Ontology-Based Information Infrastructure for Autonomous Ships

Dean Sumić[1], Lada Maleš[2], Tonćo Marušić[3], Marko Rosić[4]

The maritime industry is witnessing a revolutionary transformation with the development of autonomous ships. Unmanned vessels employ advanced technologies such as artificial intelligence, especially machine learning, and intelligent agents for autonomous navigation. The successful integration of autonomous ships into existing maritime ecosystems requires a robust infrastructure capable of addressing various challenges, including interoperability of different ship systems. To ensure interoperability, it is crucial to establish a suitable information infrastructure. The implementation of ontologies within the information infrastructure of autonomous ships facilitates the use of intelligent agents. This paper discusses the concept of ontological information infrastructure tailored for autonomous ships and gives an example of an ontology designed specifically for autonomous vessel firefighting systems.

KEY WORDS

~ Autonomous ships,
~ Interoperability,
~ Ontologies.

[1] University of Split, Faculty of Maritime Studies, Split, Croatia
[2] University of Split, Faculty of Humanities and Social Sciences, Split, Croatia
[3] University of Mostar, Faculty of Science and Education, Mostar, Bosnia and Herzegovina
[4] University of Split, Faculty of Science, Split, Croatia
e-mail: dean.sumic@pfst.hr

doi: 10.7225/toms.v13.n01.008

Received: 16 Oct 2023 / Revised: 15 Feb 2024 / Accepted: 1 Apr 2024 / Published: 20 Apr 2024

1. INTRODUCTION

As the volume of global maritime traffic continues to grow, the maritime industry is undergoing a revolutionary change with the advent of autonomous ships. The integration of autonomous ships has social, economic, legal, and technical implications beyond the maritime industry (Kim et al., 2020). In the maritime context, the term autonomous shipping refers to the ability of a ship to autonomously monitor and execute its actions during transportation from one port to another (Rødseth, 2017). Autonomous ships can lead to long-term cost savings for shipping companies by eliminating crew-related costs and optimising operational efficiency. These cost savings can potentially trickle down to consumers in the form of lower transportation prices. Numerous projects that aim to increase the efficiency, safety, and cost effectiveness of such vessels are underway. These research efforts result in advances in maritime transportation that primarily bring the following benefits:

- Cost savings on crew salaries – this aspect is crucial as crew salaries represent a significant portion of overall expenses (Ziajka-Poznańska and Montewka, 2021). These savings also include savings on crew accommodation and food expenses (Hogg and Ghosh, 2016).
- Enhanced operability - autonomous vessels can work continuously without the need for rest and shifts, enabling uninterrupted and faster transportation of goods.
- Energy consumption savings - the absence of a crew, coupled with the implementation of new technologies on board, contributes to energy efficiency (Allal et al., 2018). The crew-free aspect is also considered in the autonomous vessel design stage, resulting in enhanced aerodynamics that further contributes to energy savings.
- Facilitated use of new shipping routes, such as the Arctic route (Munim et al., 2022).
- Increased cargo capacity of ships due to a new design paradigm that does not require space for the crew (Hogg and Ghosh, 2016).
- Improved safety. The human factor is the most common cause of safety issues in maritime traffic. This factor is eliminated by introducing autonomous vessels equipped with appropriate safety mechanisms (Galieriková, 2019).
- Improved logistics and supply chains - autonomous vessels ensure better integration with other parts of the logistics chain, including storage, cargo handling, and port services, leading to reduced congestion in ports, as the introduction of autonomous ships can optimize traffic management, minimize congestion and waiting times for berthing and cargo unloading (Kurt and Aymelek, 2022).

The successful integration of autonomous ships into commercial maritime processes is accompanied by various challenges that need to be addressed. This paper focuses on the data-driven segments of autonomous ships. Data-related challenges are as follows:

- Data integration - Autonomous ships generate large amounts of data from various sources. Incorporating and interpreting this data in real time is crucial for making relevant decisions.
- Communication with external stakeholders - The maritime industry has a variety of stakeholders, including shipping companies, port authorities, regulatory bodies, and providers of various other services. Ensuring seamless communication and data exchange of an autonomous ship with these stakeholders is essential.
- Data safety and security - Safety is of utmost importance in maritime operations. Ensuring the safety of data and systems from cyber threats is crucial to prevent potential accidents or unauthorized access.

