Implementation of an Operational Capability Model: Case Study on Offshore Support Vessel Fleet Management

Hadi Mazloum, Arthur Doliveira, Christophe Roman, Guillaume Graton, Mustapha Ouladsine

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

18th International Conference on Integrated Modeling and Analysis in Autonomous Control and Cognitive Agents (IMAACA 2025), held within the 22nd International Multidisciplinary Modeling & Simulation Multiconference (I3M 2025)

Implementation of an Operational Capability Model: Case Study on Offshore Support Vessel Fleet Management

Hadi Mazloum[a,*], Arthur Doliveira[a], Christophe Roman[a], Guillaume Graton[a,b], Mustapha Ouladsine[a]

[a]*Aix-Marseille University, LIS UMR CNRS 7020, Marseille, France*
[b]*Centrale Méditerranée, Technopôle de Château-Gombert, 38 rue Frédéric Joliot-Curie, F-13451 Marseille Cedex 13, France*

Abstract

This paper presents a numerical implementation of an operational capability model for selecting an Offshore Service Vessel (OSV) within a fleet, based on mission requirements. The initial model, grounded in dynamical systems theory, represents the system state as a labeled multigraph with attributes on its vertices. It is extended to weighted edges to provide a more pertinent representation of the OSV's operational state. The evolution of the system is formalized using algebraic operations, which were initially developed for simple graphs and are extended to multigraphs in the present paper, giving it a discrete nature. A UML diagram is proposed to ensure the reproducibility of the implementation.

Keywords: Numerical Modeling; Graph State Space; Labelled and Weighted Multi-Graph with Attribute.

1. Introduction

Money functions as a foundational instrument for facilitating exchange in modern economies. Prior to the advent of monetary systems, barter was the primary method of trade, requiring a mutual alignment of needs between parties—a process that was both inefficient and limited in scope. The introduction of money provided a universal medium of exchange and a standardized representation of value, thereby streamlining economic interactions. In contemporary societies, however, economic transactions extend beyond the mere exchange of currency for goods or services. They are embedded within legal and regulatory frameworks designed to ensure fairness, accountability, and long-term reliability. For instance, the purchase of a vehicle is not only a financial transaction but also a legally binding agreement, often involving warranties, quality standards, and compliance with safety regulations.

The following example illustrates the complexity of modern exchanges, particularly when they involve engineered systems. Let us consider a car which constitutes a complex assembly of interdependent subsystems—each with spe-

* Corresponding author. Tel.: +33-774-495-003.
E-mail address: hadi.mazloum@lis-lab.fr

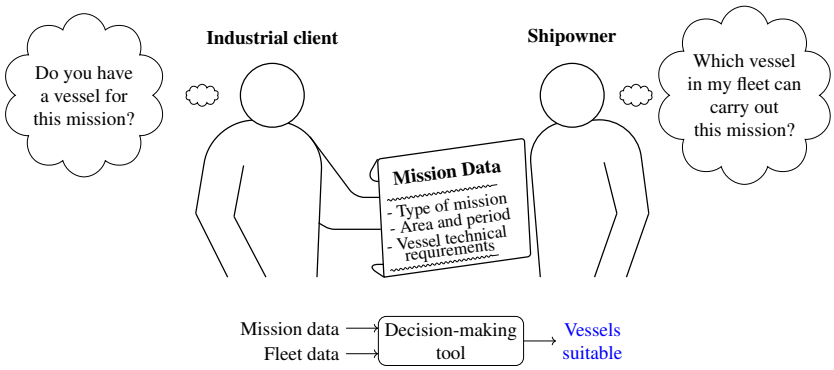

Figure 1: Client request to a shipowner for the acquisition of one of their OSVs, based on the data of the mission to be carried out.

cific functional roles. Buyers consider a range of parameters such as engine performance, seating capacity, tire specifications, and onboard technology. These elements must not only satisfy individual preferences but also conform to a coherent structural and functional design. A misconfiguration—such as storing the tires in the trunk rather than mounting them—makes the system non-operational, regardless of the monetary transaction. Thus, efficient exchange in modern contexts requires not only a shared valuation medium but also systemic consistency, regulatory clarity, and a shared understanding of functional integration. The question arise :

How can we formalize exchanges related to operational capability?

This question — central to this work — originates from a more specific yet equally complex scenario: the exchange of Offshore Support Vessels (OSVs). Without loss of generality, we consider the example illustrated in Figure 1: a client requiring a vessel (understood as a physical asset, including the operational status of its equipment, certifications, and crew) to carry out a defined set of tasks or missions. On the other side of the exchange is an operator or shipowner, who possesses a fleet of OSVs that can be modified, equipped, or maintained to meet the client's requirements. The objective is to formalize this interaction by capturing the constraints, capabilities, and decision-making processes involved in matching a client's operational needs with a suitable maritime asset. Offshore operations — including platform towing, subsea installation, equipment transport, and wind farm maintenance — demand a precise alignment between vessel capabilities and mission requirements. These missions are inherently complex, subject to safety-critical constraints, and often conducted under variable and uncertain environmental conditions. A central operational challenge can therefore be formulated as follows: given a specific mission profile, which vessel within the fleet is best suited to execute it? Conversely, given a vessel in a particular configuration and health state, is it capable of accomplishing a given mission at the initial time t_0, as well as over the time interval required by the mission?

