Comparison of system modelling techniques for autonomous ship systems

Sunil Basnet*, Osiris A. Valdez Banda, Meriam Chaal, Spyros Hirdaris and Pentti Kujala
Aalto University, School of Engineering, Marine Technology Group, Espoo, Finland

ABSTRACT

As autonomous ships are currently developed, modern technologies are implemented into ship systems for enabling autonomous operations. Tight coupling in safety-critical systems created new challenges for the engineers and operators. Designing, operating and analyzing these complex systems requires a deep understanding about the system composition, requirements and expected behavior or functionality. The increasing complexity of the systems requires the implementation of modern model-based approaches. Instead of large texts, these new modelling techniques aim to present detailed system information with simplified models. This paper compares system modelling techniques known as System Modelling Language (SysML) and Object Process Methodology (OPM). These methods are used to model a Dynamic Positioning system (DP-system). Results show that the SysML is more suitable than OPM for modelling the autonomous ship systems due to its ability to present detailed system information in a simple and coherent way.

Keywords: Modeling methods; System Modeling Language; Object Process Methodology; Autonomous ship system; Dynamic Positioning System

1 INTRODUCTION

Numerous maritime industry stakeholders are currently exploring the options for developing new autonomous ship concepts (MUNIN, 2016). To date, research and engineering efforts focus on the development of technologies that could enable safe autonomous ship operations. However, the increased functionalities of autonomous engineering systems onboard ships are supported by advanced software and enable complex operations (Levander, 2017). This increased complexity represents new challenges for engineers in the system's development, operation and management. These challenges link with three keystreams of innovation: (a) the management of system complexity; (b) the understandability of the system, and (c) the communication of the gathered information (Holt & Perry, 2019). The solutions to handle these issues need to have a consistent interconnection. This need for interconnection is evident as the absence of complexity management generates a difficult process for gathering information and results in an inefficient communication of the system information. This is the reason why there is a need for new alternatives for the systemic handling of complexity in modern autonomous systems and associated operations. These alternatives should support engineers and operators in the processing of system information. Moreover, the alternatives must effectively communicate and utilize the system information throughout diverse engineering processes such as development, analysis, operation and maintenance (Holt & Perry, 2019).

Advancement of technologies and their complexity requires a modern model-based approach (Grobshtein et al., 2007). Accordingly, engineers and operators must understand how system components interact with each other and surrounding systems (Weck et al., 2011). Novel methods based on Model Based Systems Engineering (MBSE) and System of Systems Engineering (SOSE) could offer solutions for the modelling of the complex modern systems. The International Council on Systems Engineering (INCOSE) defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation. It begins at the conceptual design phase and continues throughout development and later life cycle phases" (Friedenthal et al., 2007). Most of the methods based on MBSE are supported with computer tools for handling the system complexity. These enable a systematic organization of the modeling

* Sunil Basnet: +358451812811, Sunil.basnet@aalto.fi

process. MBSE implements the principles of SOSE for a holistic modeling that begins with a general system level and proceed to subsystems and components. This holistic modeling includes system descriptions that are effectively communicated by diagrams and text.

The models providing the system information have been utilized for various purpose in systems safety engineering. Some analysis methods such as the System's Theoretic Process Analysis (STPA) and the Functional Resonance Analysis Method (FRAM) utilize models to guide the analysis process. For example, in STPA, the so-called safety control structure model provides system information about the controllers, controlled processes and the interactions between them. This information is then used to define the scenarios where the system can be in an unsafe state (Leveson & Thomas, 2018). On the other hand, in FRAM, the model presents functional system interactions using different aspects such as requirements, resources, control, time, input and output. This system information is then utilized to understand the system's couplings and performance variability (Hollnagel, 2012). Thus, the models providing the system information have a crucial role when conducting a system analysis. With these purposes, models have been effectively utilized in several domains for enhancing system design decision making (Russell, 2012), system development (D'Ambrosio & Soremekun, 2017), Safety analysis (Mhenni et al., 2013) and Security analysis (Best et al., 2007).

This paper compares two state of the art modeling methods based on MBSE and SOSE principles knowns as System Modelling Language (SysML) and Object Process Methodology (OPM). The paper includes a case study where a Dynamic Positioning (DP) system is modeled to compare the funcitonality of these methods. The DP-system is selected for this study because it is considered to be one of the main systems where autonomy in maritime operations is critical in terms of both positioning and navigation. Based on the generated models, the similarities and differences of these methods are compared. Finally, the applicability of these methods in autonomous ship systems is discussed.

2 METHODS

2.1 System Modelling Language (SysML)

SysML is a modelling language developed by Object Management Group (OMG) in 2007. It was derived from UML (Unified Modeling Language), which is widely used in Software Engineering (Friedenthal et al., 2015). SysML has been recognized by the International Standards Organization (ISO) since 2017 (ISO/IEC 19514:2017). OMG defines SysML as "a general-purpose graphical modelling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. It is an enabler of a MBSE approach to improve productivity, quality, and reduce risk for complex systems development." (Object Management Group, 2017). SysML contains 9 different types of diagrams with each of them having a specific purpose, different level of abstraction and providing a different view of the same system. These diagrams aim to model the structural and behavioral aspects of the system and are classified as: (Holt & Perry, 2019):

I. Structural diagrams:

a. The **block definition diagram**, labelled *bd,* presents the hierarchy of system elements (systems, sub-systems and components) to define their properties. For each element, a block is created with a possibility to add any related properties or specifications as appropriate such as weight, dimensions and cost.
b. The **internal block diagram**, labelled *ibd,* presents the internal structure of the system or sub systems. It presents how the parts are interconnected from an inside perspective by adding parts, ports and connectors.
c. The **requirements diagram**, labelled *req,* is used to list the requirements of the system elements, which can include guidelines, standards, rules etc. During the system development life cycle, this diagram allows engineers to verify and validate these requirements.
d. The **parametric diagram**, labelled *par,* is used for conducting engineering analysis as well as verification and validation of the system requirements. The

constraints for the system analysis are defined and then the performance of systems is calculated and evaluated using the property values defined in system elements block. It can include various engineering analysis such as trade studies, sensitivity analysis, performance analysis and design optimization (Friedenthal et al., 2015).

e. The **package diagram**, labelled *pkg,* is used to organize all SysML diagrams. The diagrams with similarities are identified and grouped in a package (folder) which can be referenced with a unique name as suitable by the modeler. Furthermore, if a diagram belongs to two different groups then an import relationship can be applied to import the diagram or diagram elements from one package to another.

II. Behavioral diagrams:

a. The **activity diagram**, labelled *act,* presents the flow-based behavior of the system when performing certain activities. It shows the flow of control, flow of objects, input required for executing the activity and output that the activity produces.
b. The **sequence diagram**, labelled *sd,* shows the sequential flow and exchange of messages between different system elements when they interact with each other.
c. The **use case diagram**, labelled *uc*, describes the functionality of the system by presenting the actors (users of the system) and tasks required to execute the function. Furthermore, the operational requirements of the system can be refined using the use case diagram, which shows how the system functions fulfil the system requirements.
d. The **state machine diagram**, labelled *stm,* presents the behavior of the system elements when the state transitioning occurs. It is used to describe the state-dependent behavior of the system elements during the system operation.

