2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

# Semantically enhanced system and automation design of complex marine vessels

Nikos Kougiatsos, Jesper Zwaginga, Jeroen Pruyn and Vasso Reppa
*Maritime and Transport Technology Department*
*Delft University of Technology*
Delft, the Netherlands
{n.kougiatsos; jesper.zwaginga; j.f.j.pruyn; v.reppa}@tudelft.nl

***Abstract*—To integrate and assist the system and automation design phases of complex marine vessels, this paper proposes a two-level semantically enhanced scheme. At the design level, the system components are described and automatically connected by a developed graph-making tool using semantic "knowledge". Decisions regarding the system selection are made based on certain Quality of Service Criteria (QoS) and enforced in the final semantic database using a dedicated cognitive agent. The automation level leverages the selected systems semantic information with that of the associated automation components and reuses the graph-making tool to update the connection graph. The resulting knowledge-graph is then used to "reason" for the creation of feasible closed-loop control architectures while a cognitive agent determines which closed-loop architecture to use based on various QoS criteria. The chosen closed-loop architecture can then change in an online manner during the vessel operation in case that system reconfiguration is required either due to malfunctioning components, or aiming to satisfy mission's goals. The applicability and efficiency of the proposed method are shown using a case study for marine propulsion.**

***Index Terms*—semantic knowledge models, cognitive architecture, computer-supported coordination, network modelling, control systems, marine systems**

## I. Introduction

Nowadays, marine vessels' operation is characterized by continuously growing complexity due to the highly demanding and safety-critical tasks that they need to perform. Meanwhile, demands from the International Maritime Organisation for vessel emission reduction [1] are currently shifting the vessel owners' focus to alternative fuels and exhaust gas treatment technologies. This results in more complex designs whose components are often sourced from different manufacturers (e.g., MAN, ABB), are equipped with their own automation components (controllers, sensors) and use different protocols and naming conventions. It is the task of the designer (marine engineer) to select the components, manually create the connection graph based on their expertise (knowledge graph), reiterate the design until specifications are met (e.g., attain a speed of 12 knots, satisfy emission regulations) and then design the control system in such a way that all automation components can function together smoothly and effectively (interoperability).

However, the future of the maritime sector regarding what alternative fuels will be used is still greatly uncertain. At the same time, the vision for autonomous vessels has begun materializing in recent years. The increase in the required cyber-devices, communication and coordination algorithms as well as more sophisticated control architectures are just the start of the new Internet of Ships (IoS) era [2]. The selection of vessel components, their interconnections, and the corresponding automation are thus difficult decisions to be made by the designer, since any future modification (e.g. conversion to another fuel, component maintenance, re-tuning a controller) will be associated with considerable labor time and cost.

To deal with complexity during design, methods like "Point-based design" start at the requirements definition and subdivide the design into increasingly detailed subdivisions [3]. However, this makes it difficult to adapt systems when requirements change without restarting the process [4]. Instead, systems engineering enables better tractability of requirements and allows investigating the impact of changes on the systems levels [5]. The impact of changes in the operating environment, like surviving damage, are commonly researched [6]. However, changes to the design are dealt with as a separate engineering activity during the lifetime. Integrating new systems during the re-configuration of an existing vessel architecture to meet updated requirements can be expensive and time-consuming [7]. Therefore, several authors have proposed design preparations to support reconfiguration by implementing attributes that facilitate system changes into the design [8].

Even though the investment can be large [9], it has been shown that preparation performs well in uncertain scenarios through model-based case studies [10]. However, in these abstract models, no connection is made to what triggers a reconfiguration and what should be done to meet the trigger requirements in reality. Therefore, while such models are necessary to support investments, human decision-making during the operation should also be considered. This could be done by adding a human-in-the-loop [11]. Especially as fleet renewal timing is still separated from the design process [12], the execution of a reconfiguration strategy during the operation is left to operators (captain, vessel engineers) without clear guidelines from the design. For instance, the vessel operators will eventually decide if component maintenance is necessary based on their experience and sensor information, in contrast to automotive applications where a service schedule is prescribed by the design. This lack of communication between designers and operators can result in design modifications not being executed at the right time, as reconfiguration thresholds are unclear.

From the control aspect, once the system is slightly modified, closed-loop design will be redone from scratch due to its centralized structure. The operators, on the other hand, rely on the system feedback (e.g. speed, course, engine torque information from sensors) to make decisions. Thus, reliable and continuous communication of this information is essential to prevent flawed decision-making. The authors of [13], [14] both describe the use of digital twin technology for fault diagnosis and accommodation of marine propulsion plants using neural networks. Nevertheless, no discussion is made on the semantics aspect that will enable such technologies to be applied. Then, in [15], the authors propose a co-simulation platform that will enable fast and reliable testing and optimization of vessel system and automation designs before construction, and that can also be used for training purposes of crews during the vessel's life-cycle. However, both papers are using a fixed component library to choose from, and connections to closed-loop control systems are proven difficult to model. In addition, the occurrence of faults during the operational phase of the vessel is not explored as a change mechanism for the installed system configuration.