1.1. Mathematical Formalism for Operational Capability Modeling

Recently, Doliveira et al. [1] proposed a generic model for assessing operational capability, formulated within the framework of dynamical systems theory and illustrated in Figure 2. The model is based on the triplet $\langle \mathcal{S}, \mathcal{M}, \rho \rangle$, where $\mathcal{S}$ denotes the current state of the system in the state space $\mathbb{G}$, $\mathcal{M}$ the assigned mission, and ρ the method used to evaluate the system's ability to accomplish it. The mission is formalized as the pair $\mathcal{M} = \langle [s^*], C_{\text{xt}} \rangle$, where $[s^*] \subset \mathbb{G}$ represents the mission objective and $C_{\text{xt}} \subset \mathbb{G}$ its context. The set $[s^*]$, the mission target, corresponds to an equivalence class of states in which $\mathcal{S}$ must lie to succeed. In the military context, where operational capability has been extensively studied for weapon systems in defensive scenarios [2, 3], $[s^*]$ corresponds to the well-known notion of *Required Operational Capabilities* (ROCs). An ROC translates a successful mission scenario into requirements the system state must satisfy [4]. When a system meets an ROC, it maximizes its success rate [5]. ROCs are generally estimated through expert judgment or simulation [5, 6]. The Doliveira model assumes that the ROC is known in advance and precisely defined.

The context C_{xt} is the set of elements that can influence the system state and cause it to enter or exit $[s^*]$. The method ρ evaluates operational capability by computing the distance between the system state and the target set $[s^*]$.

Since the state evolves under the influence of the context, this distance must be assessed continuously, as illustrated in Figure 2. A distinctive feature of Doliveira's model is that the state space consists of labeled and weighted multigraphs with attributes (see Figure 3 and Definition 1). Vertices, endowed with attributes, represent system components, and weighted edges represent their relationships. This representation allows modeling the vessel's complexity not only as a list of parameters but also accounting for its topological structure, i.e., the spatial organization of components and their interactions. Such an extension is common in military studies of operational capability [7, 8].

Two major implications follow from the fact that the state space $\mathbb{G}$ consists of labeled and weighted multigraphs with attributes. First, Doliveira et al. [1] establish ρ as a weak discrete distance [9], which assesses both topological similarity—whether the system state $\mathcal{S}$ has a structure comparable to the target $[s^*]$—and the equivalence of vertex attributes and edge weights, i.e., whether $\mathcal{S}$ satisfies the mission requirements. In [1], this distance ignores edge weights and is called discrete because it provides a binary yes/no evaluation. Second, the evolution of a system represented by a labeled and weighted multigraph affects topology (through addition or removal of vertices, edges, or subgraphs), vertex attributes, and edge weights simultaneously. In a subsequent study [10], the authors proposed a mathematical framework to model dynamic systems in the state space of labeled and weighted simple graphs with attributes, based on hybrid systems theory [11]. In this framework, the state space $\mathbb{G}$ is endowed with algebraic operations that act simultaneously on topology, vertex attributes, and edge weights. Restricted to these algebraic operations, it is possible to construct a discrete dynamic model of the system state. However, these operations in [10] are limited to simple graphs.

1.2. Contributions and Organization

This article presents a numerical implementation of the operational capability model developed by Doliveira et al. [1] to address the problem illustrated in Figure 1. Given an offshore mission and a fleet of OSVs, two questions arise: (i) which OSV can carry out the mission at a given time t_0? and (ii) can the selected OSV complete the mission within the required time window? In [1], the model does not account for edge weights in multigraphs, which are necessary to describe a vessel (See Subsection 3.1 and Figure 4). We therefore propose to extend the model to the space of labeled and weighted multigraphs with attributes. This numerical implementation wants to leverage the algebraic operations developed in [10] to model discrete system dynamics. These operations were originally defined for labeled and weighted simple graphs with attributes; here, we extend them to multigraphs and provide a UML diagram to reproduce the numerical implementation. Finally, the distance ρ, used to evaluate operational capability in [1], is discrete and yields a binary response. In addition, we consider a more nuanced metric to characterize operational capability.

In the following sections, we first construct the effective discrete dynamic model for assessing operational capability (Section 2). Next, we illustrate the problem of selecting an OSV from a fleet for a given mission, as shown in Figure 1 (Section 3). The numerical implementation is described in Section 4, and finally, the conclusion is presented in Section 5.

2. Operational capability model

The practical implementation of the operational capability model by Doliveira et al. [1] within the framework of dynamical systems requires a clear definition of the main objects involved in the illustration of Figure 2, namely: the state space $\mathbb{G}$, the distance ρ, and the evolution law. This section aims to clarify these elements.

2.1. Graph space and algebraic operations

Let $\mathcal{I} \subseteq \mathbb{N}$ be a countable set of labels. We denote by $2^{\mathcal{I}}$ the power set of $\mathcal{I}$, defined such that for any $I \in 2^{\mathcal{I}}$, $I \subseteq \mathcal{I}$. Let $\mathbb{V} = \{(i, x_i) \mid i \in \mathcal{I}\}$ be a set of points indexed by $\mathcal{I}$, where $x_i \in \mathbb{R}^{n_i}$ is an attribute associated with the point labeled i, and $n_i \geq 1$ denotes the dimension of this attribute. We denote by $\mathcal{J}$ a set of labels distinct from $\mathcal{I}$.

Definition 1. *A labeled, weighted simple graph with attributes G, built on $\mathbb{V}$, consists of a pair (V_G, E_G), where $V_G \subseteq \mathbb{V}$ and E_G are the sets of vertices and edges of the graph G, respectively, such that*

$$V_G = \{(i, x_i) \mid i \in \mathcal{I}_G, x_i \in \mathbb{R}^{n_i}\} \quad \text{and} \quad E_G = \{(p, q, w_{p,q}) \mid (p, q) \in \mathcal{L}_G, w_{p,q} \in \mathbb{R}\}, \tag{1}$$

where $\mathcal{I}_G \subset \mathcal{I}$ and $\mathcal{L}_G \subseteq \mathcal{I}_G \times \mathcal{I}_G$ denote the sets of node and edge labels of G, respectively, and $w_{p,q}$ represents the weight associated with the edge (p, q). A label, weighted multi-graph with attributes (called in short attributed multi-graph) is a pair (V_G, E_G) defined as

$$V_G = \{(i, x_i) \mid i \in \mathcal{I}_G, x_i \in \mathbb{R}^{n_i}\} \quad \text{and} \quad E_G = \{(p, q, j, w_{p,q,j}) \mid (p, q) \in \mathcal{L}_G, j \in J(p, q), w_{p,q,j} \in \mathbb{R}\}. \tag{2}$$

where $J(p, q) \subset \mathcal{J}$ is the set of edge labels that exist between vertices p and q.