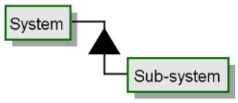
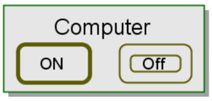
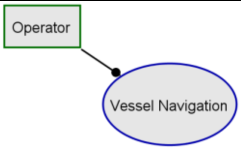
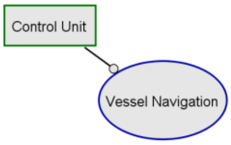
2.2 Object Process Methodology (OPM)

OPM is a system modelling paradigm introduced by Prof. Dov Dori in 1995 (Dori, 1995). It is a holistic approach, which models a system's structural and behavioral aspects in a single and unified diagram. It can support engineers during system design, development, maintenance, and effective communication (Weck et al., 2011). In 2015, ISO recognized OPM as an international standard modeling language for producing conceptual models of a system (ISO/PAS 19450:2015).

OPM focuses on three entities that are inherent in a system namely objects, processes and the links in between them. Dori (2002) defines objects as the "things that exist in the system" and consequently, he defines processes as the "things that happen in the system. For presenting these system descriptions, it uses both the graphical form via Object-Process Diagrams (OPDs) and textual form via Object-Process Language (OPL).

The OPD's consist of one System Diagram (SD) and many In-Zoomed diagrams. The SD presents the system level (top level) elements, while the In-Zoomed diagrams present each of the elements of SD in higher detail. In this way engineers may start designing at abstract level and add refinements as needed. Each OPD includes a collection of sentences, in OPL format, (textual modality). OPL is defined by Dori (2016) as "a subset of English that expresses textually the OPM model that the OPD set expresses graphically". This feature creates human readable auto-generated texts that can be useful for engineers that prefer texts over graphics and can also be used to create technical specifications (Dori, 2016). Table 1 presents some symbols and their representation in OPDs.

Table 1: Some symbols and their representation in OPD's

Symbol	Representation	Symbol	Representation
	**Non-physical Object**: Flat rectangle		**Process**: Eclipse
	**Physical Object**: Shaded rectangle		**Aggregation -participation link**: Black filled triangle **OPL**: System consists of Sub-system
	**States**: rounded rectangle **Initial state (on):** thick border **Final state (off):** double border		**Agent link**: line with black filled circle **OPL:** Operator handles Vessel Navigation
	**Environmental Object**: with dashed border		**Instrument link**: line with white filled circle. **OPL**: Vessel Navigation requires Control Unit.

3 MODELING A SHIP DYNAMIC POSITIONING SYSTEM (DP-SYSTEM)

3.1 Dynamic Positioning system

The International Maritime Organization (IMO) defines Dynamically Positioned vessel (DP-vessel) as "a unit or a vessel which automatically maintains its position (fixed location or predetermined track) exclusively by means of thruster force". It also defines the Dynamic Positioning system (DP-system) as "the complete installation necessary for dynamically positioning a vessel comprising of a power system, thruster system and DP-control system" (IMO-MSC/Circular.645).

The DP-system estimates the required thrust and rudder angle to maintain the vessel position against wind, waves and current; and controls the engine, thrusters, propellers and rudders accordingly for reaching and maintaining the desired position. It consists of several position-reference units and sensors to estimate the vessel motion and position in addition to the forces affecting them. The sample DP-system, K-Pos DP-21 (Kongsberg, 2014), is modelled in the upcoming sections with a focus on automatic vessel positioning mode. This system satisfies the requirements of IMO equipment Class 2 as specified in IMO-MSC/Circular.645. The details of this system and its functions are available in Kongsberg (2014) and is used for modeling in Sections 3.2 and 3.3.

3.2 Modeling with SysML using Visual Paradigm tool

Figure 1 presents the package diagram for the DP-system. The package diagram consists of 4 main packages: requirements, behavior, structure and parametric, which are further refined into sub packages as shown in the figure.

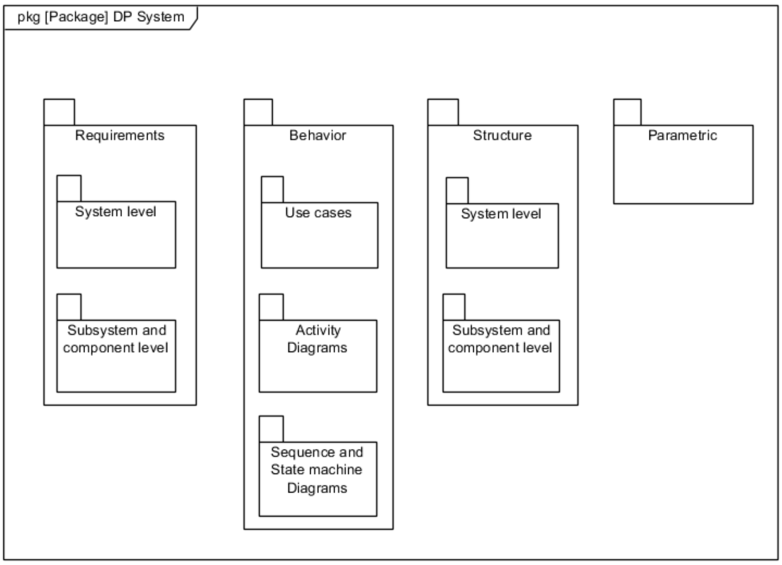


Figure 1: The package diagram for DP-system

Next, inside the requirements package, the requirements diagrams are created for the DP system, sub-systems and their components. Figure 2 presents the system level (DP-system) requirements from IMO MSC/Circular 645 for Class 2 vessels. Each of the requirements inside the diagram has a title (heading), text (describing the requirement) and ID (identity number). Furthermore, as shown in the requirement titled as “International Standard Units” in Figure 2, it can also include various fields such as verify method, risk involved due to unfulfilled requirements, and current status of the requirement verification.