In previous work [16], [17] the impact of uncertainty when selecting modifications or a starting design was investigated through the use of exploratory methods dealing with deep uncertainty, stochastic and robust optimization approaches. However, both only considered simulated scenarios, not taking the exact reasons for changes in the design or specific system modifications into account. In addition, in [18], [19], the authors proposed a switching logic between hardware and virtual sensors and between different energy management controllers. However, the implementation aspects of these modular architectures were not explored.

This paper proposes a two-level semantically-enhanced architecture aiming to assist the design and operation of marine vessels while addressing current and future challenges of environmental changes and design uncertainty in the maritime field. The key aspects of the proposed architecture are the semantic database of components (see Section II) and the cognitive agent architecture (see Section III). An automated tool for connecting system/ automation components based on their semantic description, thus creating the knowledge graph, reduces the active workload of the designer. The design of the architecture takes into account the human-in-the-loop factor by providing the designers or vessel operators with tools for better decision-making. Moreover, the use of multiple cognitive agents is aimed to enforce the design/operation decisions by making use of semantic information, the knowledge graph, and certain Quality of Service (QoS) criteria. In the first level of the architecture, these criteria are related to the uncertain transition path between different fuel options while in the second level, these criteria are also tied to safe operation and performance. A practical use-case of the proposed cognitive architecture for marine propulsion is provided in Section IV, followed by concluding remarks in Section V.

Considering the application, the main contribution of this paper is the creation of a semantic database that enables system and automation design revisions during the life-cycle of the vessel, providing a sustainable engineering solution to efficiently manage and process the uncertainties in design decisions. The database is further enhanced with an automated knowledge graph tool that can produce the connection graph between components based on their semantic description. Besides, the semantic description of components mitigates the compatibility issues between parts of different manufacturers. The addition of the cognitive architecture then provides effective decision support to the human-in-the-loop element. From the methodology point of view [20], the occurrence of unexpected events (e.g. sensor faults) whose diagnosis can lead to online reconfiguration of the control architecture using the semantic database is considered. The semantic representation of automation components is adapted to marine vessels and includes virtual sensors and monitoring agents (see Section II), in addition to control components [20].

## II. Semantic Database of vessel components

In this paper, we are discussing the use of a cognitive architecture that will support the decision process of vessel designers and operators. The core of this method consists of a semantic database that can be seen in Fig.1. The semantic database consists of a Components Database ($F$) where semantic information about available components are stored (e.g. input, output, type), the knowledge graph where the connections between components are visualized, and multiple Quality of Service (QoS) criteria that are used for assessment purposes. The components database is enriched by semantic information provided either by the designer (e.g., the electric motor requires electric power from the generators to produce mechanical torque) or system manufacturers (e.g., operational maps). Finally , in this paper, an automated knowledge graph (G) tool is proposed to assist the design process.

### A. Components database ($\mathcal{F}$)

In [20] a components database was created to facilitate the reconfigurability of temperature control in smart buildings. Motivated by this work, a component database is designed considering the high complexity of the vessel system. Particularly, to be able to represent the large number of physical components in marine vessels and their complex interconnections without exponentially increasing computation time, the physical component description (e.g. Systems) is simplified. Moreover, semantic information about the grouping of hardware sensors for condition monitoring purposes is also used. The database is thus further enhanced with the following component descriptions:

*1) "System":* Systems $\Sigma^{(I)}, I = 1, \cdots, n_I$ each have necessary input and output mediums. Therefore, system components are added to the database including input and output information for the specific medium (e.g. water, air etc.). For instance, the fuel pump(s), electric motors, internal combustion

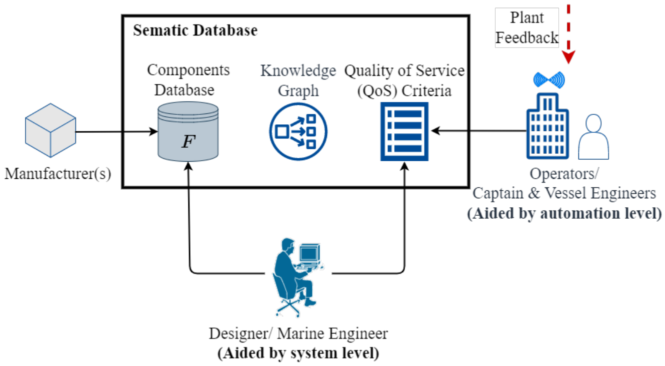


Fig. 1. Different actors' role in the design of the semantic database

engines, batteries, propellers found inside marine vessels can be considered as system components.

*2) "Monitoring agent":* The monitoring agent $\mathcal{M}^{(I)}$ is used to oversee the health of sensors $\mathcal{S}^{(I)}$ belonging to system $\Sigma^{(I)}$, $I = 1, \cdots, n_I$. Due to the complexity associated with marine systems, each "monitoring agent" is typically composed of one or more "monitoring modules" $M^{(I,q)}$, $q = 1, \cdots, q_I$. The decision vector resulting from this comparison is then compared to certain binary sensor fault signature matrices in two levels of isolation, as already described in [21]. The result of the diagnosis is a mapping $R^{(I)} \to \mathcal{S}_F^{(I)}$ with $R^{(I)} = r^{(I)}$ denoting the set of residuals and $\mathcal{S}_F^{(I)}$ denoting the faulty sensor set, as a result of the diagnosis process.