Figure 3 provides an illustration of a multigraph as defined in Definition 1. From this point forward, any reference to a (multi-)graph will imply a labeled, weighted graph with attributes. We denote by $\mathbb{G}$ the set of simple graphs and multigraphs defined over the vertex set $\mathbb{V}$. Both the target $[s^*]$ and the context C_{xt} are considered as subsets of $\mathbb{G}$. We now introduce the algebraic operations used to update the system state, which is represented as a (multi-)graph. We begin by recalling the algebraic operations defined for simple graphs in [10], and then extend these operations to the case of multigraphs.

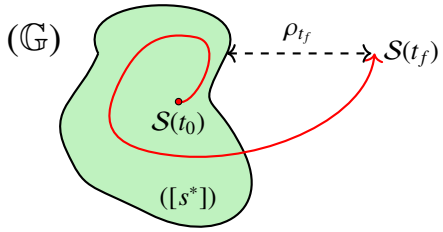

Figure 2: Operational capability as a dynamical system assessment, where $\mathbb{G}$ denotes the state space.

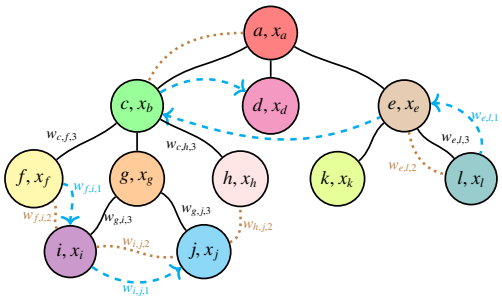

Figure 3: Example of a labeled, weighted multigraph with attributes[1].

Definition 2. *Considering two simple graphs $X = (V_X, E_X) = (\{(i, x_i)\}_{i \in \mathcal{I}_X}, \{(k, l, w_{k,l})\}_{(k,l) \in \mathcal{L}_X})$, and $Y = (V_Y, E_Y) = (\{(i, y_i)\}_{i \in \mathcal{I}_Y}, \{(k, l, v_{k,l})\}_{(k,l) \in \mathcal{L}_Y})$. The overlapping union of X and Y is an operation $X \cup_+ Y = (V_X \cup_+ V_Y, E_X \cup_+ E_Y)$ such that*

$$V_X \cup_+ V_Y = \{(i, x_i + y_i) \mid i \in \mathcal{I}_X \cap \mathcal{I}_Y\} \cup \{(i, x_i) \mid i \in \mathcal{I}_X \setminus (\mathcal{I}_X \cap \mathcal{I}_Y)\} \cup \{(i, y_i) \mid i \in \mathcal{I}_Y \setminus (\mathcal{I}_Y \cap \mathcal{I}_X)\}; \tag{3}$$

$$E_X \cup_+ E_Y = \{(k, l, w_{k,l} + v_{k,l}) \mid (k, l) \in \mathcal{L}_X \cap \mathcal{L}_Y\} \cup \{(k, l, w_{k,l}) \mid (k, l) \in \mathcal{L}_X \setminus (\mathcal{L}_X \cap \mathcal{L}_Y)\} \cup \{(k, l, v_{k,l}) \mid (k, l) \in \mathcal{L}_Y \setminus (\mathcal{L}_Y \cap \mathcal{L}_X)\}. \tag{4}$$

The overlapping intersection of X and Y is an operation $X \cap_+ Y = (V_X \cap_+ V_Y, E_X \cap_+ E_Y)$ such that

$$V_X \cap_+ V_Y = \{(i, x_i + y_i) \mid i \in \mathcal{I}_X \cap \mathcal{I}_Y\} \quad \text{and} \quad E_X \cap_+ E_Y = \{(k, l, w_{k,l} + v_{k,l}) \mid (k, l) \in \mathcal{L}_X \cap \mathcal{L}_Y\}. \tag{5}$$

These operations affect not only the structure but also the attributes and weights. The overlapping union $\cup_+$ modifies the structure by combining elements, whereas the overlapping intersection $\cap_+$ reduces the structure. Regarding attributes and weights, both operations act in an additive manner. The algebraic properties of these operations are studied in [10].When $\mathbb{G}$ is restricted to simple graphs, the triplet $(\mathbb{G}, \cup_+, \cap_+)$ forms an algebraic structure. In this structure, both $(\mathbb{G}, \cup_+, \cdot)$ and $(\mathbb{G}, \cap_+, \cdot)$ are complete and simple semi-vector spaces over the field of real numbers $\mathbb{R}$ (see Note **1** and theorems **1** and **2** in [10]). Here, $\cdot$ refers to the external multiplicative operation defined in [10]. In this work, we propose an extension of these operations to the case where $\mathbb{G}$ includes multigraphs; however, we do not investigate their algebraic properties here.

Definition 3. *Considering two multigraphs $X = (V_X, E_X) = (\{(i, x_i)\}_{i \in \mathcal{I}_X}, \{(k, l, j, w_{k,l,j})\}_{(k,l) \in \mathcal{L}_X, j \in J_X(k,l)})$ and $Y = (V_Y, E_Y) = (\{(i, y_i)\}_{i \in \mathcal{I}_Y}, \{(k, l, j, v_{k,l,j})\}_{(k,l) \in \mathcal{L}_Y, j \in J_Y(k,l)})$. The overlapping union of X and Y is an operation $X \cup_+ Y =$*

[1] Multigraph associated with the attribute G where $\mathcal{I}_G = \{a, b, c, \ldots, l\}$ is the set of vertex labels, x_α is the attribute of vertex $\alpha \in \mathcal{I}_G$, and $w_{\alpha,\beta,\gamma}$ is the weight of the edge connecting vertices $(\alpha, \beta) \in \mathcal{I}_G^2$, of type $\gamma \in J(\alpha, \beta) \subset \mathcal{J}$. The set $\mathcal{J}$ represents the possible types of links, depicted here using colors.

