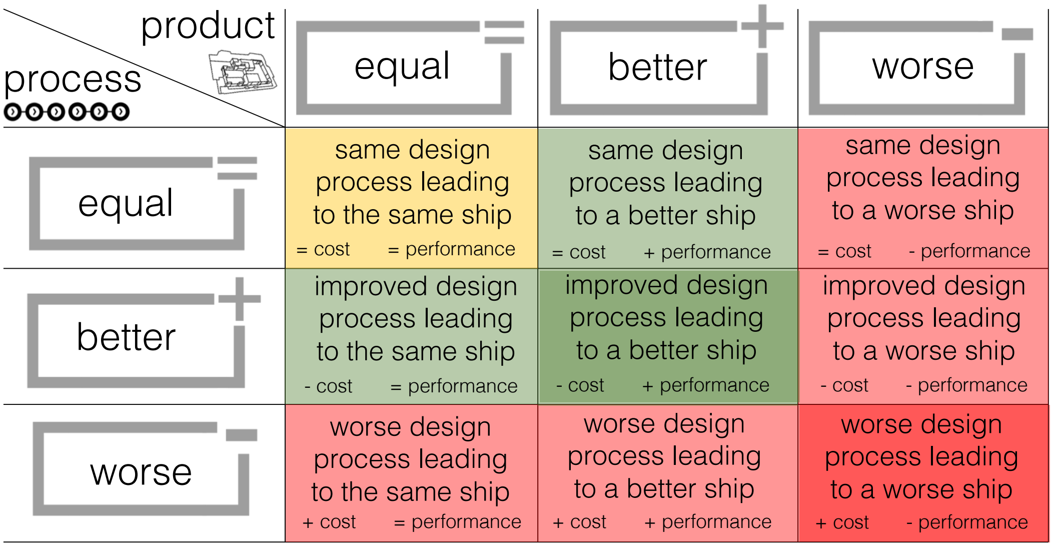
Fig.1: Data-driven methods applied to visualize data during ship design via D3, *Gaspar et al. (2014)*

In the rest of the paper, I use data-driven concepts to investigate how data is understood and handled during ship design. Section two presents a main taxonomy for data, based on the product (ship) and process (design). Later, product lifecycle management as commercial solution for handling data is commented, with pros and cons. Data-driven design is thus presented as an update of the current methods to incorporate modern software development ideas, especially an efficient data feedback into the lifecycle process, calling for a DevOps approach. Vessel.js is introduced as an open and collaborative library able to incorporate some of the data-driven elements discussed. The paper closes with a call for more open and collaborative data-methods in ship design.

2. Ship Design Data

2.1. Product and Process Data in Ship Design

Data from the product (ship) and process (design) seems to be a good start point when deciding which taxonomy should we use to converge all necessary information required for identifying a good ship. If data is strongly connected to metrics, we are therefore able to evaluate the effect of a data-driven approach by quantifying how better is a ship (e.g. higher performance than previous ship) and how better is a process (e.g. more efficient than the previous design). Note that both measurements of quality are not necessarily linked. A better ship can be obtained from a worse design process, while a more efficient process can create a ship with a lower performance than previous ships, Fig.2.



product process	equal	better	worse
equal	same design process leading to the same ship = cost = performance	same design process leading to a better ship = cost + performance	same design process leading to a worse ship = cost - performance
better	improved design process leading to the same ship - cost = performance	improved design process leading to a better ship - cost + performance	improved design process leading to a worse ship - cost - performance
worse	worse design process leading to the same ship + cost = performance	worse design process leading to a better ship + cost + performance	worse design process leading to a worse ship + cost - performance

Fig.2: Concept of product (ship) and process (design) data quality in ship design, with green areas as objectives when incorporating new data, while red areas present the negative consequences

Assuming that the insights from Fig.2 are connected to the reality of the ship design activity, it is possible to drawn certain conclusions regarding product and process data. First, a given design process will deliver a certain ship. If no additional data is incorporated, the same process should lead the same ship as before (yellow zone). The concept of improvement (or the effect of learning, as commented by *Erichsen (1994)*) in both product and process is thus connected to improvement compared to a benchmark, usually previously designs and/or ship. A better process in this context is strongly connected to the additional data that lead to the improvement of the design activities. Therefore, we can measure a better design by either achieving the same ship in lower time or even a better ship in lower time (green zones). The red zone that should be avoided is a more efficient design process that leads to a lower performance ship – no designer is happy to justify lower costs if the price to pay is a product worse than the previous.

Similarly, a better product may be the consequence of improvements in the technology and equipment available, and the same design process can lead to a better ship if established design practices are able to incorporate the data from social and technological developments, such as a more efficient propulsion system or optimized hull lines.

The greenest zone at the centre of the matrix exemplifies the key objective of incorporating new data and methods to the ship design activity: improved design efficiency (less time) with a better final ship (higher performance). The dangerous of ineffective data and mishandling the wrong information lies on the lower right corners, the reddest zone of the chart, with new data leading to a worse ship at higher cost.

The rest of this section investigates in more detail which type of data is expected to be better handled to achieve a better product (via bottom-up and top-down approaches) and process (via three layers of the upstream value chain) in ship design.

2.2. Bottom Up and Top Down approaches for Ship Data

Ship data can be represented in two complimentary categories, top-down and bottom-up, *Gaspar (2018)*. The first comprises data from formulas, regressions, previous designs and parametric studies based on existing solutions, such as the one commented in *Parsons (2011)*, most of *Watson and Gilfillan (1976)* and *Roh and Lee (2018)*. Such top-down approach to collect data usually finds a practical and consensual solution at short time and cost but lacks in innovation and carries on the bias and out of date insights from previous designs. Bottom-up data is connected to specific key elements and subsystems that directly affects the mission of the ship. Bottom-up data presents different taxonomies for vessel division and understanding, such as in methods like systems-based ship design, *Levander (2006)* and design building block, *Andrews and Pawling (2005)*. Bottom up data is the starting point for innovative and technological break-through designs, but suffers from the lack of knowledge connected to the uncertainty of one-of-a-kind projects.