*3) "Virtual sensor":* Each "virtual sensor" instance leverages the analytical redundancy of the system in order to create virtual and fault-free measurements and is part of a "monitoring agent". It is activated after the detection and isolation of sensor faults by the respective "monitoring module", thus increasing computational effectiveness. A "virtual sensor" is described by the equation

$$\begin{aligned}\hat{x}^{(I)}(k) =& f_v^{(I)}(\hat{x}^{(I)}[k-1], y^{(I)}[k], u^{(I)}[k], \\ & \hat{x}^{(I)}[k]; \zeta_s^{(I)}, \mathcal{S}_F^{(I)}),\end{aligned} \tag{1}$$

where $\zeta_s^{(I)}$ denotes the design parameters of the virtual sensor.

In previous work [18], three types of "virtual sensors" have been defined for Differential-Algebraic systems and may be used under this module label; dynamic virtual sensors, static virtual sensors, and Set Inversion via Interval Analysis (SIVIA)-based "virtual sensors".

The previously described system and automation modules can be considered as being elements of the sets $\mathcal{F}_p$ and $\mathcal{F}_\alpha$ respectively, with:

$$\mathcal{F}_\alpha = \mathcal{F}_a \cup \mathcal{F}_c \cup \mathcal{F}s \cup \mathcal{F}e \cup \mathcal{F}y \cup \mathcal{F}u \cup \mathcal{F}m \cup \mathcal{F}v, \tag{2}$$

where $\mathcal{F}_a$, $\mathcal{F}_c$, $\mathcal{F}_s$, $\mathcal{F}_e$, $\mathcal{F}y$, $\mathcal{F}_u$ denote the set of "actuators", "controllers", "sensors", "state-estimators", "pre-control functions" and "post-control functions" respectively. The novelty of the present paper regarding the semantic module database resides in a richer description of the "plant" and its associated set $\mathcal{F}p$ and the addition of module sets for "monitoring agents" and "virtual sensors" denoted as $\mathcal{F}m$, $\mathcal{F}v$ respectively. From the description of the modules belonging to each set, it can be seen that modules of different sets can be connected with each other by Input/Output coupling. This feature is exploited by the Knowledge graph tool presented next. Finally, the components database is defined as:

$$\mathcal{F} = \mathcal{F}_p^{(s)} \cup \mathcal{F}_\alpha, \tag{3}$$

where $\mathcal{F}_p^{(s)} \subseteq \mathcal{F}_p$ denotes the selected systems set and $\mathcal{F}_\alpha$ denotes the automation set, already defined in (2).

### *B. Knowledge-graph (G)*

Having described the various plant and automation components for the semantic database, the knowledge graph is created automatically using the semantic information of components, in Algorithm 1. As a result, both the system designer and the operator roles are assisted by this "smart" feature. The "knowledge graph" tool is designed with two purposes in mind; **(i)** to help the designer illustrate the interconnections between the physical system components in the design level and, **(ii)** to enable the synthesis of feasible closed-loop architectures, also known as the process of "semantic matching" [20], by combining the automation ($\mathcal{F}_\alpha$) with the selected physical plant ($\mathcal{F}_p^{(s)}$) components, again illustrating the interconnections, in the automation level. As a result, the "knowledge graph" is created with the following features; **(1)** vertices ($V$): The entries of the semantic database, **(2)** edges ($E$): The connections between vertices, **(3)** mediums ($\Upsilon$): The information carried by the connection.

The knowledge graph tool is built in two levels; the design and the automation level. In the design level, the knowledge graph algorithm 1 begins with a list of systems that are considered by the designer for the installation. The algorithm then starts for instance from a propeller (see vertex $v_1$ in line 2) and connects the system components (vertices $v_2$) that are necessary for it to be operational in lines 3-10. When multiple systems have the same input or output, graphs are duplicated. The procedure is run for all the considered systems and results in a knowledge graph (line 11) corresponding to the physical plant connections. A decision is then made on which systems will be installed using the cognitive architecture that will be described in the next Section III. Thus, the set $\mathcal{F}_p^{(s)}$ is obtained (line 12). In order to make the chosen configuration operational, the addition of automation components is necessary. In the automation level, the algorithm stars by connecting the hardware automation components (e.g. sensors, actuators) and the relevant controllers, as shown in lines 13-14. Moreover, the information about the grouping of hardware sensors is used to generate the "monitoring agents" (belonging to set $\mathcal{F}_m$) and their connections in lines 15-24. A "monitoring agent" requires the output of relevant hardware sensors and controller(s) to provide decisions on the occurrence of faults. Moreover, its

[0]The expression $a.b$ in Algorithm 1 denotes the use of attribute $b$ of the set of attributes comprising object $a$. For instance, a vertex $a$ can have attributes $b \in \{input, output, \cdots\}$ that correspond to the semantic information carried by vertex $a$.