$(V_X \cup_+ V_Y, E_X \cup_+ E_Y)$ *such that*

$$V_X \cup_+ V_Y = \{(i, x_i + y_i) \mid i \in \mathcal{I}_X \cap \mathcal{I}_Y\} \cup \{(i, x_i) \mid i \in \mathcal{I}_X \setminus (\mathcal{I}_X \cap \mathcal{I}_Y)\} \cup \{(i, y_i) \mid i \in \mathcal{I}_Y \setminus (\mathcal{I}_Y \cap \mathcal{I}_X)\}, \tag{6}$$

$$\begin{aligned} E_X \cup_+ E_Y =& \{(k, l, j, w_{k,l,j} + v_{k,l,j}) \mid (k,l) \in \mathcal{L}_X \cap \mathcal{L}_Y,\ j \in J_X(k,l) \cap J_Y(k,l)\} \\ &\cup \{(k, l, j, w_{k,l,j}) \mid (k,l) \in \mathcal{L}_X \cap \mathcal{L}_Y,\ j \in J_X(k,l) \setminus (J_X(k,l) \cap J_Y(k,l))\} \\ &\cup \{(k, l, j, v_{k,l,j}) \mid (k,l) \in \mathcal{L}_X \cap \mathcal{L}_Y,\ j \in J_Y(k,l) \setminus (J_X(k,l) \cap J_Y(k,l))\} \\ &\cup \{(k, l, j, w_{k,l,j}) \mid (k,l) \in \mathcal{L}_X \setminus (\mathcal{L}_X \cap \mathcal{L}_Y),\ j \in J_X(k,l)\} \\ &\cup \{(k, l, j, v_{k,l,j}) \mid (k,l) \in \mathcal{L}_Y \setminus (\mathcal{L}_Y \cap \mathcal{L}_X),\ j \in J_Y(k,l)\}. \end{aligned} \tag{7}$$

The overlapping intersection of X and Y is an operation $X \cap_+ Y = (V_X \cap_+ V_Y, E_X \cap_+ E_Y)$ *such that*

$$V_X \cap_+ V_Y = \{(i, x_i + y_i) \mid i \in \mathcal{I}_X \cap \mathcal{I}_Y\}, \tag{8}$$

$$E_X \cap_+ E_Y = \{(k, l, j, w_{k,l,j} + v_{k,l,j}) \mid (k,l) \in \mathcal{L}_X \cap \mathcal{L}_Y,\ j \in J_X(k,l) \cap J_Y(k,l)\}. \tag{9}$$

2.2. Method for evaluating operational capability

The evaluation of operational capability consists in verifying whether the system state $\mathcal{S} \in \mathbb{G}$ belongs to the target set $[s^*] \subset \mathbb{G}$. This target is defined as an equivalence class expressed as:

$$[s^*] = \{g \in \mathbb{G} \mid g \cong s^*,\ x(g) \sim x(s^*),\ w(g) \sim w(s^*)\} \tag{10}$$

where the symbol $\cong$ denotes the existence of a graph isomorphism between g and s^* [13]. This implies that the multigraphs g and s^* share the same topological structure. For any $s \in \mathbb{G}$, the terms $x(s)$ and $w(s)$ represent the concatenation of the vertex attributes and edge weights of the multigraph s, respectively. The symbol $\sim$ denotes equivalence between the operands, meaning:

$$x(g) \sim x(s^*) \iff x(g) \in [x(s^*)] \subset \mathbb{R}^{n^*} \qquad \text{and} \qquad w(g) \sim w(s^*) \iff w(g) \in [w(s^*)] \subset \mathbb{R}^{m^*}, \tag{11}$$

where $[x(s^*)]$ and $[w(s^*)]$ are the equivalence classes of admissible attributes and weights defined by the structure of the multigraph s^*. These sets specify the valid values that the system, in state $\mathcal{S}$, must exhibit in order to fulfill the mission $\mathcal{M}$. The membership of $\mathcal{S}$ in $[s^*]$ is evaluated by the distance between the point $\mathcal{S}$ and the target set $[s^*]$:

$$\begin{aligned} \rho \colon \mathbb{G} \times 2^{\mathbb{G}} &\longrightarrow \mathbb{N} \\ (s_i, [s_j]) &\longmapsto \rho_s(s_i, s_j) + \rho_e(s_i, s_j) \end{aligned} \tag{12}$$

where $\rho_s(s_i, s_j)$ measures the topological similarity between s_i and s_j, and $\rho_e(s_i, s_j)$ assesses the equivalence of attributes and weights. It is important to note that the size of $\mathcal{S}$—that is, the number of its vertices and even its edges—is typically much larger, especially for complex systems, than the size of the target multigraph s^* [1]. Under these conditions, $\rho_s(\mathcal{S}, s^*)$ does not aim to find a full graph isomorphism between $\mathcal{S}$ and s^*, but rather a subgraph isomorphism [12]. This means that one seeks a subgraph s of $\mathcal{S}$ such that $s \cong s^*$.

Once such a subgraph $s \subset \mathcal{S}$ with $s \cong s^*$ is identified, the function $\rho_e(\mathcal{S}, s^*)$ evaluates whether the attributes and weights also match, *i.e.*, whether $x(s) \sim x(s^*)$ and $w(s) \sim w(s^*)$. The functions ρ_s and ρ_e are defined in [1], with ρ_e applying only to graph attributes. Its extension to weights is straightforward, since they are represented as vectors like attributes. In [1], ρ_s and ρ_e (restricted to attributes) correspond, respectively, to a weak discrete distance and a discrete pseudo-distance in $\mathbb{G}$. A graph $\mathcal{S}$ belongs to $[s^*]$ if $\rho(\mathcal{S}, [s^*]) = 0$, and does not otherwise. Membership is determined solely by the metric ρ.