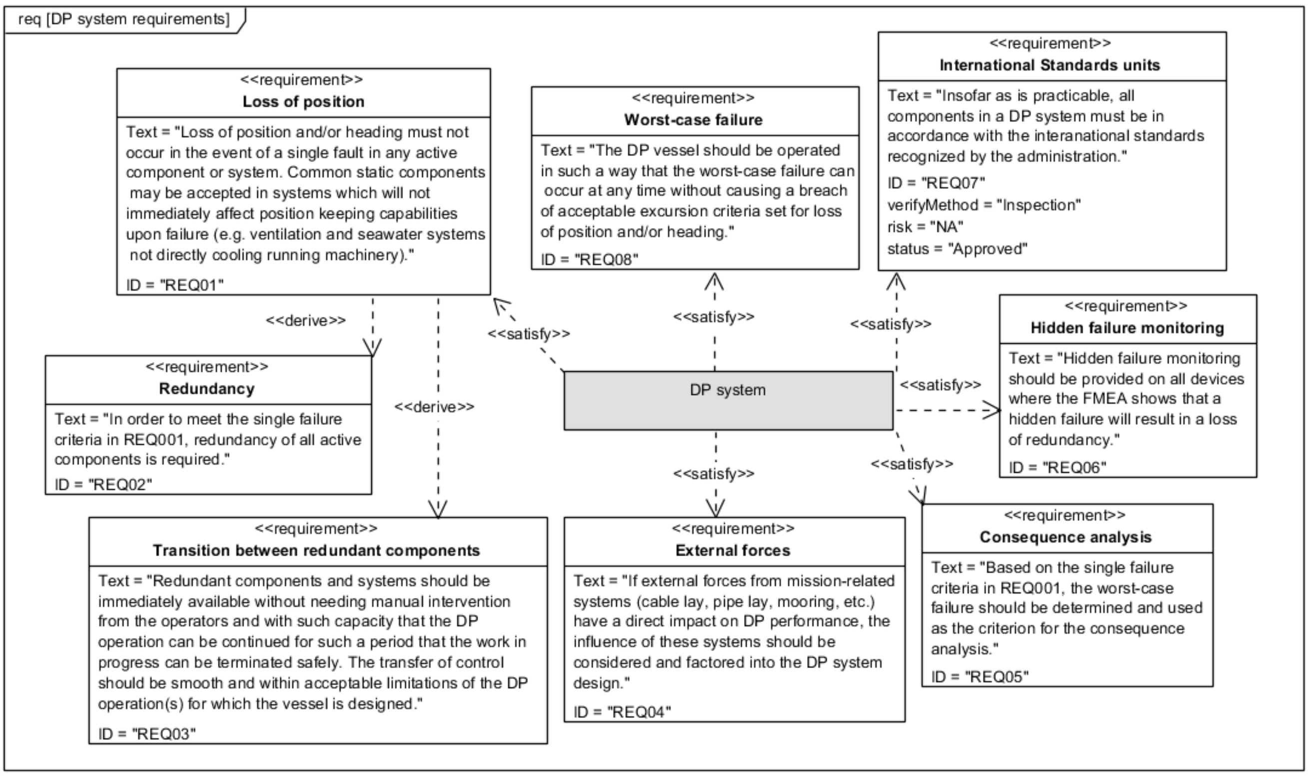


Figure 2. The requirement diagram for DP system

Then, the diagrams presenting the structural aspects of the DP-system are created. Figures 3 and 4 present the block definition diagram of the DP-system domain and DP control unit respectively. Each of the system elements are modeled as a block and are connected with a composite association (whole-part relationship). Furthermore, each of the components block can include relevant properties and values. For example, the “operator station” block in Figure 6 includes fields where the information about its current, voltage, dimension and weight can be placed.

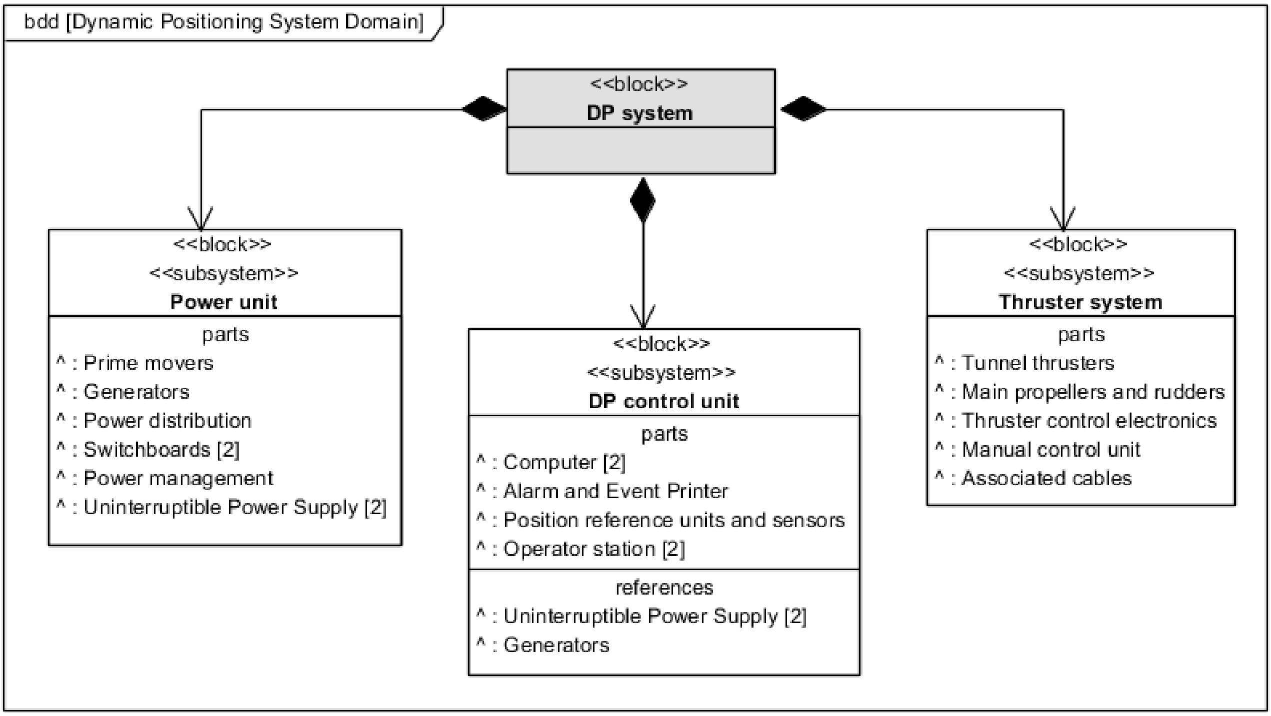


Figure 3: The block definition diagram presenting the structure of DP-system domain

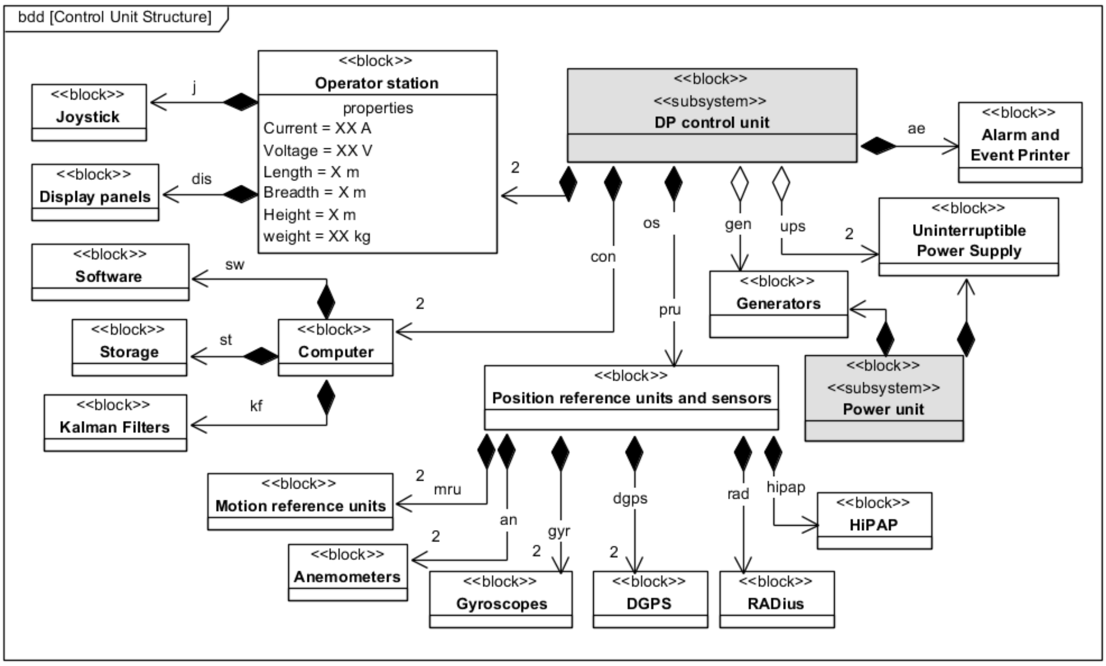


Figure 4: The block definition diagram presenting the structure of the control unit

Figures 5 and 6 present the parametric diagram for analyzing power consumption of control unit and power efficiency of the DP system respectively. This is achieved by importing the property values (see the Operator station block in Figure 4) and using power equations.