**Algorithm 1** Multi-level Knowledge graph tool

**Input:** $\mathcal{F}_p, \mathcal{F}_\alpha - (\mathcal{F}_m \cup \mathcal{F}_v), \mathcal{F}_v$ ▷ Databases
**Output:** $G$ ▷ Knowledge Graph

1: $V \leftarrow \mathcal{F}_p$; $E \leftarrow \emptyset$; $\Upsilon \leftarrow \emptyset$ ▷ **Design level**
2: **for** $v_1$ in V **do** ▷ Physical plant connections
3:     **for** $v_2$ in V **do**
4:         $y \leftarrow v_2.output \cap v_1.input^1$ ▷ $y$: medium
5:         **if** $y \neq \emptyset$ **then** ▷ Components can be connected
6:             $E \leftarrow E \cup \{v_2, v_1\}$
7:             $\Upsilon \leftarrow \Upsilon \cup \{y\}$
8:         **end if**
9:     **end for**
10: **end for**
11: $G \leftarrow \{V, E, \Upsilon\}$ ▷ Physical plant knowledge graph
12: **Database update module**: $\mathcal{F}_p \mapsto \mathcal{F}_p^{(s)}$ ▷ Eq.(4)
13: $V \leftarrow \mathcal{F}_p^{(s)} \cup \{\mathcal{F}_\alpha - (\mathcal{F}_m \cup \mathcal{F}_v)\}$ ▷ **Automation level**
14: **Execute lines 2-10** ▷ Hardware automation connections
15: $sg \leftarrow \mathcal{F}.sensor\_groups$ ▷ Sensor grouping information
16: **for** i=1:length(sg) **do** ▷ Monitoring agents generation
17:     $V \leftarrow V \cup \{M\_i\}$
18:     $S \leftarrow \{s \in \mathcal{F}_s \cap sg[i]\}$ ▷ Connect sensors
19:     $C \leftarrow \{c \in \mathcal{F}_c \cap S.edges\}$ ▷ Connect controllers
20:     $E \leftarrow E \cup \{\{S, M\_i\}, \ \{C, M\_i\}\}$ ▷ Update edges
21:     $\Upsilon \leftarrow \Upsilon \cup \{S.output, \ C.output\}$ ▷ Update mediums
22:     $E \leftarrow E \cup \{M\_i, C\}$ ▷ Update edges
23:     $\Upsilon \leftarrow \Upsilon \cup \{M\_i.output\}$ ▷ Update mediums
24: **end for**
25: $V \leftarrow \mathcal{F}_p^{(s)} \cup \mathcal{F}_\alpha$ ▷ Virtual sensors addition
26: **Execute lines 2-10** ▷ Virtual sensors connections
27: $G \leftarrow \{V, E, \Upsilon\}$ ▷ Cyber-Physical knowledge graph

output (fault decision) can be used as input to the controller(s) it is associated with in a fault-tolerant control scheme [18]. The "virtual sensors" are then connected in lines 25-26. Finally, the complete cyber-physical knowledge graph is generated based on the prescribed vertices, edges, and mediums, as shown in line 27.

### C. Quality of Service criteria

The Quality of Service (QoS) criteria for the target architecture are classified as design and automation criteria. The design criteria are related to the available systems, the necessary interconnections and subsystems, and the operational requirements. By visualizing the interconnections and additional systems for different selections, the designer gains insight into the consequences of each selection. Physical plant knowledge graphs can be ranked by using operational criteria such as available mass and volume thresholds, or energy capacity targets. After a selection, separate systems within the graph can be replaced to test improvements in system characteristics.

The automation criteria are related to the vessel systems' performance and safety. For instance, the components belonging to the hardware ($\mathcal{F}_s$) and the virtual ($\mathcal{F}_v$) sensor set are to be used interchangeably by the system when one or more hardware sensors fail during operation. Considering control system performance, switching to the sensor with the minimum reference tracking error is preferable. However, certain types of virtual sensors require a long time for convergence, so choosing a sensor with a higher convergence rate but moderate reference tracking error to avoid danger is also a reasonable option. The time to switch is also taken into consideration as multiple consecutive switches might compromise control stability. The aforementioned criteria have already been explained in detail in [16]–[18].

The next step is to design a suitable cognitive architecture that can assist human decisions regarding the system, automation design, and operation phases of marine vessels by utilizing the semantic database presented before.

## III. Multi-level Cognitive Architecture

In this work, the marine vessel system and automation design processes are integrated using a cognitive architecture consisting of two levels; design and automation, as shown in Figure 2. As previously discussed, each component in the semantic database can be categorized into "sets" of similar components. The design level determines the system selection ($\mathcal{F}_p^{(s)}$) from the physical plant "set" ($\mathcal{F}_p$) based on the design criteria. The automation component sets ($\mathcal{F}_\alpha$) are then added to $\mathcal{F}_p^{(s)}$ in the automation level. The various "sets", in both levels, are processed internally and in correlation to each other by the "knowledge graph" in order to determine their physical or cyber interconnections. The "sets" can be thus linked to each other to form a "smart" vessel system feedback control and monitoring scheme, as shown in Figure 2.