A model of an information infrastructure for autonomous ships based on ontologies that establish a shared understanding of data and their interrelationships is presented in this paper. In computer science, an ontology is a hierarchically organized structure containing a set of concepts that can be used as a foundational framework for building a knowledge base (Swartout et al., 1996). This paper shows how ontologies provide a formal representation of knowledge that describes entities related to ships and the maritime environment (objects, concepts, events), their properties, and interrelationships, thus enabling the organization and categorization of information in a way that facilitates analysis and decision making.

The paper is organized as follows. Section 2 explains the need for internal and external interoperability of autonomous ships. Section 3 describes ontologies and methods for achieving interoperability using ontologies. Section 4 discusses the impact of using ontologies in autonomous ships. Section 5 gives an example of an ontology designed to support the operation of an autonomous ship. This section gives an overview of the ontology-based interoperability model for autonomous ship fire protection systems. The conclusion and recommendation for establishing standards for ontological information infrastructure are given in section 6.

2. INTERNAL AND EXTERNAL INTEROPERABILITY OF AUTONOMOUS SHIPS

The International Maritime Organization (IMO) developed a preliminary four-category classification of ship autonomy levels (IMO, 2021). At the lowest level, there are ships that require a human crew, with occasional assistance from automated systems. At a higher level, there are ships that still require a human crew, but certain functions of these ships can be remotely controlled. The next level are unmanned remotely operated ships, while fully autonomous ships are the highest level. As shown in Figure 1, as the level of vessel autonomy increases, so does the importance of its information infrastructure. Since fully autonomous ships operate without direct human intervention, a reliable and secure data flow becomes of utmost importance.

Autonomous ships are complex systems composed of various subsystems. Figure 2 shows a potential breakdown of ship subsystems. Here, ship subsystems at the highest level are divided within three groups: (i) navigational subsystems; (ii) safety subsystems; and (iii) monitoring subsystems. Figure 2 shows that some subsystems, such as the collision avoidance subsystem, are divided into two functional affiliation categories. Due to its functionality, this system falls both in the category of navigation subsystems and the category of safety-related subsystems.

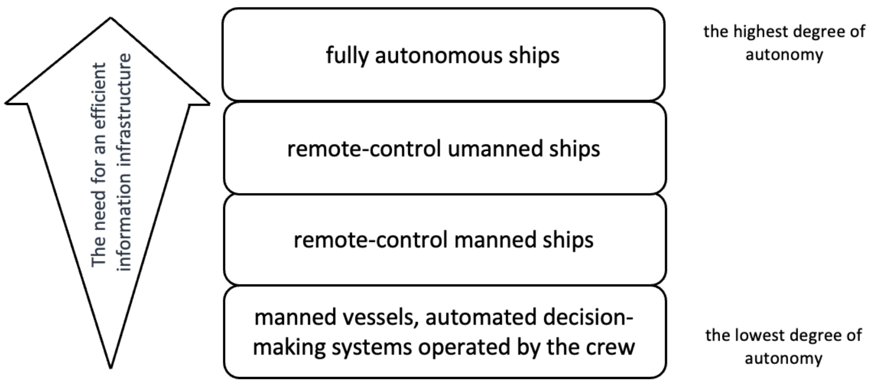

Figure 1. Vessel autonomy levels and the importance of information infrastructure efficiency

The efficient operation of an autonomous ship is inconceivable without the interoperability of the subsystems described here. The IEEE Standard Computer Dictionary defines interoperability as the ability of two or more systems to exchange information and use such exchanged information (1991). The interoperability of various subsystems is ensured through the establishment of an appropriate information infrastructure (Im et al., 2018). One of the main challenges when planning the implementation of interoperability is the fact that different ship subsystems are manufactured by different producers. A reliable data exchange and comprehension method needs to be devised to ensure the harmonious functioning of the whole, while avoiding any negative impact on the ability to work on improving each system component.