2.3. Quasi-static model of operational capability

As illustrated in Figure 2, the system state $\mathcal{S}$ is assumed to evolve under the influence of elements from the mission context C_{xt}. We propose a simple discrete-time dynamic evaluation model of operational capability in $\mathbb{G}$. We define a time sampling set $\mathcal{T} = \mathcal{T}_i \cup \mathcal{T}_d \subset \mathbb{N}$, where $\mathcal{T}_i \cap \mathcal{T}_d = \emptyset$. The context C_{xt} is assumed to be a subset of $\mathbb{G}$, *i.e.*, $C_{\text{xt}} \subset \mathbb{G}$. At each time step $t \in \mathcal{T}$, we consider a pair of elements $(A_t, B_t) \in C_{\text{xt}}^2$, where A_t tends to increase or expand the topological structure of the system state $\mathcal{S}_t$, while B_t has the opposite effect. Both elements also act on the attributes and weights associated with the system state. We assume that for every $t \in \mathcal{T}$, the elements A_t and B_t are known and

that the target $[s^*]$ remains fixed over time. The evolution of the system is then described in a general form as follows: for all $t \in \mathcal{T}$, we have

$$\begin{cases} \rho_t = \rho(\mathcal{S}_t, [s^*]), \\ \mathcal{S}_{t+1} = f(\mathcal{S}_t, A_t, B_t), \end{cases} \tag{13}$$

with $\rho_t \in \mathbb{N}$. Using the overlapping union and intersection operations described in Definition 3 the function f is

$$f(\mathcal{S}_t, A_t, B_t) = \begin{cases} \mathcal{S}_t \cup_+ A_t, & \text{if} \quad t \in \mathcal{T}_i, \\ \mathcal{S}_t \cap_+ B_t, & \text{if} \quad t \in \mathcal{T}_d. \end{cases} \tag{14}$$

From a deterministic perspective, the dynamics of the system state $\mathcal{S}_t$ induced by the function f can be interpreted as an update of the state given the contextual elements A_t and B_t at each time step $t \in \mathcal{T}$.

3. Application to OSV fleet management

To instantiate the operational capability model described in this paper, we consider two relevant implementations for managing an OSV fleet. These two implementations correspond to different phases derived from the scenario illustrated in Figure 1. This figure presents the case of a client seeking to charter an OSV from a shipowner to carry out a specific mission. To this end, the client provides the shipowner with details such as the mission type, the geographical region, the time period, and the technical specifications of the vessel they require. Based on this information, the shipowner is expected to select the most suitable OSV from their fleet to fulfill the mission, relying on the concept of operational capability. As an example, let us consider a mission involving towing and the installation of an offshore oil platform. The region corresponds to a route extending from point P_1 to point P_2 in the Atlantic Ocean. The time period covers a two-week span at a specific time of the year. The technical specifications, in turn, are defined as follows.

(TS): "The vessel must have both a captain and a deck officer in good health. They need to communicate effectively to coordinate their actions. The captain must have access to the control and dynamic positioning system. The thruster must have a power output greater than 2450 *kW."*

To determine the most suitable OSV for accomplishing a mission, we consider two evaluation phases, each corresponding to a specific questions: *i) At a given initial time, which OSV in the fleet is best suited for the mission? ii) Once this OSV has been selected, and given the mission context, is the OSV capable of successfully completing the mission over the specified time period?* The first question focuses on the initial selection process, whereas the second pertains to the monitoring of the selected vessel.

3.1. Initial selection of an optimal OSV for a mission

Selecting a vessel from a fleet for a given mission is not necessarily a trivial task, primarily due to the inherent complexity of a vessel. A vessel is a complex system composed of both material resources (such as thrusters, generators, etc.) and human resources (such as the captain and the deck officer), all interconnected through multiple relationships (see Figure 4). Each of these components is itself a complex subsystem, associated with large volumes of highly heterogeneous data. Moreover, some operators manage fleets with dozens or even hundreds of vessels.

Given this level of complexity, it becomes essential to organize vessel information in a structured and efficient manner. In this context, the multigraph proves to be a suitable modeling tool. To identify the most suitable vessel for a mission, the operational capability model $\langle \mathcal{V}_i, \mathcal{M}, \rho \rangle$, for $i = 1, \ldots, N_v$, can be applied to each of the N_v vessels in the fleet, where $\mathcal{V}_i \in \mathbb{G}$ represents the state of the i-th vessel. For illustration purposes, we limit our scope to a fleet of two vessels, and instead of multigraphs, vessel states will be represented using simple graphs.

Figure 4 illustrates the evaluation process for selecting the appropriate vessel. The mission target is denoted by $[\mathcal{V}^*]$, which represents, in graph form, the set of technical specifications *(TS)* previously described by the client. This graph encodes constraints on various vertex and edge attributes. The objective is to determine whether the two vessels, whose states are denoted $\mathcal{V}_1$ and $\mathcal{V}_2$, satisfy the following: First, the minimal topological structure required by $\mathcal{V}^*$, namely, the deck officer must be connected to the captain via a communication link; the captain must have access to the control system, which in turn must be connected to the thruster. Second the target attribute constraints: deck officer

health score $\delta > 0.6$, captain health score $\delta > 0.65$, communication signal strength between captain and deck officer $S_s > 50\%$, thruster power $P > 2450$ kW.

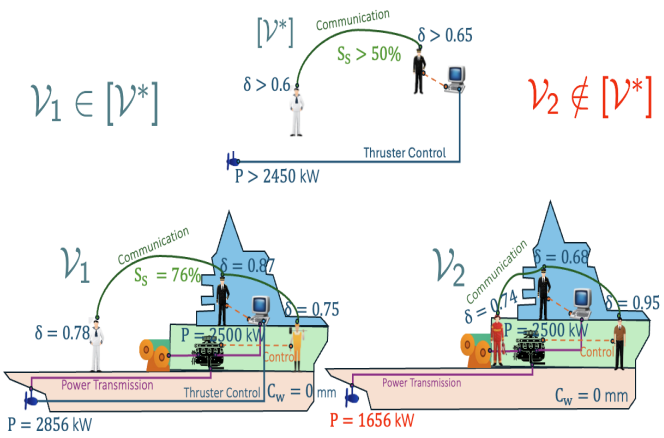

Figure 4: Operational capability assessment of two vessels in a fleet, with states $\mathcal{V}_1$ and $\mathcal{V}_2$, for the towing and installation of an oil platform. The mission target is denoted by $[\mathcal{V}^*]$.