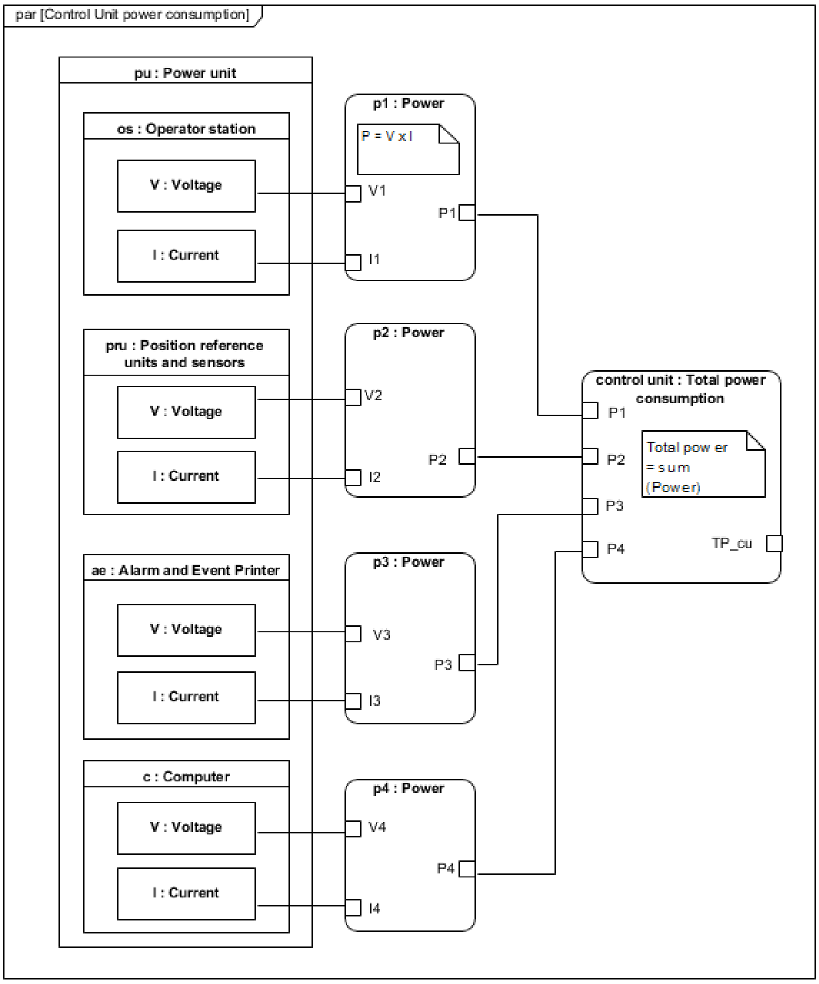


Figure 5: The parametric diagram for calculating power consumption of DP control unit

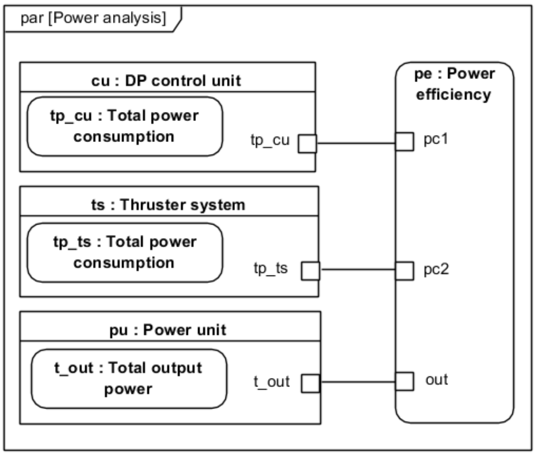


Figure 6: The parametric diagram for power analysis

After modeling the structural aspects of the system, the diagrams presenting the behavioural aspects are created. Figure 7 presents the use cases and the users that are involved in automatic vessel positioining. The <<include>> stereotype denotes the cases that are always active during the main operation, while <<Extend>> stereotype denotes the cases that are only active in certain scenarios.

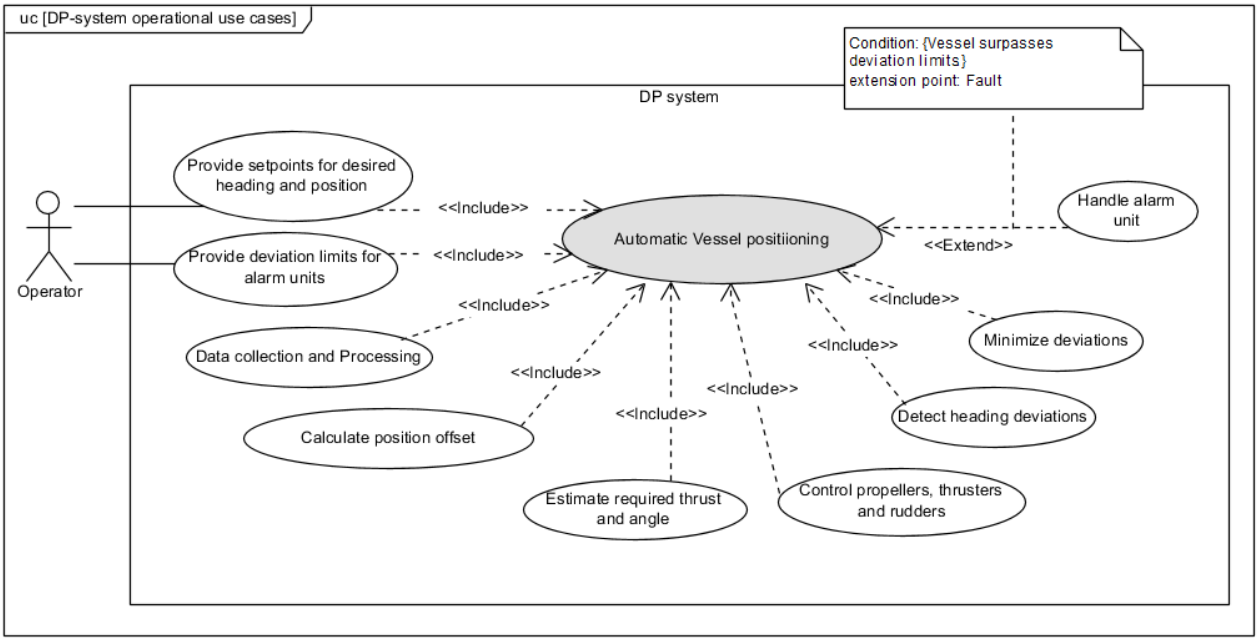


Figure 7: The use case diagram presenting use cases of DP-system

Figures 8 and 9 present the activity diagrams for the DP operation and its sub-function (data collection and filtering) respectively. It presents the sequence of the tasks involved in the process using direction of arrows. The tasks in parallel are shown with Join nodes and Fork nodes (black filled rectangles). In addition to the sequence of the tasks, the activity diagram of data collection and filtering process presents the involved components, required inputs (satellite, transponders etc.) and outputs (filtered positional data, wind speed data etc.).