### A. Design level

At the design level, the semantic information of "systems" is used to form clusters of systems. The clusters considered for the vessel design receive an index value (e.g. 1,2,$\cdots$) and are then assessed based on the design criteria. This process results in a design decision $\sigma_d \in \mathbb{Z}_+$ regarding the systems that will be considered in this iteration of the design. The physical plant database $\mathcal{F}_p$ is consequently filtered and the chosen configuration's semantic information, denoted as $\mathcal{F}_p^{(s)}$, is communicated to the automation level. The database update module can thus be defined using the mapping operator:

$$f_d : \Sigma_d \times \mathcal{F}_p \mapsto \mathcal{F}_p^{(s)}, \tag{4}$$

where $\Sigma_d$ is the space of configuration decisions $\sigma_d$, $\mathcal{F}_p$ is the physical plant database and $\mathcal{F}_p^{(s)} \subseteq \mathcal{F}_p$ is the chosen part of the physical plant database space for the current configuration.

### B. Automation level

The automation level employs the full component database $\mathcal{F}$, defined in (3), formed from the semantic information about the chosen configuration's plant components $\mathcal{F}_p^{(s)}$ with the addition of related automation components $\mathcal{F}_\alpha$. The knowledge graph is then used to "match" components in feasible closed-loop architectures, in a process called "semantic matching".

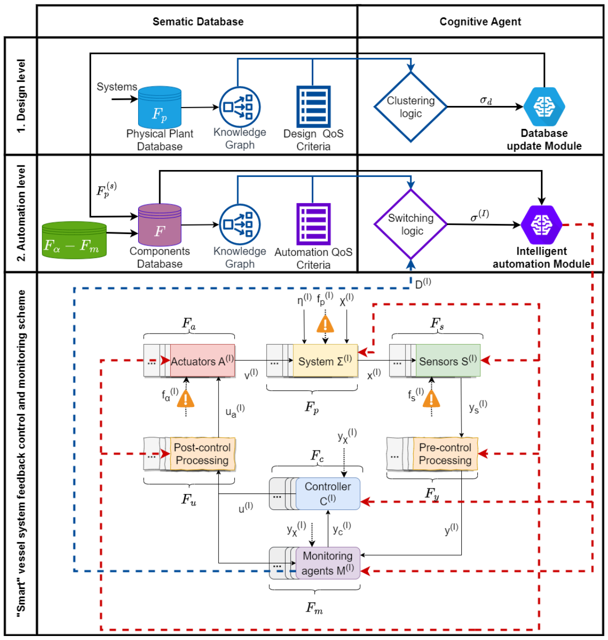


Fig. 2. Multi-level cognitive architecture for intelligent marine vessels. At the design level, the cognitive agent determines the system selection $F_p^{(s)}$ from a physical plant database $F_p$ of potential options based on design criteria. In the automation level, the cognitive agent determines the closed-loop configuration to be used based on the automation criteria and the monitoring agents decision $D^{(I)}$ regarding the occurrence of faults ($f_a^{(I)}$, $f_p^{(I)}$, $f_s^{(I)}$).

This information in conjunction with the automation criteria is used by the switching logic to "reason" about which feasible closed loop architecture will be rendered active, in a process called "semantic reasoning" [20]. The switching logic is an online feature, taking into consideration that the decisions of the various "monitoring agents" also affect the choice of closed-loop configuration during operation. In case of sensor faults ($f_s^{(I)}$) affecting one or more sensors of the plant ($I$), an efficient logic has been proposed in [18]. The output of the switching logic is in general a switching vector signal:

$$\sigma^{(I)} = [\sigma_p^{(I)}\ \sigma_a^{(I)}\ \sigma_s^{(I)}\ \sigma_c^{(I)}\ \sigma_y^{(I)}\ \sigma_u^{(I)}\ \sigma_m^{(I)}\ \sigma_v^{(I)}]^\top, \quad (5)$$

where $\sigma_p^{(I)}$, $\sigma_a^{(I)}$, $\sigma_s^{(I)}$, $\sigma_c^{(I)}$ $\sigma_y^{(I)}$, $\sigma_u^{(I)}$, $\sigma_m^{(I)}$, $\sigma_v^{(I)}$ denote the indices of the modules that are required from the sets $\mathcal{F}_p$, $\mathcal{F}_a$, $\mathcal{F}_s$, $\mathcal{F}_c$, $\mathcal{F}_y$, $\mathcal{F}_u$, $\mathcal{F}_m$ and $\mathcal{F}_v$. The intelligent automation module can then be defined using the mapping operator

$$f : \Sigma \times \mathcal{F} \mapsto \mathcal{I}, \quad (6)$$

where $\Sigma$ is the space of configuration decisions $\sigma^{(I)}$, $\mathcal{F}_p^{(s)}$ was defined in (4), $\mathcal{F}$ was defined in (3) and $\mathcal{I}$ is the space of active configurations of the feedback control scheme.