A classic example of top-down are regression analysis and empirical formulas observed in most of ship design compendiums. A recent work from *Roh and Lee (2018)* adds another example of such compendiums, with formulas for the assessment of a computational ship design. Main dimensions are thus determined by traditional weight and volume equations, Fig.3a. Bottom-up approaches, such as the design building block, Fig.3b, usually starts from the key subsystem that affects the ship's mission, and main dimensions are obtained after the blocks are spatially placed.

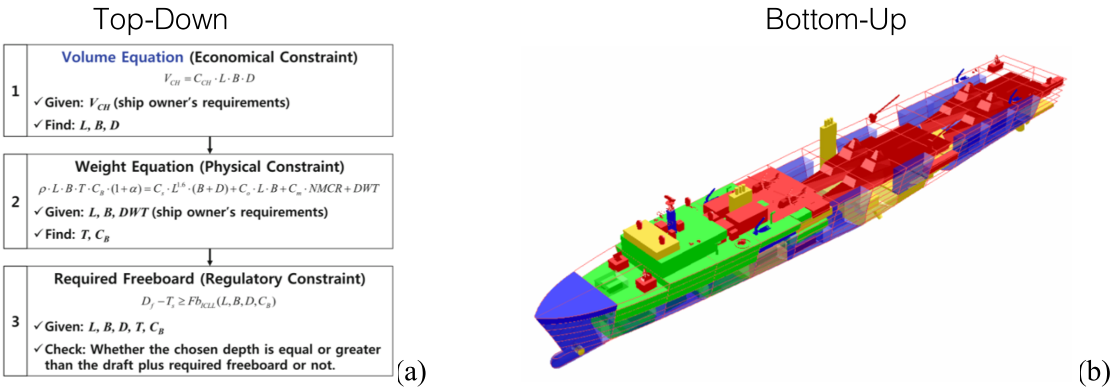

Top-Down
Bottom-Up
1
Volume Equation (Economical Constraint)
$V_{CH} = C_{CH} \cdot L \cdot B \cdot D$
✓Given: V_{CH} (ship owner's requirements)
✓Find: L, B, D
2
Weight Equation (Physical Constraint)
$\rho \cdot L \cdot B \cdot T \cdot C_B \cdot (1+\alpha) = C_s \cdot L^{1.6} \cdot (B+D) + C_o \cdot L \cdot B + C_m \cdot NMCR + DWT$
✓Given: L, B, DWT (ship owner's requirements)
✓Find: T, C_B
3
Required Freeboard (Regulatory Constraint)
$D_f - T_s \geq Fb_{ICLL}(L, B, D, C_B)$
✓Given: L, B, D, T, C_B
✓Check: Whether the chosen depth is equal or greater than the draft plus required freeboard or not.
(a)
(b)

Fig.3: (a) Top-down, *Roh and Lee (2011),* (b) bottom-up approach, *Andrews and Pawling (2005)*, to define ship design data

Note that both approaches are not excluding, and good design practices do use bottom-up and top-down data when necessary, as exemplified in *Levander's (2006)* systems based design approach, combing volume calculation and regressions based on gross tonnage. The current challenge seems to be how to properly connect both ends of the ship data in a same standard, especially during conceptual design. While a well calibrated parametric model is able to fast delivery a feasible solution (Fast Track Tool)*, Brett et al. (2018)*, the incorporation of new technologies and disruptive design changes seems to have the opposite effect.

The solution to this discrepancy seems to be yet in the hands of the experienced designer, with the empirical knowledge accumulated in the years of practice playing a larger role in decision-making than any stepwise top-down or bottom-up approach.

2.3. Design Process and Value-Chain Data Layers

The understanding of the physical elements of the ship passes through the merely idea, on the conceptual design phase, to the real and solid parts of the construction and delivery phases. Design and subsequent value chain processes should be able to incorporate this change in the essence of the product though the data type and, more important, keep a common data standard across all lifecycle processes.

Every design company has already some sort of data-driven methodology for handling value chain processes, even if not integrated. Ships are designed and analysed; Yards receive all the drawings and owners and users are able to operate and schedule maintenance of the real product. What differentiates the current system from what data-driven methods aim is strongly connected to the level of integration among the many data layers. Let us take for instance the lifecycle phase between the sales, design and the construction. The final product of the sale is a concept able to convince the owner that she should invest in that ship; the design products are the thousands of drawings required by the yard, while the final product of the yard is the ship itself. Current practice is that a department works on the concept, on an exclusive 3D model, while designers in other departments recreate the same model in several other standards (files, data-types) with much re-work until the final basic documentation. It is not rare for the shipyard to re-do many of the drawings to adjust the outfitting, due to conflicts in standards, data types (files, extensions) and sensitive design/yard information.

Fig.4 connects the traditional upstream processes from the ship design value chain, *Ulstein and Brett (2012)*, from concept to delivery, to a characterization of three data layers across all the phases: overall standards, third-party choices and proprietary data. Such understand of design data is heavily influenced by the author's experience in recent industrial projects, *Gaspar (2017abc), Andrade et al. (2015)*.