Target Components	**V1**	**V2**
Deck officer present	✓	✗
Captain present	✓	✓
Control System present	✓	✓
Thruster present	✓	✓
Deck officer–Captain link present	✓	✗
Captain–Control System link present	✓	✓
Control System–Thruster link present	✓	✗
Structural distance (ρ_s)	0	1
Deck officer ($\delta > 0.6$)	✓	✗
Captain ($\delta > 0.65$)	✓	✓
Thruster ($P \geq 2450$ kW)	✓	✗
Deck officer–Captain ($S_s > 50\,\%$)	✓	✗
Attribute distance (ρ_e)	0	1
Total distance ($\rho = \rho_s + \rho_e$)	**0**	**2**
Mission Fit	**Yes**	**No**

Table 1: Evaluation of mission fit based on structural and attribute constraints.

Vessel $\mathcal{V}_1$ satisfies all the above constraints and is therefore suitable for the mission. On the other hand, vessel $\mathcal{V}_2$ does not meet the minimum structural and attribute requirements. Structurally, it lacks a deck officer to coordinate operations with the captain, and the control system is not linked to the thruster. Additionally, the thruster does not meet the required power threshold. Consequently, only vessel $\mathcal{V}_1$ is eligible to undertake the mission. In Table 1, we show the result of the result of the comparison between $\mathcal{V}_1$ and $\mathcal{V}_2$ with the target $[\mathcal{V}^*]$ as represented in Figure 4.

3.2. Monitoring OSV during mission

According to Figure 4, the vessel best suited to carry out the mission at the initial time is the one whose state is represented by $\mathcal{V}_1$. Let us now assume that this vessel is deployed for the mission and that its state is updated at specific time steps based on the evolving context. In this case, it becomes necessary to monitor the vessel's state regularly, at a defined frequency, to ensure that it continues to meet the minimum requirements defined by $[\mathcal{V}^*]$, as illustrated in Figure 4.

Figure 5 illustrates a scenario in which the vessel state evolves under the influence of the mission context until it becomes non-operational. The vessel state is assumed to be updated regularly. This update can be carried out according to the model (13), where A_t and B_t are the contextual factors affecting the state of the system $\mathcal{S}_t$. An implementation of this model is illustrated in Subsection 4.2. On day one, the vessel is in an initial state suitable for the mission. By day four, adverse weather conditions—rain and strong winds—temporarily make continuing the mission dangerous. These conditions jeopardize the health of the deck officer and complicate communication between the officer and the captain, which is essential for coordinating operations. The mission is therefore suspended pending a favorable weather window [14], which, according to this scenario, occurs on day seven. On day twelve, a fire breaks out in the engine room near the generator. This incident destroys the connection between the control system and the thruster and requires a firefighter's intervention to extinguish. Both consequences alter the topological structure of the graph representing the vessel. The loss of connection between the control system and the thruster causes the vessel $\mathcal{V}_2$ to fall outside its target configuration $[\mathcal{V}^*]$ (See Figure 4). Furthermore, the fire damages the generator to the extent that it delivers insufficient power to the thruster. This failure renders the vessel inoperative for the remainder of the mission.

4. Numerical implementation of a mathematical framework, a UML

It is important to note that the application described in Section 3 represents a simplified version of the real-world scenario. In practice, vessel data are extensive, covering technical specifications of fixed and mobile equipment, health indicators of components and subcomponents (when available), certifications, and detailed crew information (e.g.,

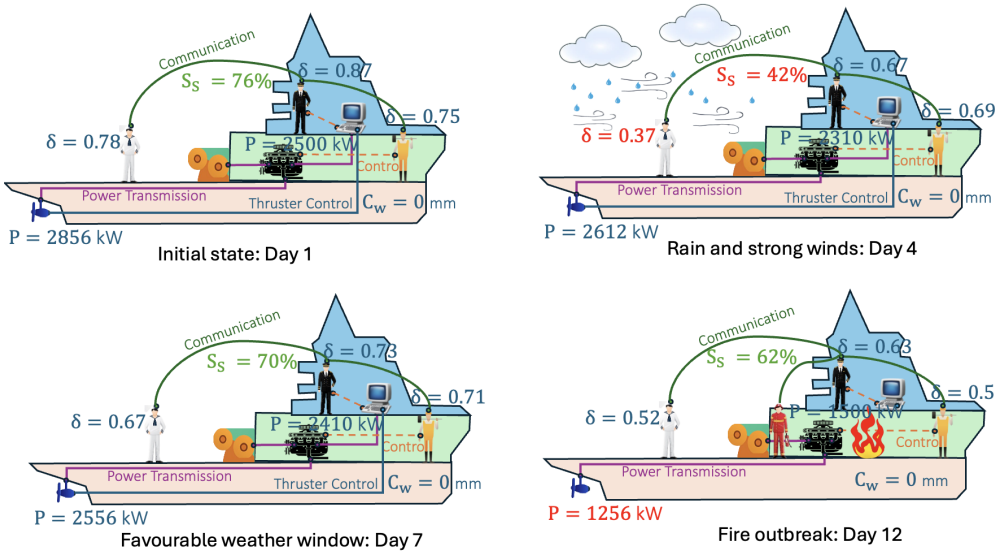

Figure 5: Example scenario illustrating the dynamics of an OSV influenced by its operational context during mission execution.

vaccination status, experience, and rank). The vessel's structure, describing the interconnections between components within the scope of available knowledge, can also be considered. Altogether, this can amount to more than 10,000 data points per vessel, while shipowners typically manage fleets exceeding 100 vessels.