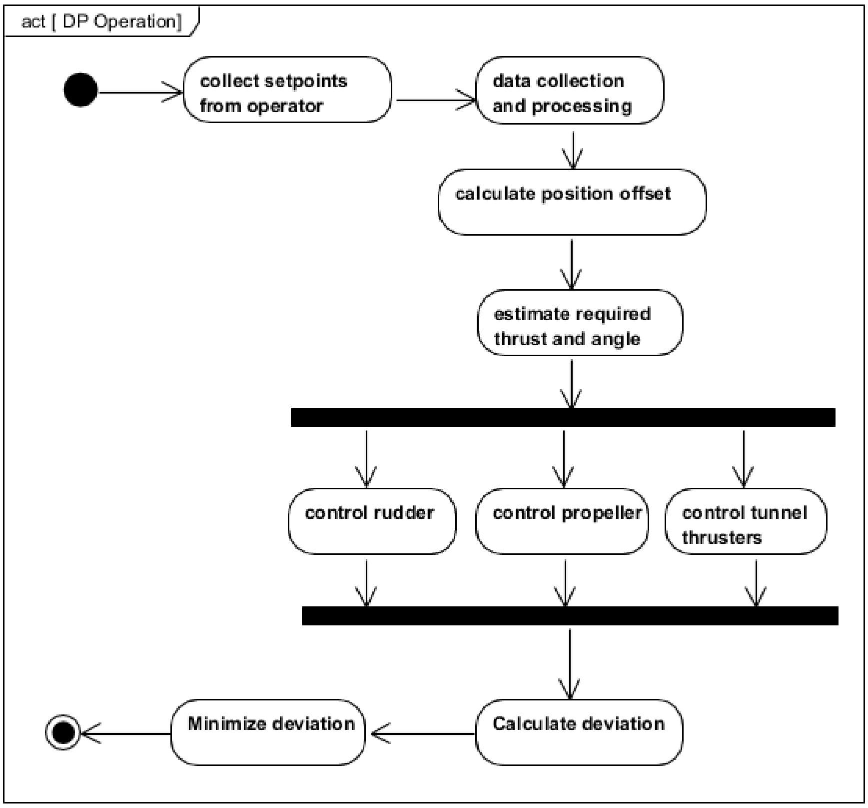


Figure 8: The activity diagram presenting the actions during DP operation

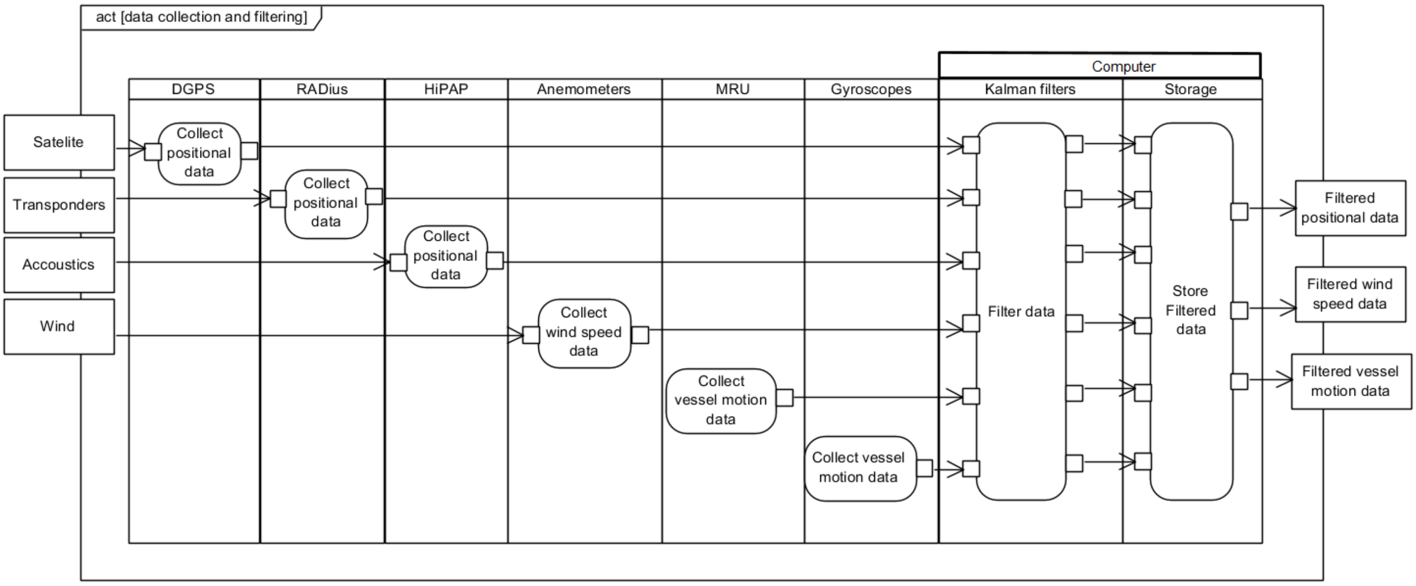


Figure 9: The activity diagram for data collection and filtering process

Figure 10 presents the state machine diagram for "handle alarm unit" use case. The diagram presents different states of the alarm unit (Alarm off, Alarm on etc.), their triggering causes (e.g. Activate warning if..), system functions during that specific state (e.g. Activate warning message and signals). All of these behavioural diagrams are then placed in their corresponding packages in the package diagram.

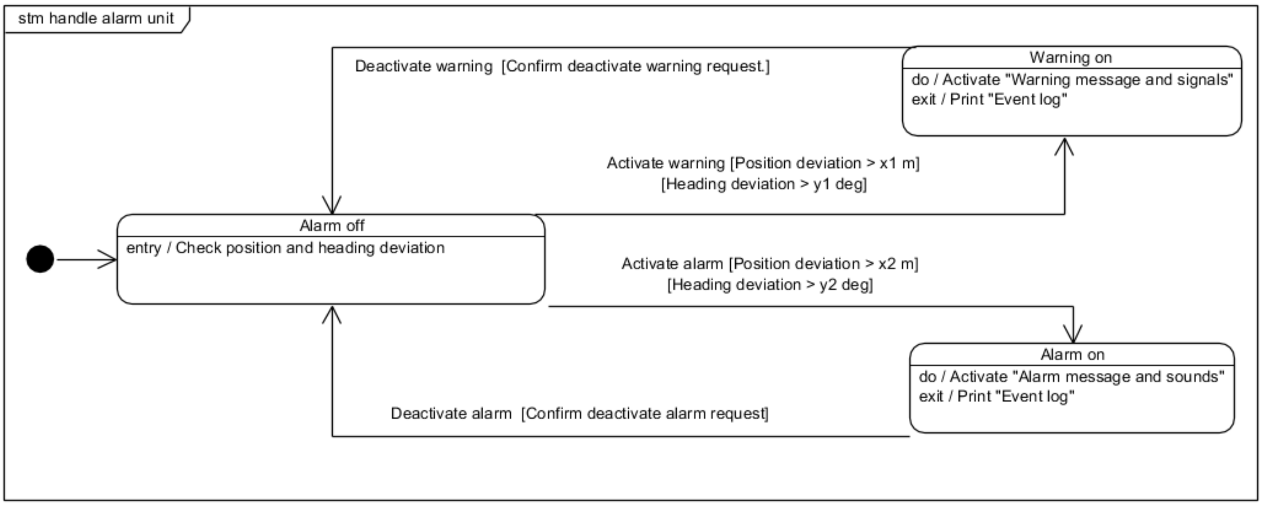


Figure 10: The state machine diagram for handling alarm unit of DP-system

3.3 Modeling with OPM using OPCAT tool

Figure 11 presents the System Diagram (SD, top hierarchical level) for Automatic vessel positioning. It shows the main process (automatic vessel positioning), involved objects (e.g. operators, individual DP-systems, etc.) and their links. Furthermore, it also presents the state transition of an object (vessel) from initial position to required position achieved by automatic vessel positioning function.