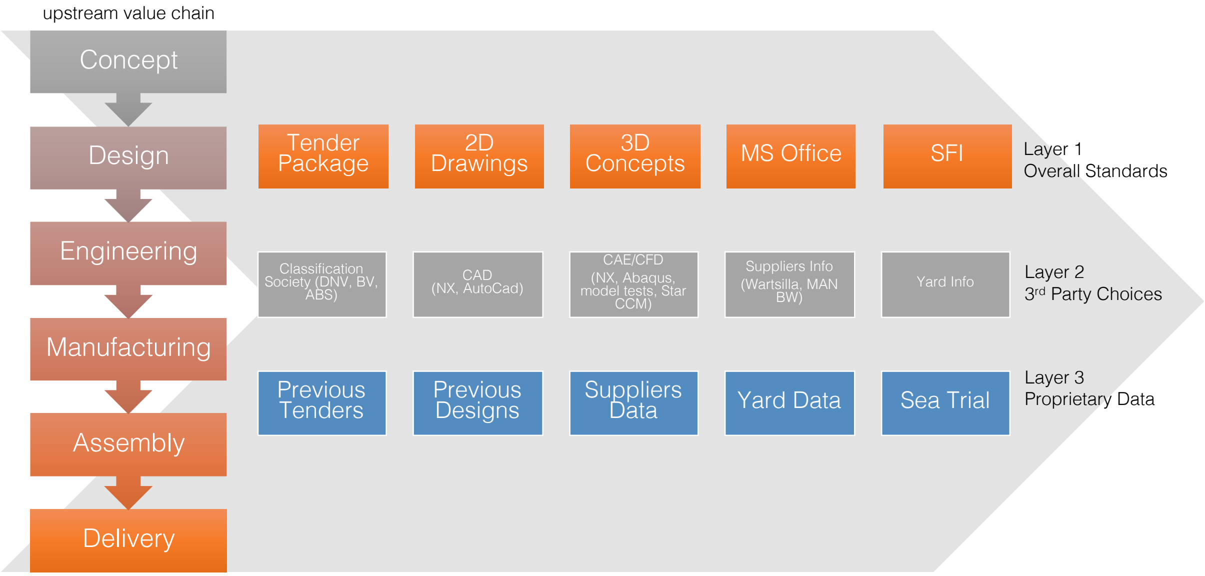


Fig.4: Design data across the upstream value chain, with data observed in three layers: overall standards, third-party choices and proprietary data.

The principle behind the insight from Fig.4 states that, while upstream value chain is a common characterization of the lifecycle process, the ship data that is transformed during these processes must be understood in face of three distinct layers. The first is connected to the overall standards that all stakeholders of the design process are able to exchange among themselves. This includes tender packages with initial description of the vessel before gaining a contract, to the presentation of 2D and 3D drawings that are discussed with the clients, suppliers and yards. MS Office and PDF documents are standard in these cases, given that they can be accessed by all involved in the process. For Norwegian yards, SFI codes is yet a common way to link design and construction. While this choice may not be the most effective to the re-use of previous data, it is usually the neat 3D concept that gains the contract for a client, while higher performance is promised presenting data guesses on previous successful designs. In theory the first layer of overall standards data should be understood and easily readable by all stakeholders during all design phases.

The large amount of options when dealing with more detailing tasks in the design and engineering processes affects tremendously how design data is handled, and it is proposed as starting point to be examined if the objective is a more efficient design process. Such third-party choices are the software, rules, suppliers and yard choices that the designer must consider and/or choose during the design process. The list of such options is too large to detail, and it is usually connected to previous successful choices from the past (e.g. use of AutoCad for 2D and 3D drawings) and promises from the new suppliers (e.g. use of a product lifecycle management suite for tackling product data handling, such as Siemens NX/Teamcenter or Aveva Suite). Such third-party data are usually very effective in presenting knowledge for a specific problem of the ship design task but are so different in definitions and standards that the mapping between one data type to another data type ends up as being a large part of the designer's work, specially via re-engineering and poor re-use of previous models. For the sake of exemplification, imagine a ship designed based on DNV-GL rules, with drawings in the standard for construction in Norwegian yards and European suppliers. The same ship constructed with ABS rules for Chinese yards with Asian suppliers would have a very similar overall presentation (1st Layer), while most of the detailing and equipment needed to be redrawn in the new third-party requirements. As noted, it is not rare that the yard redrawn info from de design office in its own standards just to comply with its standards, increasing not only the time and cost, but the risk of a serious deviation.

Proprietary data is connected to the know-how of the designer's office and yard. The challenge with these type of data is not necessarily its standards or overall type – it can be a calibrated Excel spreadsheet or a complex CFD optimization analysis – but how to protect it when needed and shared when required. Such type of data contains sensitive information from previous tender packages, designs, engineering analyses, suppliers and yard shortcuts and sea trial adjustments. It is not common that even between departments in the same company these type of data is not shared, with colleagues exchange among themselves just the minimum required to satisfy the project manager, usually in a non-numerical data format, such as power point, CAD file in PDF or docx.

3. Product Lifecycle Management Tools: Expectations and Reality

Commercial product lifecycle management (PLM) software are gaining more and more attention (and clients) in the ship design industry, with the promise of presenting a common standard to create, provide storage and sharing capabilities of the value-chain data across the ship design community. This trending goes in line with the development of such methods in other industries (e.g. automotive and aeronautic) and its features aim to tackle some of the ship design data issues discussed so far. Both advantages and limitations of PLM systems used as data-driven tools in ship design are briefly discussed in the rest of this section.