## IV. Marine Propulsion Use-Case

Out of all systems involved in marine applications, the propulsion system is considered to be the most safety-critical [21]. At the same time, it's associated with high complexity in system interconnections and due to continuously stricter emission-control regulations, the modifications in this system

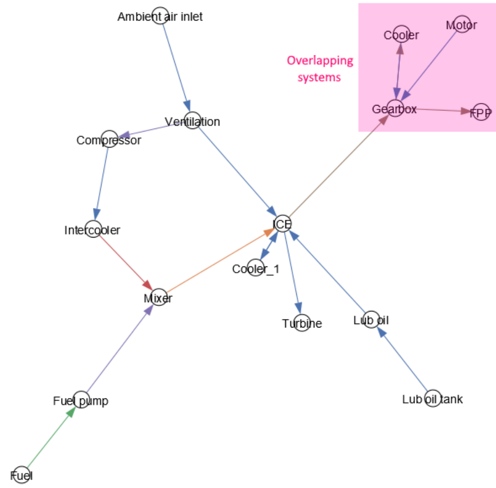


Fig. 3. Knowledge graph for diesel-electric propulsion. Each color of the edges represents a different medium used. The semantic description of the overlapping plant components (electrical propulsion cluster) shown in the shaded area is given in Table I

are expected to be the most frequent and uncertain. In this section, the cognitive architecture described in Section III is applied to a database of systems concerning marine propulsion, modelled as discussed in Section II.

The design target for the marine vessel in this scenario is to be able to convert between diesel-electric ($\sigma_d = 1$) to methanol-electric propulsion ($\sigma_d = 2$). The database is thus comprised of two parts, the changeable mechanical part (diesel/methanol) and the overlapping electrical part, whose systems' semantic description is provided in Table I. Taking into consideration safety, space, and methanol toxicity concerns, already elaborated on in [16], [17], the design decision given as the output of the design level is $\sigma_d = 1$, indicating the use of a diesel-electric configuration for the initial design. Using the automated graph-making tool described in Algorithm 1, the system connections for the diesel-electric installation are shown in Figure 3. The application of this tool was done using the Python Igraph package and Python 3.9.

TABLE I
Semantic Information for Overlapping Systems Involved in Target Propulsion Architectures

| System | Inputs | Outputs |
|---|---|---|
| Gearbox | Cool air<br>Motor Power<br>Engine Power | Hot air<br>Propeller Power |
| Fixed Pitch Propeller | Propeller Power | Thrust |
| Electric Motor | Voltage | Motor Power |
| Cooler | Hot air | Cool air |

Having connected the physical plant components, the semantic database proceeds to acquire information about the automation components. The knowledge graph tool updates the physical plant knowledge graph using Algorithm 1, by connecting the hardware (sensors, actuators) and cyber automation components (cotrollers, monitoring agents, virtual sensors,

TABLE II
EXCERPT OF SEMANTIC INFORMATION FOR AUTOMATION COMPONENTS

| Component | Inputs | Outputs | Sensor set | Units |
|---|---|---|---|---|
| Speed controller | Shaft speed | Injected Fuel | - | - |
| Shaft speed sensor | Thrust | Shaft speed | $S^{(1,2)}$ | rps |
| Motor torque sensor | Motor Power | Motor Torque | $S^{(2,1)}$ | Nm |
| Torque controller | Motor Torque | Voltage | - | - |
| Shaft speed reference | - | Shaft speed | - | rps |
| Motor torque reference | - | Motor Torque | - | Nm |

TABLE III
FEASIBLE CLOSED LOOP CONFIGURATIONS USING COMBINATIONS OF HARDWARE AND VIRTUAL SENSORS FOR CONTROL FEEDBACK. MORE INFORMATION ON VIRTUAL SENSORS CAN BE FOUND IN [18]

| Configuration ID | **ID01** | **ID02** | **ID03** |
|---|---|---|---|
| **Speed Feedback** | Hardware | Hardware | Virtual (dynamic) |
| **Torque Feedback** | Hardware | Virtual (static) | Hardware |
| Configuration ID | **ID04** | **ID05** | **ID06** |
| **Speed Feedback** | Virtual (dynamic) | Virtual (static) | Virtual (static) |
| **Torque Feedback** | Virtual (static) | Hardware | Virtual (static) |

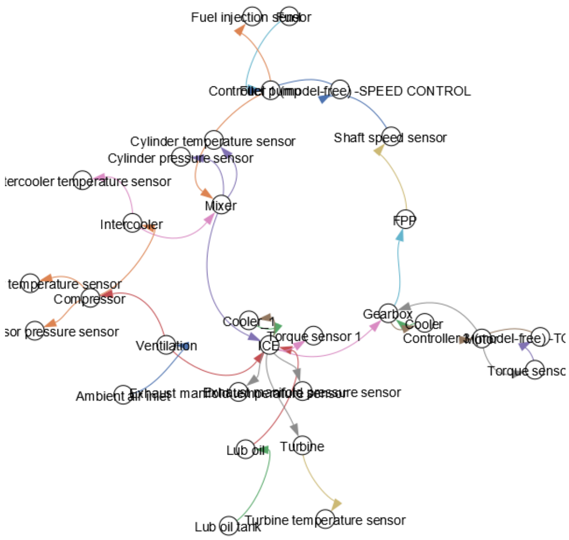


Fig. 4. Addition of hardware automation components in the knowledge graph. An excerpt of the semantic description of the additional hardware automation components can be found in Table II.