The objective is to recast the model $\langle S, M, \rho \rangle$ into flexible, readable, and extensible software components, in order to build a decision-support tool, as illustrated in Figure 1. To this end, we designed a high-level architecture, whose UML diagram is shown in Figure 6, comprising four main classes: **Graph**, **State**, **Target**, and **Distance**. The **Graph** class encodes the system structure, where vertices represent components and edges represent interactions between them. Both vertices and edges are labeled. The **State** class, inheriting from **Graph**, contains data members and methods. Among the data members are the graph representing the system's topological structure (`topology`), the attributes associated with vertices (`attributes`), and the weights of edges (`weights`). Its methods include `OverlappingUnion()` and `OverlappingIntersection()`, as defined in Definition 3.

The **Target** class includes three data members: `targetGraph`, which defines the required topological structure (e.g., Figure 4); `vertexConstraints`, specifying admissible attribute values for selected vertices, possibly of dimension greater than one; and `edgeConstraints`, specifying admissible weights for selected edges. The **Distance** class contains two data members—the system state and the target—and a method ρ that evaluates the membership of a state with respect to a target. This distance is defined using the metrics ρ_s and ρ_e from [1]. The metric ρ_s relies on subgraph isomorphism to verify whether the system state contains the minimal required topological structure (e.g., Figure 4), while ρ_e evaluates whether the attributes and edge weights belong to their respective sets defined by the target. In [1], ρ_e is defined for attributes; its extension to edge weights is straightforward.

Additionally, the **ContextEvent** class acts on the system state. It inherits from **State** and adds two data members: `time`, which defines a discrete set of instants, and `mode`, specifying the type of action corresponding to the function f defined in (14). This class induces a discrete dynamic, as discussed in Subsection 2.3. Extending it to a continuous dynamic would require reconsidering the implementation within the mathematical framework proposed in [10].

4.1. *Fit matrix for multiple missions*

In fleet-screening practice, however, engineers often need a quick *ranking* of several candidate vessels against several mission targets. We therefore compute, for each pair (Target, Vessel), the *fit fraction*

$$\phi = \frac{\#\text{ matching nodes} + \#\text{ matching edges}}{\#\text{ required nodes} + \#\text{ required edges}} \in [0,\ 1], \tag{15}$$

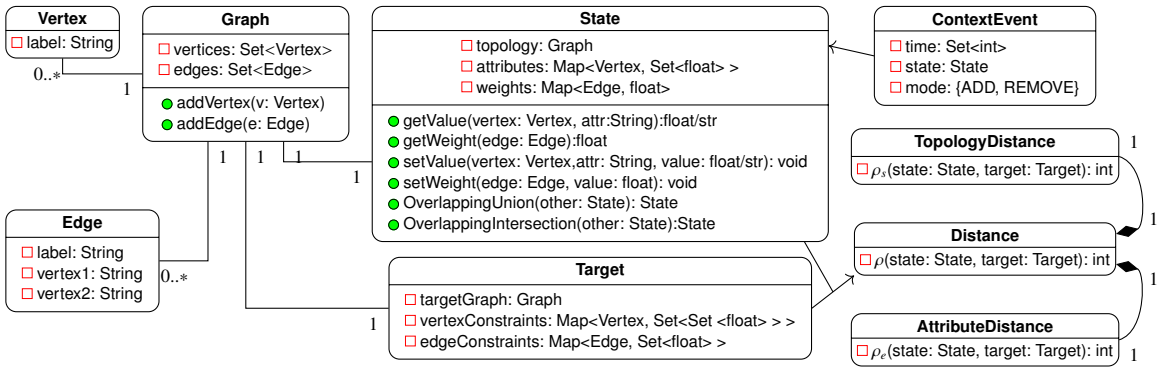

Figure 6: UML class diagram for the operational capability implementation.

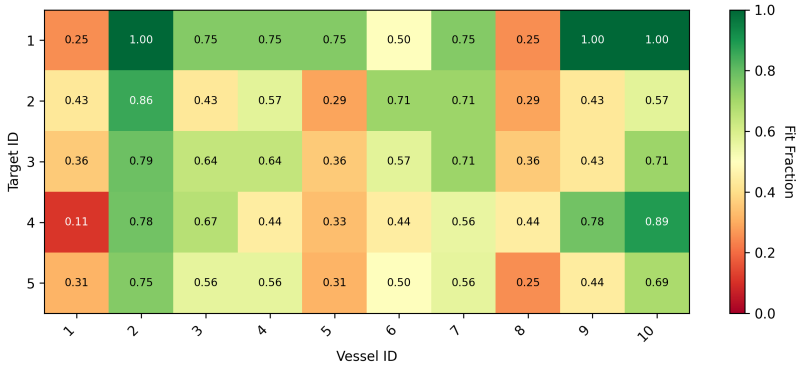

Figure 7: Fraction of mission–component matches for every *Target ID* (rows) against every *Vessel ID* (columns). A value of 1 means the vessel satisfies *all* structural and attribute constraints of the target; intermediate values show partial compliance (*e.g.* 0.57 means 4 out of 7 required elements are present with correct attributes).

i.e. the percentage of target elements whose structure and attributes are found in the current vessel graph. Figure 7 shows an example with five synthetic mission targets and ten vessel variants. Dark-green cells ($\phi = 1$) correspond to $\rho = 0$; yellow to red cells highlight partial compliance and allow planners to identify the *closest* retrofit option at a glance. The output illustrated in Figure 7 can assist ship managers in making quick decisions by providing a concise and visual summary of each vessel ability to perform a given mission. For example, Vessel 2 could be assigned to Target 5, Vessel 10 to Target 4, Vessel 7 to Target 3, Vessel 6 to Target 2, and Vessel 5 to Target 1.

Real-world decision-making involves additional data and expert judgment, such as from the shipowner. Our approach provides a visual representation of the alignment between vessels and missions using structured data. It is not meant to give final decisions but to clarify the operational constraints that must be considered.