PLM can be defined as the activity of managing a company product all the way across their life-cycles in the most effective way, *Stark (2006)*, and it is one of the best-known methods to maintain a good organization of the product different parts, services, costs and suppliers through all the cycles involving a product life, from conception to scrapping. The PLM umbrella embraces many other concepts, such as modularization tools, product and systems architecture, libraries, product data management and enterprise content management, *Andrade et al. (2015).*

PLM methods provide a way of dealing with huge amount of data in complex products life-cycle. This can be achieved through many techniques, such as efficient information indexing, database management, product decomposition and analysis and project management. PLM involves not only control over the design and engineering, but also some sort of control over all value chain steps observed in Fig.2. PLM can be divided into six fundamental elements, namely *CIMdata Inc. (2011)*:

1. Database, related to indexation tools and document management
2. Modelling and Simulation tools, composed by all the software used to design and virtual

prototype the vessel

3. Value Chain Processes, related to the management of the lifecycle processes
4. Product Hierarchy management, handling the diverse classification ship functions, systems and components
5. Product Management, administrating all the information related to every physical component
6. Project Management, connecting the processes among the vessel life-cycle

PLM software promises an integrated design platform merging PLM and virtual prototype concepts, such as 3D libraries of components, CAE/CFD tools and re-use of previous designs. This platform benefits from the organizational concepts from PLM methodology, as well as virtual prototype techniques used to simulate main lifecycle process, from design visualization to construction.

A well-integrated PLM system is (theoretically) able to manage product data and process-related information as one system by use of software, providing accessibility for multiple teams across the company, such as CAD models, documents, standards, manufacturing instructions, requirements and bill of materials. However, even if the usage of PDM in maritime industry exposes great advantages, the implementation of the system evokes difficulties due to the necessity of well-established requirements, compatibility and expectations from the PLM/PDM system from the ship design company as well as the shipyard.

Keeping the re-use of 3D models as example of contrast between PLM expectations versus reality, it is possible to affirm that conventional assembly approach, which deals with connection features between pre-defined geometric entities defining the geometric positions, orientations, mating conditions, and parent-child relations is a common 3D strategy used in maritime engineering, *Levišauskaitė et al. (2017)*. It is also called traditional structuring approach where, despite of the CAD software employed in ship design processes, the connection features conserve its essential characteristics. This hierarchical assembly structure that consists of assemblies, components and features, which owns the set of entity attributes, is a distinctive feature of this approach, very similar to a traditional (but rigid) SFI classification system.

Modern PLM systems should be able to overcome such rigidness, incorporating multiple organizational breakdowns for the ship data, treating the design element as an independently managed component of collaborative design with unique and declared characteristics as access privileges, maturity status, position in ship, set of attributes, revision history, unit effectivity, and locking status. In other words, for controlling, accessing and managing the design data the components in the assembly do not need to be hierarchically ordered. Thus, it leaves an option for the ship designers to decide on the level of detail in assembly by making separate parts or subassemblies as design elements. This allows multiple taxonomies/views of an assembly, such as functional and physical, Fig.5, instead of pre-determined subassemblies of a product which add duplicates, *Siemens (2013)*.

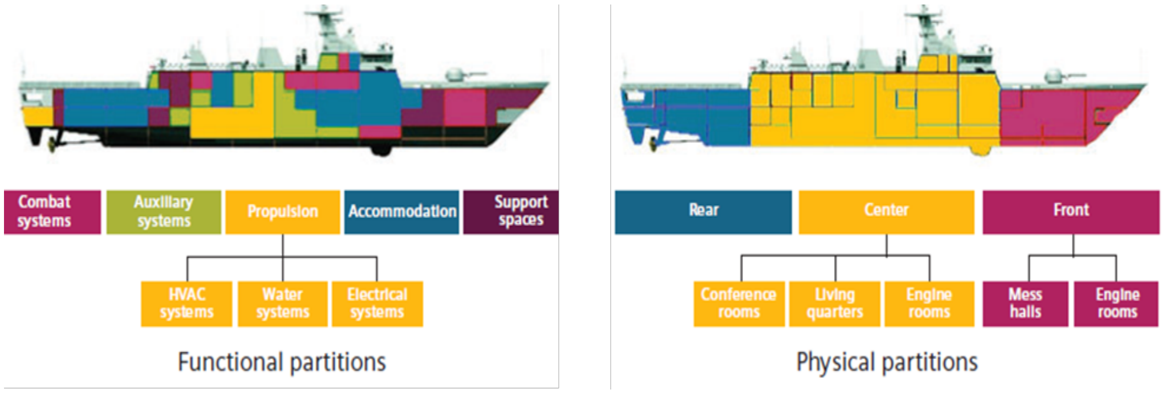


Fig.5: Multiple taxonomy for organizational breakdowns, exemplified by functional and physical partitions of a vessel, *Siemens (2013)*

By reading literature favourable to the implementation of modern PLM systems in ship design one may have the impression that the data integration problem is solved, and today designers are able to efficiently handle with a high flexibility the multidisciplinary nature of the design process and efficient 3D re-modelling facilitation. But this is not what is observed in real ship design practices. CAD and PLM tools promises for over two decades that 3D models created at the beginning of the process could be re-used during the detailing process, especially with automatic and parametrized routines able to convert the 3D in 2D with a single click, as well as smartly reusing previous models to parametrize common changes and modular vessel configuration. This is not the reality observed by the author when working direct with ship design companies in industrial projects. Much of the precious engineering time is yet used to convert, fix and re-use old models rather than engineering a more efficient hull or propulsion system.