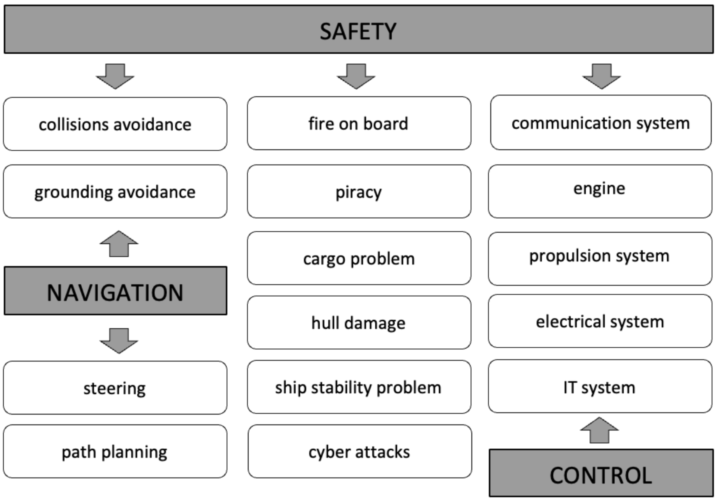

Figure 2. Breakdown of ship subsystems

The need for interoperability is not limited to achieving interoperability of autonomous ship subsystems, i.e. internal interoperability. External interoperability is required as well. External interoperability refers to the interoperability of relevant ship systems with systems located outside the ship. This set of systems includes authorized stakeholders in maritime processes, such as port authorities, regulatory bodies, ship owners, and shipping companies. Here, it is particularly important to emphasize safety challenges, as the communication between an autonomous ship and its environment can be a potential weak link in the safety chain and an entry point for cyber-attacks.

The development of agent-based autonomous ship subsystems is a current trend in the further development of autonomous vessels (Males et al., 2022). An agent is an entity that is autonomous and possesses the properties of reactivity, proactivity, and sociability. Each agent can be represented at the highest level with the structure shown in Figure 3. An agent obtains information on environmental conditions through its sensors. The agent's knowledge includes the knowledge of the methods for analysing the information received, as well as methods for deciding on the actions that agent will perform. In an environment, an agent takes actions through its effectors. Agents are often structured as multi-agent systems, with agents often collaborating in their actions. The functions of autonomous ships can be realized through the implementation of multi-agent systems, where agents take over tasks traditionally performed by the ship's crew. This methodology makes it possible to conceptualize a complete ship as an integration of different cooperative multi-agent systems.

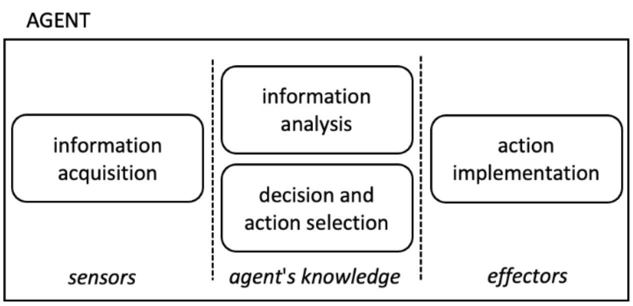

Figure 3. Agent structure (Males et al., 2022)

Suitable interoperability infrastructure needs to be developed to ensure effective collaboration. The implementation of ontologies into the information infrastructure of autonomous ships facilitates the use of intelligent agents, which is yet another motivation for further research in this field.

3. ONTOLOGIES

The quantity of data generated and stored in today's digital age requires efficient data organization and interpretation methods. Ontologies are a key solution that enables a more in-depth understanding of data. The most frequently used definition of ontology in computer science is the one devised by Gruber (Gruber, 1993), who defines ontology as an explicit specification of the concepts in a domain, i.e. ontology can be described as a conceptual framework that assists in modeling and representing knowledge about a specific domain. The purpose of an ontology is to enable humans and machines to better understand and utilize exchanged data, which ensures interoperability. The most significant aspects of ontology use are:

- elimination of ambiguity and confusion in communication due to the precise definition of concepts,
- facilitated data integration,
- better understanding of the context in which data are used,
- improved data search due to precise query definitions.

The use of ontologies in knowledge-based systems can be illustrated by a three-layer model shown in Figure 4. The lowest layer of this model consists of languages used to define ontology. The middle layer represents ontologies from various domains, defined by languages from the lowest layer. The top layer comprises data described by ontologies from the middle layer.

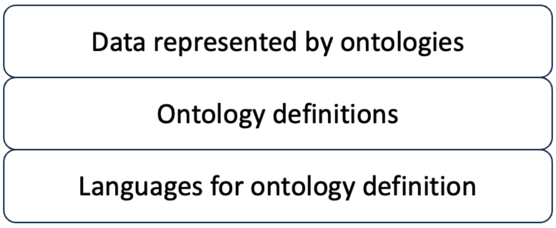

Figure 4. Three-layer ontology model

In order to access data represented by a certain ontology, an entity must possess the capability to interpret the given ontology. Moreover, the entity must possess knowledge of the language in which the ontology is defined. In more complex systems, data can be represented using more than one ontology. These data can be fully accessed only by entities that have the ability to integrate ontology definitions.