reference signals), resulting in the cyber-physical knowledge graph. An excerpt of the semantic information regarding the hardware automation components is shown in Table II while the updated graph containing the vertices corresponding to the hardware automation components is shown in Figure 4. At the end of this process, the knowledge graph is composed of 50 vertices and 154 connection edges. After acquiring the complete automation graph, the process of "semantic matching" forms the feasible closed loops through simple traversal of the graph. In this case study, the resulting loops differ on whether they use hardware or virtual sensors for feedback and condition monitoring. For brevity purposes of subsequent analysis, it is assumed that only hardware sensors are used for monitoring purposes and as input to virtual sensors while control feedback can stem both from hardware and virtual sensors. Under this assumption, the cognitive agent reasons for the feasibility of 6 closed loop configurations. The details of the closed-loop feedback control architectures are given in Table III. In this paper, the case study presented in [18] is reused to illustrate the use of the cognitive architecture. The control objective is for the vessel to achieve a reference power profile with a magnitude of $P_D = 9400$ kW. The system is simulated with two permanent abrupt offset sensor faults affecting the shaft speed sensor at 20 sec and the electric motor torque sensor at 50 sec. The initial conditions for the simulation, power split strategy, parameters, switching criteria, and switching logic have been previously described in [18] and will thus be omitted. The decisions of the monitoring agents regarding the occurrence of sensor faults are shown in Figure 5 while the closed-loop configurations implementations by the intelligent automation module can be seen in Figure 6. As observed in Figure 5, the sensor fault in the shaft speed sensor is diagnosed at $t = t_1 = 20$ sec (decision receives the value 1) while the decision regarding the diagnosis of faults in the motor torque sensor remains 0. Subsequently, the control configuration changes from 1 to 5 (a static virtual sensor is used for the shaft speed), as can be seen in Figure 6. Then, at $t = t_2 = 50$ sec, both decisions of the agents become 1 meaning that faults have been diagnosed in both the shaft speed and the motor torque hardware sensors. Thus, the control configuration changes from 5 to 6 (static virtual sensors are used for both the shaft speed and the motor torque), as presented in Figure 6. Based on the above results, the automated knowledge graph tool manages to effectively connect the physical plant, hardware, and cyber automation components using their semantically-enhanced description. The cognitive agent designed at the automation level managed to match the cyber-physical components to create the feasible closed-loop architectures, shown in Table III. Moreover, the monitoring agents' feedback regarding the occurrence of sensor faults has been used effectively to alter the configuration selection by the intelligent automation module.

## V. CONCLUDING REMARKS

This paper introduced a two-level semantically-based methodology aiming to assist and connect the system and automation design processes of marine vessels. To this end, the plant components, their connections and design specifications have been described using well established knowledge representation techniques and used to form the semantic database. The suggested semantic description of vessel plant and automation components was kept general in an effort to facilitate the integration of parts from different manufacturers, using different protocols and naming conventions into a unified inter-operable system. At the design level, the physical plant components database was filtered by using the stored semantic information to connect them, assessing the resulting knowledge graphs and selecting the physical plant components that will be used in the current installation. Then, at the automation

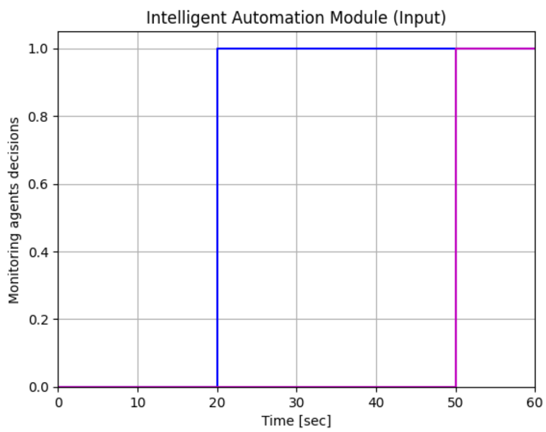


Fig. 5. Feedback of monitoring agents' decisions to the intelligent automation module for the specified sensor fault scenario (blue: monitoring agents' decision regarding the occurrence of faults in the shaft speed sensor, magenta: monitoring agents' decision regarding the occurrence of faults in the motor torque sensor). A value of '0' indicates that faults have not been diagnosed while a value of '1' indicates that faults were diagnosed.

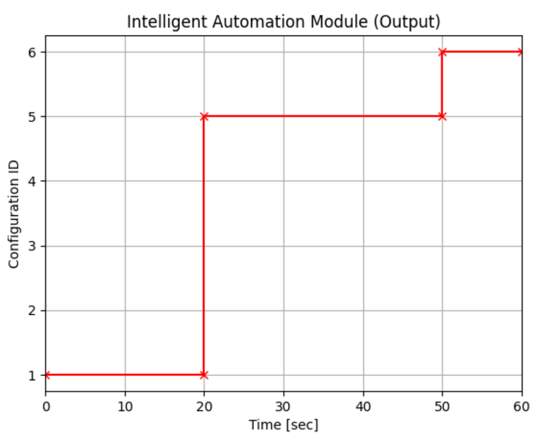


Fig. 6. Closed-loop configuration implementation by the intelligent automation module for the specified sensor fault scenario. The online switching mechanism is activated by the decisions of the monitoring agents regarding the occurrence of faults.

level, the database was appended with semantic information regarding automation components, the new component connections and operation specifications. A decision logic was then used to determine the automation components that would be used in the control configuration and the decisions were implemented using an intelligent automation module.