Criticism from personal field studies shows that the integration and flexibility promised by such systems is not yet delivered in everyday ship design activities, with productivity levels even lowering when fully 3D practices are implemented, touching the core pre-requisites issues for an efficient data-driven implementation methods commented in Section 1. Add to this the economic price of proprietary software, with a high integration cost for purchasing and training among the many engineering and yard departments and limited freedom to customize libraries and engineering procedures. Some of the criticism is also extended to the users of the tool, unable to adapt traditional techniques to the new standard – but who would blame them, if the old technique still gets the work done? Some of the criticism of implementing PLM tools in ship design is listed as follows:

- Additional work to add existing data in the new format/standard/library
- Lack of ship design terms, tools and methodologies incorporated in the available tools, with the designer adapting to a more mechanical engineering approach to traditional ship design analyses.
- Lack of seamless integration with third-party ship design tools
- Proprietary and closed software package, constraining customization
- High cost to acquire, install, train personal and keep servers running.
- Resistance to experienced engineers to use a new tool
- Risk of being locked to a system, and lose independency if features and licenses terms changes
- Ignore that individuals have different preferences on how they solve the problem, and that a certain methodology proposed by the PLM system is not the most effective among users
- Inability to incorporate commonly used ship design files (data types) in the database, as well as to open the large diversity of CAD files
- Forget that Word, Excel and PPT are key means to compile and share information, forcing internal reports

It is my impression that this (lack of) compatibility problem will not solve itself soon, as some PLM vendors advertise. No type of one size fits all software is yet able to properly cover efficient data-drivenness in ship design. On the other hand, PLM tools seems a necessary evil, presenting a great step towards efficient modelling and storage of CAD models, and a large library of model and designs running seamless across the whole engineering office is a reality – these models not necessarily talk the same language, but they are there, in a common place.

4. Data-Driven Ship Design

Compiling what was discussed so far, one can conclude that efficient data-driven ship design methods must connect the fundamentals of data-driven implementation (Section 1) with the many taxonomies for product and process data (Section 2), while keeping the expectations and solving the issues presented by modern PLM systems (Section 3). In other words, data-driven ship design aims to properly identify, understand and manipulate available ship value chain data towards a better process and product. As pointed, every design process has some sort of data-drivenness incorporated to it, given that new and better ships are being developed through the years. The challenge is to make use of these state of the art techniques and increased computational power in order to keep the next design on the greenest area of the matrix, Fig.2, with more efficient processes giving higher performance ships.

Investigating literature from COMPIT along the years, for instance, we realize that the quality, number of choices and reliability of the tools used to handle the data are only better and better. It includes not PLM tools already mentioned, or the computational power per se available for designers, but the software and methods able to use it. A large number of commercial (and some free) tools are available, such as hull modelling, optimization tools, automatic rule check, CFD, powering and propulsion, flooding simulation and space allocation, to cite few. These standalone ship design tools, that most of the time can be obtained from a link in the internet and a credit card number, shows that we do have the parts that provide data, and consequentially knowledge, to solve a subset of specific ship design problems. The challenge them seems to have a common standard and/or culture able to collect, access, analyse, assure quality and, most important, connect this data among all the lifecycle phases without the need of being locked to a single proprietary and rigid system.

Some of the current challenges to incorporate a data-driven culture in ship design are summarized as follows:

- Gap between prototype (e.g. own optimization tool) and trustable/maintained tool (e.g. ship design suite)
- Integration of tools and methods with current modus operandi
- Integration among software, able to connect standard, third-party and proprietary data, Fig.2, varying from open .PDF files to advanced CFD analyses.
- Accessible to stakeholder among all value chain phases, while keeping the right level between availability and privacy
- Large investment to small number of potential buyers

The answer nowadays for tackling these challenges is simple: specialized engineering data is digested into: word reports, excel spreadsheets and power-point presentations, 2D drawings, with a small part of the information (and consequentially) knowledge being kept between the phases and gates of the process. The bottleneck relies on the large amount of available data formats and software versions that are not able to communicate among themselves, as well as bad software decision making in terms of openness and communication with competitors.

Integrated data-driven should re-uses and builds up on former designs, allowing the designer to really fetch former designs from a database, building up new concepts based on the new information, as well as re-using advanced 3D models for many value-chain phases (sales, concept, basic, construction). A data-driven ship design culture should act on keeping the collected and analysed data as accessible as possible during the design process, focusing not only on a standalone problem but on the holistic of the process, across the whole value-chain. This includes access to the analyses made during the design process, options and behaviour of the systems under the multiple operational scenarios studied, without the filter of a locked proprietary system. A data-driven design must integrate smartly the data used as input and gotten as output from the available ship design tools, as well as incorporating empirical knowledge from stakeholders.

Computer science seems to have understanding the lack of compatibility problem among data well, and the high pace development of internet tools just accelerated the need for common standards and practices, with a pressure to move faster without sacrificing reliability, *Kornilova (2018)*. This is the context where modern software development and operations (DevOps) culture emerged, with the lifecycle of a software project understood as a dynamic process evolving planning, coding, building, testing, release, deployment, operation and monitoring, Fig.6.