Twenty years ago, languages for defining ontologies were particularly intensively researched in Semantic Web studies (Mishra and Kumar, 2011). At the time, most of the initial versions of these languages, which are still in use today, were defined. The choice of language for defining ontologies depends on the specific needs of the project and the complexity of modeled ontologies. The most frequently used languages for defining ontologies are the Resource Description Framework (RDF) and the Web Ontology Language (OWL) (Gomez-Perez and Corcho, 2002, Udrea et al., 2010). Although primarily developed for the Semantic Web, these two languages have surpassed the Semantic Web framework and are used in a wide range of applications where a precise and unambiguous representation of a dataset is required. RDF is a language that describes resources and their interrelationships. It uses triples consisting of subjects, predicates, and objects to describe statements about resources. RDF is often used in data structure modeling and to define basic relationships between entities.

OWL is a language for modeling ontologies and formally expressing knowledge. OWL defines complex logical relationships, restrictions, and various levels of abstraction. In addition to RDF and OWL, additional formats and sub-dialects have been developed to facilitate working with these languages. For OWL in general, there are three main sub-dialects: OWL Lite, OWL DL (Description Logic), and OWL Full (Gu and Zhang, 2021). These three sub-dialects differ in terms of complexity and the ability to represent knowledge. A number of associated languages have been developed for RDF, such as:

- RDFS (RDF Schema) - language for modeling basic ontologies, class hierarchies, and properties (ter Horst, 2005).
- SPARQL - query language for expressing RDF data queries (Pérez et al., 2009).
- Terse RDF Triple Language (Turtle) - format for writing RDF triples in a human-readable textual form (Beckett and Berners-Lee, 2008).

Ontology-based information infrastructure is an information infrastructure in which data $A = \pi r^2$ are exchanged between networked components via corresponding ontologies. In the case of autonomous ships, such an information infrastructure implies that the subsystems presented in the previous section exchange data with each other using agreed ontologies that cover their respective operational domains. In such an environment, each ship subsystem can be considered an agent with access to a shared information infrastructure that facilitates a data exchange space. This space contains ontology definitions and the data represented by these ontologies. The ship subsystems access this area to exchange data necessary for the efficient functioning of the autonomous ship (Figure 5). This approach enables interoperability that goes beyond data exchange and encompasses the exchange of entire contexts related to the information presented. Thus, this type of interoperability can be defined as semantic interoperability. Ontology integration methods are embedded in the lower layer languages of the three-layer ontology model. This is particularly important for complex systems such as a ship, as a single ontology cannot be expected to cover all the necessary areas. Ontologies enabling autonomous ship functions are built based on the bottom-up principle. By using integration methods, agents combine data from different processes on the ship. For example, the agent responsible for path planning, uses ontology integration methods to access fire status data. By integrating the data, this agent can alter the course or speed if necessary to assist fire fighting.

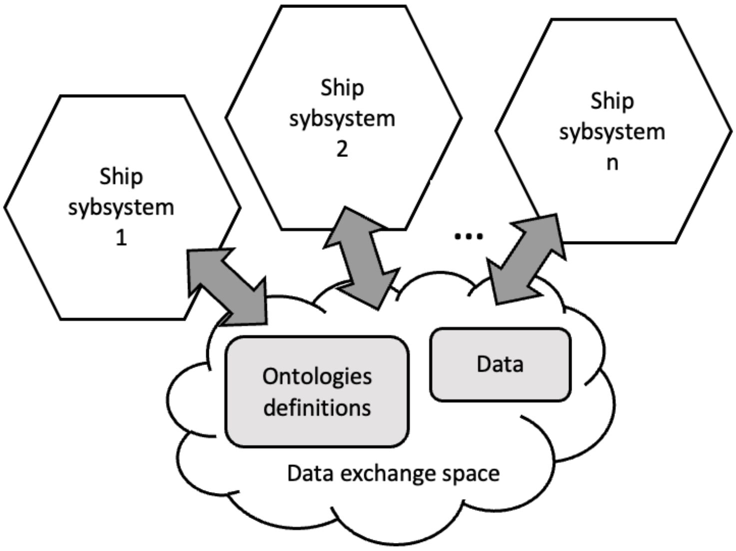

Figure 5. Semantic interoperability framework (Males, 2023)

In the following sections the positive impacts of using ontologies in autonomous ships is considered and an example of an ontology designed specifically for use within the semantic interoperability framework of an autonomous vessel presented.