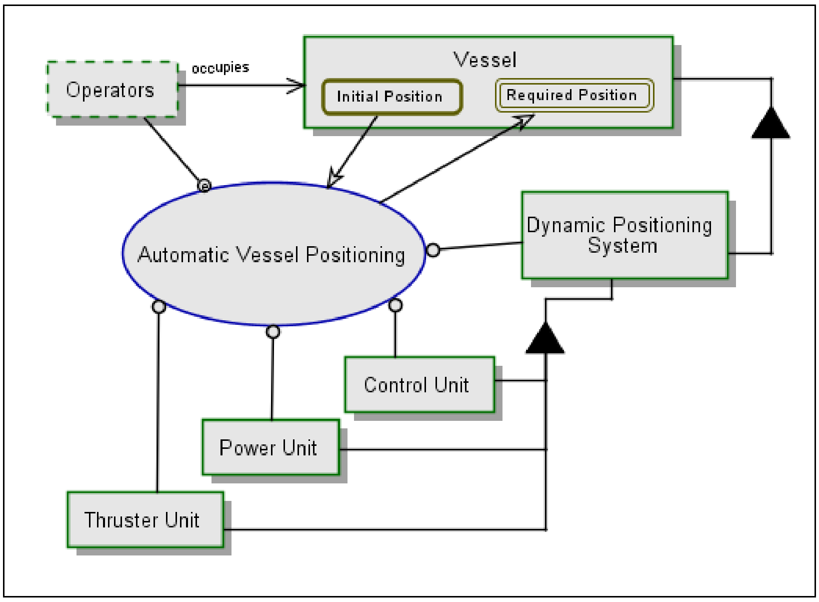


Figure 11: The top level OPD (SD) for DP-system operation

Figure 12 presents the In-Zoomed diagram of the SD. It shows the subprocesses and involved components in the automatic vessel positioning process. The sequence of the subprocesses is represented using a top-down approach.

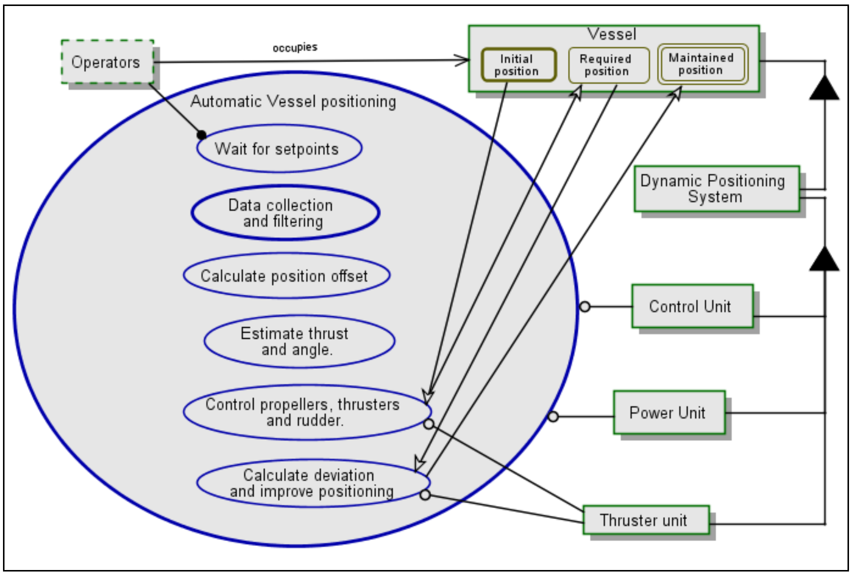


Figure 12: The In-Zoomed diagram of automatic vessel positioning

The OPD is then further refined by adding In-zoomed diagram for each of the subprocesses. Figure 13 presents the In-zoomed diagram for the subprocess “Data collection and filtering”. It presents the subprocesses and objects involved during Data collection and filtering.

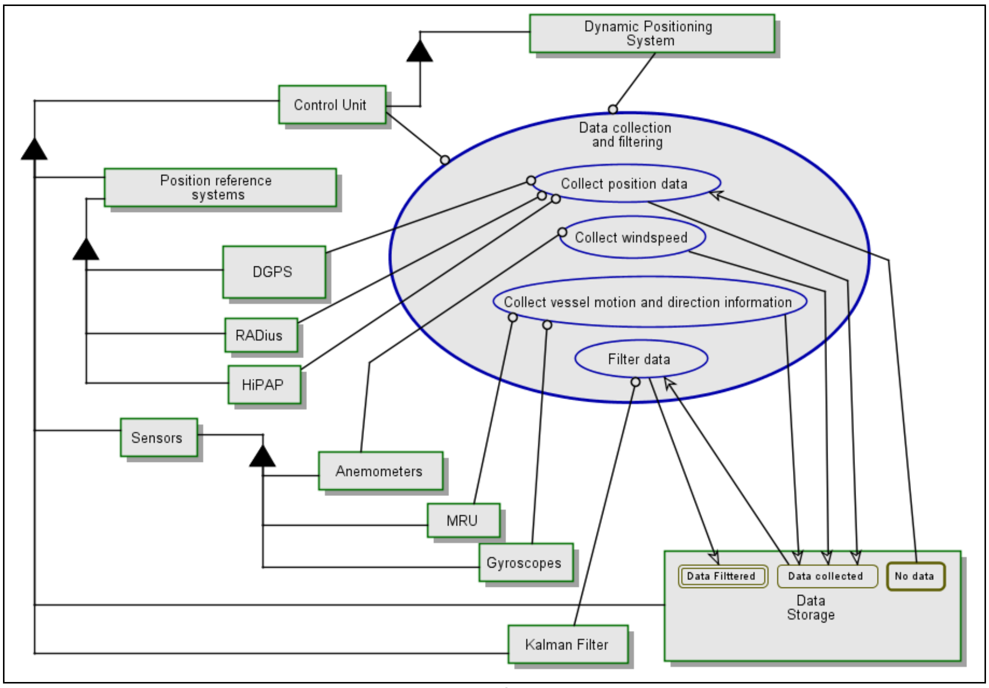


Figure 13: The Zoomed-in diagram for data collection and processing

The OPL of each of these diagrams were auto-generated. The OPL for SD is as following:

Vessel is physical.
Vessel can be Dynamic or Maintained position.
Dynamic is initial.
Maintained position is final.
Vessel consists of Dynamic Positioning System, Power Unit, and Thruster unit.
Dynamic Positioning System is physical.
Dynamic Positioning System consists of Control Unit.
Control Unit is physical.
Power Unit is physical.
Thruster unit is physical.
Operator is physical.
Operator triggers Vessel positioning.
Vessel positioning requires Control Unit, Operator, Dynamic Positioning System, Thruster unit, and Power Unit.
Vessel positioning changes Vessel from Dynamic to Maintained position.