## Acknowledgment

This work has been funded by the project READINESS with project number TWM.BL.019.002 of the research programme "Topsector Water & Maritime: the Blue route" which is partly financed by the Dutch Research Council (NWO).

## References

[1] IMO, "Initial IMO Strategy on reduction of GHG emission from ships. Resolution MEPC.304(72) (adopted on 13 April 2018)," in *MEPC.304(72)*, 2018.

[2] S. Aslam, M. P. Michaelides, and H. Herodotou, "Internet of Ships: A Survey on Architectures, Emerging Applications, and Challenges," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9714–9727, 2020.

[3] J. H. Evans, "Basic design concepts," *Journal of the American Society for Naval Engineers*, vol. 71, no. 4, pp. 671–678, 1959.

[4] R. Pawling, V. Percival, and D. Andrews, "A study into the validity of the ship design spiral in early stage ship design," *Journal of Ship Production and Design*, vol. 33, no. 02, pp. 81–100, 2017.

[5] E. A. E. Duchateau and J. Hopman, "Interactive evolutionary concept exploration in preliminary ship design," Ph.D. dissertation, Delft University of Technology, 2016.

[6] T. A. Trapp *et al.*, "Shipboard integrated engineering plant survivable network optimization," Ph.D. dissertation, Massachusetts Institute of Technology, 2015.

[7] J. Coenen, V. Ruiz, J. M. Fernandez Hernando, K. Frouws *et al.*, "Design for retrofitting," in *DS 80-4 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 4: Design for X, Design to X, Milan, Italy, 27-30.07. 15*, 2015, pp. 061–070.

[8] C. F. Rehn, S. S. Pettersen, J. J. Garcia, P. O. Brett, S. O. Erikstad, B. E. Asbjørnslett, A. M. Ross, and D. H. Rhodes, "Quantification of changeability level for engineering systems," *Systems Engineering*, vol. 22, no. 1, pp. 80–94, 2019.

[9] E. Fricke and A. P. Schulz, "Design for changeability (dfc): Principles to enable changes in systems throughout their entire lifecycle," *Systems Engineering*, vol. 8, no. 4, 2005.

[10] M. Choi and S. O. Erikstad, "A module configuration and valuation model for operational flexibility in ship design using contract scenarios," *Ships and Offshore Structures*, vol. 12, no. 8, pp. 1127–1135, 2017.

[11] M. D. Curry and A. M. Ross, "Considerations for an extended framework for interactive epoch-era analysis," *Procedia Computer Science*, vol. 44, pp. 454–465, 2015.

[12] Ø. S. Patricksson, K. Fagerholt, and J. G. Rakke, "The fleet renewal problem with regional emission limitations: Case study from roll-on/roll-off shipping," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 346–358, 2015.

[13] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-Fault Detection, Isolation and Accommodation for Digital Twins via Modular Data-Driven Architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, 2021.

[14] S. Stoumpos, G. Theotokatos, C. Mavrelos, and E. Boulougouris, "Towards marine dual fuel engines digital twins-integrated modelling of thermodynamic processes and control system functions," *Journal of Marine Science and Engineering*, vol. 8, no. 3, 2020.

[15] S. Sadjina, L. T. Kyllingstad, M. Rindarøy, S. Skjong, V. Esøy, and E. Pedersen, "Distributed co-simulation of maritime systems and operations," *Journal of Offshore Mechanics and Arctic Engineering*, vol. 141, no. 1, 2019.

[16] J. J. Zwaginga and J. F. Pruyn, "An evaluation of suitable methods to deal with deep uncertainty caused by the energy transition in ship design," in *SNAME 14th International Marine Design Conference, IMDC 2022*. Vancouver, Canada: OnePetro, 2022, pp. 1–16.

[17] J. Zwaginga, B. Lagemann, E. S.O., and J. Pruyn, "Optimal ship fuel selection under life cycle uncertainty." Montreal, Canada: Elsevier, 2023.

[18] N. Kougiatsos, R. Negenborn, and V. Reppa, "A multi-sensory switching-stable architecture for distributed fault tolerant propulsion control of marine vessels," *Proceedings of the International Ship Control Systems Symposium*, vol. 16, 2022.

[19] M. van Benten, N. Kougiatsos, and V. Reppa, "Mission-oriented modular control of retrofittable marine power plants," *Proceedings of the International Ship Control Systems Symposium*, vol. 16, 2022.

[20] G. M. Milis, D. G. Eliades, C. G. Panayiotou, and M. M. Polycarpou, "A cognitive agent architecture for feedback control scheme design," *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, 2017.

[21] N. Kougiatsos, R. R. Negenborn, and V. Reppa, "Distributed model-based sensor fault diagnosis of marine fuel engines," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 347–353, 2022, 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022.