4.2. *Step-wise demonstration of graph update operators in a mini-vessel*

To make the update law $S_{t+1} = f(S_t, A_t, B_t)$ concrete, we build a *mini-vessel* graph with three functional nodes—*Engine*, *Propeller*, and *Control*—linked by "Power" and "Ctrl" edges. **Step** t_0 Baseline state S_{t_0}. **Step** t_1 $(\cup_+ A_1)$**.** A skid-mounted *Auxiliary Generator* (**AuxGen**) is craned aboard. AuxGen can deliver roughly 20 % of the main-engine power, so we add the node *AuxGen* and an auxiliary "Power" edge to the propeller. Both items appear green in the Figure 8. **Step** t_2 $(\cap_+ B_2)$**.** A storm causes damage: the main shaft is derated by 30 % (red "Power" edge), Propeller health drops from 100 to 40, and AuxGen health drops from 100 to 70 (also red). The "Ctrl" link from *Control* to *Engine* is lost and therefore is shown as a black dotted edge.

Figure 8 visualises the three successive graphs with a fixed layout; the colour/line convention is: **Green** — structure *added* by $\cup_+$, **Blue** — attribute *increase*, **Red** — attribute *decrease* or derating, **Black dotted** — elements *removed* by

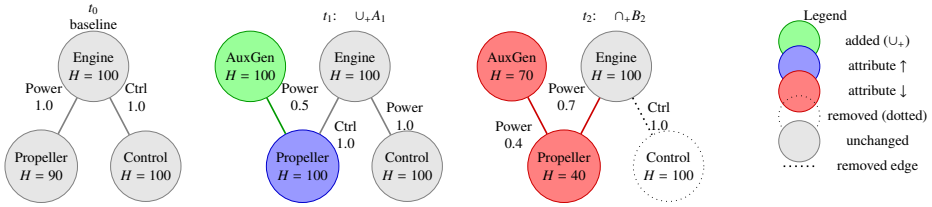

Figure 8: Illustrative three-step evolution of the toy vessel graph under one $\cup_+$ (maintenance) and one $\cap_+$ (damage) update.

$\cap_+$. This didactic run links the formal operators directly to tangible changes on board and will be reused as a legend for larger scenarios.

5. Conclusions

In this paper, we implemented a generic operational capability model for a specific problem in fleet management of OSVs by an operator: the selection and monitoring of an OSV for a given mission. Within the framework of dynamical systems theory, implementing this model required clearly specifying the nature of the state space (labeled and weighted multigraphs with attributes), the evolution law (a discrete law based on algebraic operations on multigraphs), and the evaluation method (a discrete distance on the state space of multigraphs). This study demonstrates that assessing operational capability involves not only a list of system parameters but also its topological structure. A possible extension of this work could consist of validating the proposed model.

6. Funding and acknowledgements

This work was carried out within the LIS Lab at Aix Marseille Université. We thank Bourbon Marine & Logistics for their industrial insight. This project is part of the larger CASSIOPPÉE project, supported by the French State as part of Investments for the Future Programme, now integrated into France 2030, and operated by ADEME.

References

[1] Doliveira, A., Roman, C., Graton, G., and Ouladsine, M. (2024) “Modeling the Operational Capability of a System in Relation to a Defined Mission.” *IFAC PapersOnLine* **58** (21): 156–161. doi: 10.1016/j.ifacol.2024.10.165.

[2] R. Guo. "Weapon system operational capability". *In Proceedings of the 2015 International Conference on Electrical, Automation and Mechanical Engineering*, pages 130–132. Atlantis Press, 2015/07.

[3] Ke Deng, Fei Long, Xuwen Sun, and Jia Xu. "Analysis on operational capability evaluation of ship formations : Anti-submaring firing system". *In 2014 Seventh International Symposium on Computational Intelligence and Design*, volume 2, pages 61–64. IEEE, 2014.

[4] Davis, P. K., et al. (2002) *Analytic architecture for capabilities-based planning, mission-system analysis, and transformation*. RAND, Santa Monica, CA.

[5] Cho, N., Moon, H., Cho, J., Han, S., and Pyun, J. (2022) “A framework for determining required operational capabilities: A combined optimization and simulation approach”. *Journal of Defense Management* **12**: 234.

[6] Hristov, N., Radulov, I., Iliev, P., and Andreeva, P. (2010) “Prioritization methodology for development of required operational capabilities”. *SSRN*. Available at SSRN 3135696.

[7] Jichao Li, Bingfeng Ge, Jiang Jiang, Kewei Yang, and Yingwu Chen. “High-end weapon equipment portfolio selection based on a heterogeneous network model”. *Journal of Global Optimization,* 78 :743–761, 2020.

[8] Jichao Li, Danling Zhao, Bingfeng Ge, Jiang Jiang, and Kewei Yang. “Disintegration of operational capability of heterogeneous combat networks under incomplete information”. *IEEE Transactions on Systems, Man, and Cybernetics : Systems,* 50(12) :5172–5179, 2018.

[9] Papadopoulos, A. and Troyanov, M. (2006) “Weak metrics on euclidean domains”. *arXiv* (preprint math/0609236.).

[10] Doliveira, A., Roman, C., Graton, G., and Ouladsine, M. (2025) “Dynamical System on Graph State-Space”. *Automatica* (preprint submitted).

[11] Goebel, R., Sanfelice, R. G., and Teel, A. R. (2012) *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

[12] Moorman, J. D., Tu, T. K., Chen, Q., He, X., and Bertozzi, A. L. (2021) “Subgraph matching on multiplex networks”. *IEEE Transactions on Network Science and Engineering* **8** (2): 1367–1384.

[13] McKay, B. D., and Piperno, A. (2014) “Practical graph isomorphism, II”. *Journal of Symbolic Computation* **60**: 94–112.

[14] Foo, Y. P., Gan, K., Giudice, D., and De Masi, G. (2014) “Analysis of windows of opportunity for weather-sensitive operations.” *Oil and Gas Facilities* **3** (4): 63–71.