Figure 14 presents the simulation of the SD diagram of DP-System where the left diagram represents the early stages and the right diagram represents the final stages . The simulation demonstrates the change of states and the sequence of the processes in real-time. The green color in the simulation refers to the active objects and the purple colour refers to the active processes. Furthermore, the change of state of an object is demonstrated with a moving red circle in the links and the changing color of the state itself.

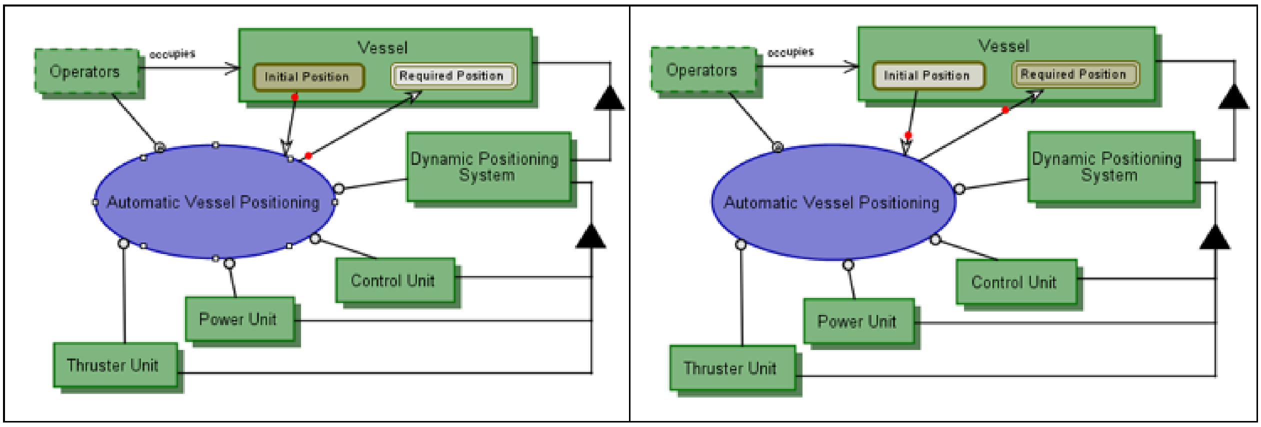


Figure 14: A simulation presenting the state change of the vessel (early stage of simulation at left and final stage of simulation at right)

4 DISCUSSION

4.1 Comparison of OPM and SysML

Based on the generated models, Table 1 summarizes the key aspects and strengths of each methods.

Table1: A summary of key aspects of OPM and SysML

Features	SysML	OPM
Modeling structural and behavioral aspects	• Detailed information of the system using different type of diagrams. • Distinction between system owned and shared components.	• Distinction between physical and non-physical objects. • Easier and faster to create and understand as it consists a single unified diagram.
Models management	• An entire diagram "package diagram" is dedicated for the model's management.	• Zoom-in and zoom-out approach is used to easily navigate between the levels. • Easier to manage because it consists of a single diagram (9 in SysML)
Availability of tools support and related features	• A wide range of tools is available in the market with limited or full features of SysML.	• The only available tool offers full features of OPM
Additional features	• Includes Requirement diagram that presents the requirements of the system elements and allows engineers to verify and validate these requirements during system life cycle. • Includes Parametric diagram that enables various engineering analysis such as trade studies, sensitivity analysis, performance analysis and design optimization.	• Allows dynamic simulation of the model, which enables the modeler to visualize the system functions and test its executability. • Generates texts describing the models automatically which can be useful for technical and non-technical stakeholders. • Allows coders to make models using codes instead of using graphical user interface.

The results show that both methods are effective in modeling the structural and behavioral aspects of the system. However, SysML provides more detailed information of the system than OPM as it consists of several types of diagram. In contrast, OPM uses a single unified diagram to present the system composition and behavior in a simple way. Due to these differences, OPM models are easier and faster to create and understand than SysML models.

There are also some important unique features in both methods. OPM allows dynamic simulation of the system. Whereas in SysML, the simulation is only possible for certain diagrams and with specific tools. Other key features available in OPM is the ability of textual modality (automatic generation of texts from the models) and also allows coders to make models using codes. On the counterpart, SysML includes requirement diagram and parametric diagram which are lacking in OPM. The requirements diagram in SysML can include the requirements that must be satisfied during the design, development and even during the operation of the system. The parametric diagram enables to conduct various engineering analysis of the systems such as trade studies, sensitivity analysis, performance analysis and design optimization.

4.2 Applicability of methods to autonomous ship systems

It is highly expected that the autonomous ship systems will consist of tightly coupled components with complex interactions within the system and with the environment (Levander, 2017). The functions of operators onboard will be added to the system components (i.e. software and hardware), which are already operating with many complex functions. The functions that cannot be assigned to the existing system components will be assigned to the new systems with advanced components. As a result, it is evident that an autonomous ship will include high number of advanced systems on board with higher interactions between them when compared to the traditional ships. Thus, the current approach, where the system description is mostly attempted to communicate with texts and traditional approaches such as tree structure and functional block diagrams might not be enough to understand these complex interactions and to communicate them. Furthermore, the advancement of information technology, and the availability of standard methods and tools allows the engineers to move from the document-centric approach to model-based methods.

Both OPM and SysML consist of features that can be useful for the development, analysis and operation of autonomous ship systems. However, there are some important features such as requirements and parametric modeling that are lacking in OPM. Marine industry is highly dependent on rules and regulations. There are several requirements such as DNV GL (2013) from regulatory bodies that need to be fulfilled before ships deployment. Thus, it is necessary to model the requirements of ship systems and their operation from the earliest design phase to avoid any potential issues afterwards. Furthermore, the parametric diagram in SysML allows engineering analysis to be conducted from the design phase and throughout the system lifecycle. Although, there has been an attempt to include requirements and parametric in OPM models (Dori, 2016), the addition of requirements and parameters in an OPD diagram may lead to information overload. Furthermore, as autonomous ships will consist of high number of components and interactions, the attempt of adding all information in a single diagram will make models large and complex to understand. Although, the features of OPM such as object process language and simulation can prove beneficial for engineers and different stakeholders, the features available in SysML are of higher importance. Nonetheless, it is still necessary to assess the suitability of adding these missing features in SysML, which can make SysML even more complete and applicable to autonomous ship systems.

Ships in general are safety-critical and the development of autonomous ships are of high interest among several stakeholders in marine industry. Thus, it is expected that the models should be as detailed as possible. As SysML provides modeling capabilities for systems in higher details than OPM and considering all of above reasons, SysML can be more suitable than OPM for modeling the Autonomous ship systems.

4.3 Exploring the model’s utilization for analyzing autonomous ship systems.

Models can provide clear and attractive communication of a system’s specification and the interaction of systems in a module. Models using MBSE methods have been increasingly used in several engineering analysis as presented in Section 1.1. The following topics related to autonomous ship systems will be explored in the future for assessing the possibility of integrating models into the methods:

- **Hazard analysis**: For identifying and analyzing the hazards in the system, the analysts must understand the system composition and behavior. For this purpose, the system information generated by the models can be used to understand the autonomous ship systems before conducting hazard analysis. Instead of relying heavily on experts and their brainstorming sessions, these models can be potentially used as an input for experts. Furthermore, the experts who are less familiar with the system can use the models to gather necessary system information required for identifying hazards. The challenge is then to identify the most important diagrams to be communicated to the analysts as providing everything can lead to the problem of information overload.