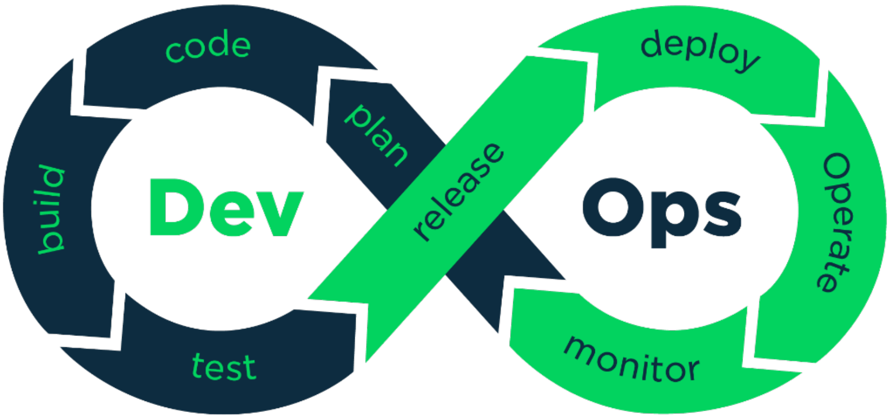


Fig.6: DevOps as culture in software development, *Kornilova (2018)*

I believe that an adaptation of DevOps software practices must be done to properly incorporate data-driven methods in ship design. Key technical practices that underpin a DevOps initiative, for instance, include getting teams to standardize on a common set of processes and tools for software delivery, *Kornilova (2018)*. Other relevant practices include:

- Data (and code/methods) available to developers and users
- Version control of files and infrastructure to enable collaboration and rollbacks
- Multi-hierarchical data, allowing plural data tags, such as functional/spatial/economic hierarchies, multiples level via *tags* or object properties (main machinery can be part of a propulsion system in one division (functional) and part of the hull in another division (physical).
- Data format as open as possible, including numbers (e.g. simulations inputs, codes and results) and models (e.g. 2D and 3D models in open source formats, such as SVG or STL)
- Collaborative storage and editing capabilities, in line with modern software repositories, such as GitHub, with features such as versioning, track of changes, reviews, ownership levels, task assignments, automatic documentation, web interface, intelligent search algorithms.
- Tools to open and manipulate the available data must be accessible to all stakeholders, without the necessity of large installation packages or extensive server configuration
- Continuous integration of collected and generated data, across the lifecycle.

DevOps also makes use of the principle of flow, feedback and continual learning of the relevant data in the process, somehow inspired by Lean practices, using as main metrics the deployment lead time for a given design project, *Kim et al. (2016)*. Attention must be giving to the monitor phase of the process, Fig.6. To monitor is strongly connected to the metrics that we take from data (Section 1), and how these data must be used to improve the quality and analyses of existing products and procedures. A culture of constant monitoring during ship design would allow a systematic and reliable storage, accessibility and quality assessment of the three data layers commented in 2.3. In a simple logic, knowledge is what is produced in a design project, via the analysis of relevant information made by people (developer or user) into content (files). Content thus is the mean to flow knowledge / information across the stakeholders. Feedback is the information gained when the data is used (or misused), with knowledge gained on how to improve in the next iteration. DevOps practices thus aims to include everyone who has a stake in the flow of knowledge by involving them early on in the collaborative process of understanding that content is the key structural element of any design project, *Kornilova (2018)*.

DevOps practices relies heavily on starting with small pilots, scaling up as debug, testing and tuning activities takes place. While ship design is usually a one-of-a-kind project, the virtual prototype technology is fully available to incorporate the small pilot approach, and such culture of testing over and over different designs under different scenarios, scaling up to include one discipline after the other, must be fully incorporated in the ship design activity, especially because the gain of using 3D models and virtual reality to test and tune analysis, from parametric hull form to seakeeping, *Chaves and Gaspar (2016)*.

5. Vessel.js: a library for Open and Collaborative Data-Driven Ship Design

Vessel.js is a JavaScript library for data-driven ship design combining conceptual ship design with a web-based object-oriented approach. It aims at the development of open and collaborative maritime engineering methods using DevOps practices. Rather than a one size fits all application, Vessel.js is considered a collection of open source examples of product and process that shares a common standard among themselves, while keeping flexibility to be customized according to the designer's need. Such development does not substitute advanced engineering and PLM tools already in practice, but rather presents a free and web-based virtual prototype platform to investigate common academy and industrial ship design tasks.

The starting point of such development is detailed in *Monteiro and Gaspar (2016)*, based on the observation of a continuous increase in the amount of information generated and handled by the design process during the vessels' life-cycle, as well as the challenge to store and exchanged the knowledge accumulated among stakeholders. Besides the standard for physical components and systems identification, it was also identified a need for a common platform for handling different types of analyses results involved in the vessel design process. Ship designers already have at their disposal advance techniques to evaluate the vessels' performance according to different merit figures, such as structural resistance, hydrodynamic forces, seakeeping capabilities and stability. A common open library having the free simulation code available, able to perform key analyses, handle the results and combine the data could definitely make a difference in the field. The conceptual design phase is yet the only real part of the value chain approach by the library, but the object-oriented approach is, in theory, able to include elements and properties from the other phases.

The library is freely influenced in its structure and development by D3, *Bostock et al. (2011)*, referring to a collection of compatible computational objects that work together to accomplish the required design task. These objects are the basic elements which the library sustains itself, building up on examples. Vessel.js represents the vessel as an JSON object, which is used to simulate different configurations, functionalities and behaviours. Currently, the library includes methods for hydrostatic and stability calculations, as well as parametric hull resistance and closed-form seakeeping equations, but it intends to grow to incorporate linear and non-linear hydrodynamics and structural models.

Vessel.js has the ambition to be an open and collaborative web-based library toolset for ship design data, able to handle relevant information about the vessel systems and key components, as well as relevant parametric data when this is within the useful range, with a special flexibility for managing different taxonomies and attributes. The data-driven framework seen here is yet in its early stages and relies heavily on community to grow but is currently able to incorporate both top-down and bottom-up approaches for the product data. The first is based on fast access of previous design data via a library of empirical formulas to create a new ship, while the second is able to incorporate innovative solutions having key subsystems modelled as JSON objects, based on a library of systems, and defining the hull and support systems around it. A more detailed paper on the capabilities of the library has been published in this conference, *Fonseca et al. (2018)*, introducing its design principles, basic functionalities, simulation examples and a call for collaborators. The library is free, open and currently maintained by the Ship Design and Operation Lab at Norwegian University of Science and Technology (NTNU) (http://vesseljs.org).