4. THE IMPACT OF THE APPLICATION OF ONTOLOGY IN AUTONOMOUS SHIPS

The application of ontologies in autonomous ships mostly includes all the positive attributes listed in Section 3 hereof. The benefits of using ontologies in the information infrastructure of autonomous ships are:

- Semantic data integration - ontologies provide a shared data representation vocabulary and logical structure which facilitates the integration and exchange of information from different subsystems of an autonomous ship.
- Interoperability and communication – ontologies establish a shared understanding of data which enables different systems and stakeholders to achieve effective communication, promoting collaboration and data sharing throughout the maritime ecosystem.
- Improved situational awareness - due to the improved interoperability with the external stakeholders of maritime processes, autonomous vessels access relevant and updated data about weather conditions, navigational hazards, and vessel traffic. This improves situational awareness and contributes to safer navigation.
- Effective decision-making - ontologies enable autonomous vessels to integrate data, which can have a positive impact on processes such as route planning optimization, fuel consumption, and overall operational efficiency.
- Regulatory compliance - ontology can be designed to incorporate international maritime regulations, ensuring that autonomous vessels adhere to the necessary standards.

Security and protection are key concerns in the maritime industry, and the implementation of autonomous ships requires a robust approach to mitigate potential risks. Ontological information infrastructure can improve security and protection in several ways:

- Risk assessment and avoidance - ontologies integrate data from various sensors, historical records of incidents, and navigational charts to provide information relevant for navigation and avoidance subsystems, assisting decision making with respect to avoiding potential hazards, such as collisions with other vessels, grounding, or adverse weather conditions, by adjusting routes and speeds as necessary.
- Cybersecurity - autonomous ships are susceptible to cyber threats due to their heavy reliance on interconnected subsystems and data exchange, as well as data interchange with external elements. Ontologies can incorporate safety measures and access controls, ensuring that critical information are protected against unauthorized access, manipulation, or cyberattacks.
- Redundancy and fault protection mechanisms - safety subsystem redundancy is a crucial safety element in autonomous navigation. Ontologies can incorporate defined action procedures that enable autonomous ships to switch to alternative subsystems or actions in case of primary subsystem failures, thus ensuring continuous and secure operations.
- Remote monitoring and intervention - ontological information infrastructure can facilitate remote monitoring and control of autonomous ships from onshore control centers. In emergency situations or in case of unforeseen events, human operators can intervene and take manual control of the vessel, ensuring human access to the safety process.

The next aspect where ontologies can contribute to the development of autonomous ships is the impact on the environment and sustainability. Autonomous ships have the potential to bring environmental benefits to the maritime industry. By optimizing route planning and fuel consumption, these ships can reduce greenhouse gas emissions and minimize their environmental footprint. Ontological integration within the information infrastructure of an autonomous ship can contribute to the following areas:

- Emission reduction - ontologies can incorporate environmental data, fuel consumption models, and emission factors to optimize ship operations. By reducing fuel consumption, autonomous ships can significantly decrease their carbon footprint and contribute to global efforts in combating climate change.
- Eco-friendly route planning - using ontology and external interoperability, real-time data on weather conditions, ocean currents, and marine habitats can be obtained to plan routes that minimize the disruption of sensitive ecosystems. This eco-friendly route planning supports efforts to protect the marine environment by reducing ship impact.
- Compliance with regulations and green initiatives - with the growing awareness of sustainability, maritime regulations are increasingly focused on emission reduction and promoting eco-friendly practices. An ontological system can ensure compliance with these regulations and provide relevant agencies with data regarding the implementation of green initiatives.

As autonomous ships operate without direct human control, ethical and legal issues also arise. An ontological approach to the information infrastructure of an autonomous ship can result in the following contributions:

- Transparency of decision-making - ontologies can provide a clear understanding of the decision-making processes of autonomous ships. This transparency is crucial for building public trust in the technology and ensuring accountability for the actions taken by autonomous vessels.
- Accountability - in the case of accidents or incidents involving autonomous ships, determining accountability can be challenging. An ontological information infrastructure can keep records of actions taken by the autonomous system, facilitating investigations and attributing responsibility where necessary.
- Legal compliance - an ontology can be designed to incorporate legal frameworks and international maritime laws, ensuring that autonomous ships meet legal obligations, including collision regulations, maritime conventions, and the local laws of various regions they navigate through.

Let us proceed with the presentation of an ontology designed to improve fire protection onboard an autonomous ship.