- **Real-time system monitoring**: There is also a possibility to use the models as an interface for real-time system monitoring for autonomous ships. A simple example would be by adding a component status property to each of the components block in SysML where they are linked to the sensors installed on the ship. If the sensor is not sending any signals, the model will notify the operator by changing the status from “operating” to “non-operating”. In addition, it will display all other components and activities that are affected by this status change and the safety measures that need to be followed as predefined by the system’s designers. Thus, the possibility of using SysML diagrams for real-time system monitoring of autonomous ship systems should be further explored in future.

- **System maintenance:** Similar to component status, the information about the expected life-time of each component and the guidelines for executing the maintenance from the manufacturers can be also added to the diagrams and communicated to the maintenance engineer. Furthermore, a property can be added in the blocks of the components which records the date of previous maintenance. This will allow engineers to keep track of the components and use the information stored in models to guide the maintenance process.

Most of the engineering or operating tasks that require detailed information about the system could be guided by these system models. However, the implementation of these methods is heavily reliant on the tools. Thus, the tools that are currently being used need to be updated and assured for conducting these analyses for autonomous ship systems.

5 CONCLUSIONS

Graphical models could unlock various ways of communicating system information to relevant stakeholders. Thus, system modeling methods could be a viable option for autonomous ship systems. This paper explored OPM and SysML for handling the key issues related to complex systems i.e. autonomous ship systems. These key issues are complexity management, system understandability and communication of the system information. The methods were applied for the case of a typical Class II DP-system, that is believed to have a major role in autonomous ship operations. The in-depth comparison shows that both methods have been able to manage the system complexity and communication of system information using graphical illustrations or models.

Although, the complexity management and communication are covered well by both methods, SysML is more effective than OPM in handling the system understandability. As SysML provides different diagrams that provide system information from different perspective, it is easier to understand the system structure and behavior in all system hierarchical level. Furthermore, SysML consists of requirements diagrams and parametric diagram for supporting the requirements analysis and other engineering analysis (i.e. sensitivity analysis, performance analysis and design optimization), which is lacking in OPM. Thus, SysML can be more suitable than OPM for modeling the autonomous ship systems. However, OPM models are easier and faster to create and understand than SysML and should be implemented in those cases with limited resource availability and where the level of detailed information provided by OPM models are sufficient for the analysts. Furthermore, adding the distinct features of OPM such as simulation and automatic text generation to the SysML should be explored in the future.

The discussions and conclusions of this research are based on the comparison of DP-system models generated using the system operation manual. For improving the creditability for the research, these models and the results will be assessed using expert’s opinion in the future.

ACKNOWLEDGEMENTS

This research has been funded by RESET project. RESET has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 730888.

REFERENCES

Best, B., Jurjens, J., & Nuseibeh, B. (2007). Model-Based Security Engineering of Distributed Information Systems Using UMLsec. *29th International Conference on Software Engineering (ICSE'07)*, (pp. 581-590). Minneapolis. doi: 10.1109/ICSE.2007.55

D'Ambrosio, J., & Soremekun, G. (2017). Systems engineering challenges and MBSE opportunities for automotive system design. *International Conference on Systems, Man, and Cybernetics*, (pp. 2075-2080). Banff.

DNV GL (2013). Rules for Classification and Construction. Germanischer Lloyd SE, Hamburg. Retreived from: http://rules.dnvgl.com/docs/pdf/gl/maritimerules/gl_i-1-15_e.pdf

Dori, D. (1995). Object-process Analysis: Maintaining the Balance Between System Structure and Behaviour. Journal of Logic and Computation, 5(2), 227-249. Available at:
https://doi.org/10.1093/logcom/5.2.227Dori, D. (2002). *Object-Process Methodology- A Holistic Systems Paradigm.* Berlin: Springer-Verlag .

Dori, D. (2016). *Model-Based Systems Engineering with OPM and SysML.* Cambridge: Springer.

Enterprise Systems Modeling Laboratory. OPCAT - OPM tool. Available at: http://esml.iem.technion.ac.il/

Friedenthal, S., Griego, R., & Mark, S. (2007). INCOSE Model Based Systems Engineering (MBSE) Initiative. *INCOSE .* San Diego.

Friedenthal, S., Moore, A., & Steiner, R. (2015). Systems Engineering Overview. In *Practical Guide to SysML - The systems Modeling Language*.

Grobshtein, Y., Perelman, V., Safra, E., & Dori, D. (2007). Systems Modeling Languages: OPM Versus SysML. *International Conference on Systems Engineering and Modeling* , (pp. 102-109). Haifa. doi: 10.1109/ICSEM.2007.373339

Hollnagel, E. (2012). *FRAM: The Functional Resonance Analysis Method.* Ashgate Publishing Limited.

Holt, J., & Perry, S. (2019). In *SysML for Systems Engineering - A Model-Based Approach (3rd Edition)* (p. 3). London: The Institution of Engineering and Technology.

International Maritime Organization. (1994). *Guidelines for vessels with Dynamic Positioning Systems.* Retrieved from http://www.imo.org/blast/blastDataHelper.asp?data_id=10015&filename=MSCcirc645.pdf

International Standards Organization. (2015). *ISO/PAS 19450:2015 - Automation Systems and Integration -- Object-Process Methodology.* Retrieved from https://www.iso.org/standard/62274.html

International Standards Organization. (2017). *ISO/IEC 19514:2017- Informational Technology -- Object Management Group Systems Modelling Languuage (OMG SysML).* Retrieved from https://www.iso.org/standard/65231.html

Kongsberg (2014). *Kongsberg K-Pos DP (OS) Dynamic positioning system with Offshore Loading Application.* Kongsberg Maritime AS.

Levander, O. (2017). *Autonomous ships on the high seas. IEEE, 54*(2), 26-31.

Leveson, N. G., & Thomas, J. P. (2018). *STPA HANDBOOK.* Retrieved from https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf

Mhenni, F.¨, Nguyen, N., Kadima, H., & Choley, J.-Y. (2013). Safety Analysis Integration in a SysML-Based Complex System Design Process. *Systems Conference (SysCon).*

MUNIN. (2016). *Research in Maritime Autonomous Systems Project Results and Technology Potentials.* Retrieved from, http://www.unmanned-ship.org/munin/wp-content/uploads/2016/02/MUNIN-final-brochure.pdf

Object Management Group. (2017). OMG Systems Modeling Language. Retrieved from https://sysml.org/.res/docs/specs/OMGSysML-v1.5-17-05-01.pdf

Russell, M. (2012). Using MBSE to Enhance System Design Decision Making. *Procedia Computer Science* (pp. 188-193). Elsevier.

Visual Paradigm 15.2. SysML tool. Available at: https://www.visual-paradigm.com/

Weck, D., Roos, O. L., Magee, D., & L., C. (2011). Modeling and Analyzing Engineering Systems. In *Engineering system - Meeting Human Needs in a Complex Technological World.* MIT Press.