Fig.7a presents a structure of the current state of the library, including the ship (Ship.js), hull (Hull.js, Structure.js) and behaviour of the ship under different simulations (ShipState.js). Fig.7b presents a 3D representation of a PSV developed in the methods available in the library, using 106 objects to describe the main systems of the ship. Examples of this model used for fuel consumption and propulsion system optimization are presented in *Fonseca et al. (2018)*.

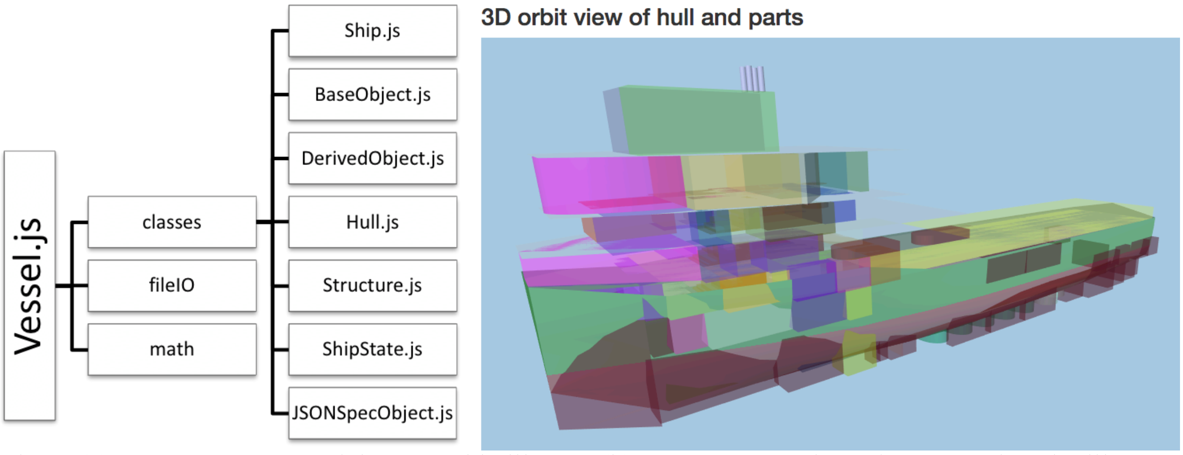


Fig.7: (a) current structure of the Vessel.js library; (b) 3D representation of a PSV using the library

The library is in its early stages of development, and it is not at the current stage able to tackle all the data related challenges discussed in the paper. Efficient handling of ship design data also faces the multifaceted aspect of its objects, with different taxonomies and hierarchies for solving different problems in each of the ship value chain phases. Finding a common standard may be practically unfeasible, and compromises must be done even in an open and adaptable object-oriented data formats, such as JSON. The library has been developed using modern DevOps concepts, with all the available features from a modern repository as GitHub, and my belief is that continuous improvement in coding, testing, feedback and monitoring will be able to tune and calibrate the methods today incorporated to be as reliable as possible when compared to existing commercial practices. This, however, will be only possible if a group of developers and operators from the academia and industry decide that it worth to use and explore the library to create and share new ship design examples.

6. A Call for Open and Collaborative Data-Driven Ship Design

I close this paper with a call for my colleagues and students to consider implementing data-driven methods in everyday design tasks. Simple prerequisites for collecting, storing, accessing and monitoring the relevant ship design data, discussed in Section 1, seems a good start. Emphasis is given in avoiding a one-size-fits-all magic tool that promises to solve all data integration – the realization that one cannot avoid the three distinct data layers at this stage presented in Fig.2 is paramount: it does not matter how integrated are the ship design analyses and drawing the in the CAD tool – the classification society and yard will be sending and receiving knowledge from that data in a different standard, and this incompatibility of a single standard must be included in the process.

In the middle to long term such incompatibility problem can be diminished if open solutions are considered. Giving up proprietary data-files in exchange of a standard among all tools seems to be a feasible path. Take for instance the JSON format, selected to describe the ship design data in the Vessel.js library. JSON structures data by directly representing objects, arrays, numbers, strings, and Booleans. Its purpose is distinctly not document mark-up, different than XML (dominant format in the web until one decade ago). JSON is flexible and does not force a natural way to represent mixed content. For this reason, became the dominant format for web applications, and should grow more, given the rapid growing of Internet of Things concept, and all existing devices will benefit much more if the newly added devices speak the most popular language, *Strassner (2014)*. Trendy concepts like 'big data', 'business intelligence' or 'internet of things' may be substituted in a close future by the 'next big thing', but it seems that the main concepts here presented as one understanding of what data-driven design may be will keep resonating. As stated in the first paragraph of this paper, such methods are not strange to maritime design, and the industry proved to be robust enough to keep the traditional successful techniques while incorporating modern concepts, such as virtual prototype and virtual reality. DevOps practices, modern repository features and open standards seems to be the next natural step.

The Vessel.js proposal here discussed are a working in process, and much of the library structure and methods intends to be improved in the years to come. The main point defended in this paper is that technology is not a bottleneck for collaborative data-driven ship design, exemplified by the current fast-paced stage of online web-development, neither the speed of the computer processors and memory size, but rather how efficient ship design data is able to be transferred from books and experience to useful reusable models. While was a good surprise to find in 2018 a new ship design compendium available for purchase, with the instigating name of 'Computational Ship Design', *Roh and Lee (2018)*, one may expect that such effort in present formulas and methods could already be digitally available for re-use, with interactive (web-based) examples of the formulas and methods discussed.

As for the development of real ship design engineering in an open library, I recognize the value of current engineering tools and PLM suites; no doubt, they are responsive for the visible gain in productivity that the maritime industry faced in the last decade. But in a long term, I believe that no other framework will allow so much continuing collaboration and re-use of codes and libraries as open and collaborative web-based repositories. Even for more advanced applications, JavaScript libraries are a reliable option, with online compilers such as WebAssembly being an actual standard feature in modern browsers, allowing C and C++ compilation direct from the client, http://webassembly.org/.

Acknowledgements

The author holds an associated professor position at the Department of Ocean Operations and Civil Engineering at NTNU (Ålesund, Norway), and has no commercial or professional connection with the software and companies cited. Statements on usability and performance reflects solely my opinion, based on personal experience, with no intention to harm or diminish the importance of current commercial state-of-the-art engineering tools. This research is connected to the Ship Design and Operations Lab at NTNU in Ålesund, which is partly supported by the EDIS Project, in cooperation with Ulstein International AS (Norway) and the Research Council of Norway.

References

ANDERSON, C. (2015), *Creating a Data-Driven Organization,* O'Reilly

ANDRADE, S.; MONTEIRO, T.; GASPAR, H. (2015), *Product life-cycle management in ship design: from concept to decommission in a virtual environment*, European Conf. Modelling and Simulation (ECMS), Varna

ANDREWS, D.J. (2003) *Marine design - requirement elucidation rather than requirement engineering.* 8th Int. Maritime Design Conf. (IMDC), Athens

ANDREWS, D.; PAWLING, R. (2008), *Concept studies for a joint support ship,* J. Nav. Eng. 44/2

BOSTOCK, M.; OGIEVETSKY, V.; HEER, J. (2011), *D^3: data-driven documents*, IEEE Trans. Visualization and Computer Graphics 17, pp.2301-2309

BRETT, P.O.; GASPAR, H.M.; EBRAHIMI, A.; GARCIA, J.J. (2018), *Disruptive market conditions require new direction for vessel design practices and tools application*, 13th Int. Marine Design Conf. (IMDC), Helsinki

CALLEYA, J.; PAWLING, R.; RYAN, C.; GASPAR, H.M. (2016), *Using data driven documents (D3) to explore a whole ship model,* 11th System of Systems Engineering Conf. (SoSE), Kongsberg

CHAVES, O.S.; GASPAR, H.M. (2016), *Web based real-time 3D simulator for ship design virtual prototype and motion prediction,* 15th Int. Conf. Computer and IT Appl. Maritime Ind., Lecce

CIMdata (2011), *Introduction to PLM*, CIMdata Inc.

ERICHSEN, S. (1994) *The effect of learning when building ships*, J. Ship Production 10/3, pp.141-145

FONSECA, Í.A.; GASPAR, H.M. (2015), *An object-oriented approach for virtual prototyping in conceptual ship design*, 29th European Conf. Modelling and Simulation (ECMS), Varna, pp.171-177

GASPAR, H.M. (2017a), *EDIS Project Proposal to the Research Council of Norway (RCN),* NTNU, Ålesund

GASPAR, H.M. (2017b), *EMIS Project: Final Report to the Research Council of Norway (RCN),* NTNU, Ålesund

GASPAR, H.M. (2017c), *JavaScript applied to maritime design and engineering*, 16th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Cardiff, pp.428-443

GASPAR, H.M. (2018), *Vessel.js: an open and collaborative ship design object-oriented library*, 13th Int. Marine Design Conf. (IMDC), Helsinki

GASPAR, H.M.; BRETT, P.O.; EBRAHIM, A.; KEANE, A. (2014) *Data-driven documents (D3) applied to conceptual ship design knowledge*, 13th Int. Conf. Computer and IT Appl. Maritime Ind., Redworth

KIM, G.; WILLIS, J.; DEBOIS, P.; HUMBLE, J. (2016), *DevOps Handbook,* Revolution Press

KORNILOVA, I. (2017), *DevOps is a culture, not a role!*, https://medium.com/@neonrocket/devops-is-a-culture-not-a-role-be1bed149b0

LEVANDER, K. (2006), *System based ship design*, NTNU Marine Technology, Trondheim

LEVIŠAUSKAITĖ, G.; GASPAR, H.M.; ULSTEIN, B. (2017), *4GD framework in ship design,* 16th Int. Conf. Computer and IT Appl. Maritime Ind., Cardiff

MONTEIRO, T.; GASPAR, H.M. (2016), *An open source approach for a conceptual ship design tools library*, 10th HIPER Conf., Cortona

MURDOCH, T.B.; DETSKY, A. (2013), *The inevitable application of Big Data to health care,* JAMA 309(13), pp.1351-1352

PARSONS, M.G. (2011), *Parametric Design,* Ship Design and Construction Vol. 1, SNAME

PROVOST, F.; FAWCETT, T. (2013), *Data Science for Business.* O'Reilly

ROH, M.; LEE, K.Y. (2018), *Computational Ship Design*. Springer

SIEMENS PLM software (2015), *4th Generation Design / Providing the next-generation design paradigm for shipbuilders,* Siemens

STARK, J. (2006), *Product Life-Cycle Management,* Springer

STRASSNER, T. (2014), *XML vs JSON,* TUFTS University

ULSTEIN, T.; BRETT, P.O. (2012*), Seeing what's next in design solutions: Developing the capability to build a disruptive commercial growth*, 11th Int. Marine Design Conf., Glasgow

WATSON, D.G.M.; GILFILLAN, A.W. (1976), *Some ship design methods,* Trans. RINA 119, pp.279-324