5. ONTOLOGY OF AN AUTONOMOUS SHIP SYSTEM

The ontology presented here is based on the system developed for fire detection and suppression on an autonomous ship (Sumić et al., 2021, Sumic et al., 2021). The system has been developed as a substitution for the existing fire protection on a non-autonomous ship. The decision-making and action tasks within the fire protection activities in this system have been taken over by agents capable of monitoring and acting on the ship's fire protection elements. The effectiveness of this fire protection solution was explored through a simulation process. That simulation process analyzed a total of 650,880 different fires. All initiated fires were extinguished within a timeframe ranging from second 21 to second 384 after the fire's inception. The average time from ignition to complete fire extinguishment was 225 seconds. Figure 6 depicts a snippet of this simulation based on real parameter values relevant for the area studied (fire spread rate, firefighting element response times, firefighting foam expansion rate, and similar variables). The simulation results of the implemented model confirmed the hypothesis that multi-agent systems are applicable to decision-making in maritime emergency situations as autonomous executive systems or advisory systems, the implementation of which has a positive impact on maritime safety.

Ontology-based interoperability has been chosen as a means of achieving interoperability between the developed system and the systems in its environment. Therefore, a corresponding ontology has been created, encompassing all aspects of the developed system. The most important components of the created ontology are as follows:

- Fire detection and firefighting equipment – within this component, different pieces of firefighting equipment available onboard are defined, such as fire extinguishers, sprinklers, and fire alarms.
- Emergency procedures - within this component, step-by-step procedures in the event of a fire outbreak are defined. This includes communication protocols, fire suppression protocols, plans for potential evacuation, and coordination with external rescue services.

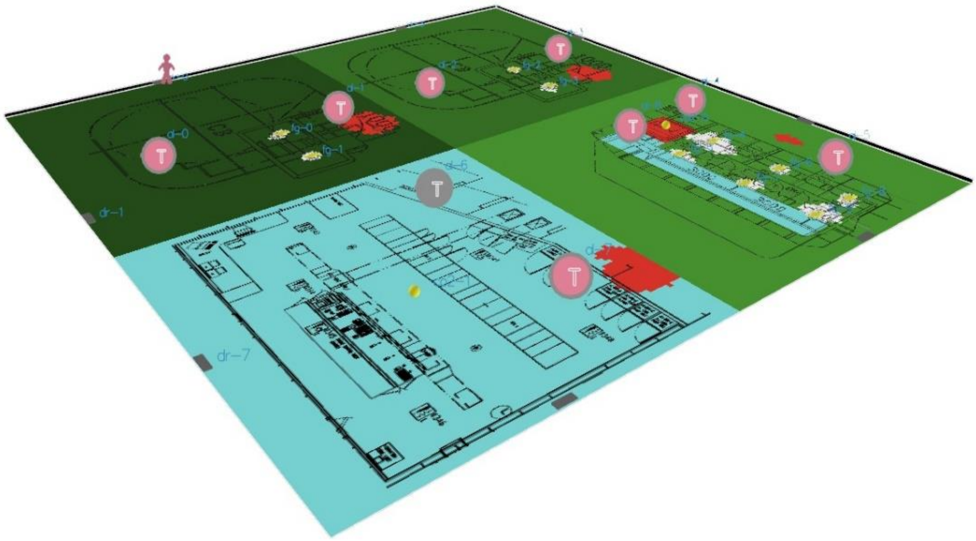

Figure 6. Snippet of a scene from the conducted simulation (Sumić et al., 2021)

The aforementioned components represent the knowledge that system agents use to make decisions related to fire protection and firefighting activities. The ontology also includes the following components:

- Classification of fire hazards - this component categorizes various types of fire hazards that an autonomous ship may encounter, including different ignition sources, such as electrical system failures, fuel leaks, or machinery overheating.
- Materials and substances - this component provides a detailed description of various materials and substances present on the ship, classifying them based on their flammability and potential impact on fire expansion.
- Risk assessment - this component assesses the potential risk of fire incidents based on the ship's configuration, cargo, and environmental conditions. It facilitates preventive planning and resource allocation.

Figure 7 is a simplified representation of a limited set of concepts from the created ontology related to ship fire protection elements. The concepts are structured through concept classes and class properties are listed. E.g., concept class *Compartments* possesses properties *permitted_method_of_extinguishing*, *CO2_quantity*, and *foam_quantity* through which allowable methods of extinguishing compartments are defined. These properties define the required number of CO2 units or fire foam units needed for fire suppression in a given space.

The *Compartments* class defines all ship compartments and that is the only class in this example that does not have its own subclasses. In the implemented system, an agent is assigned to each ship compartment which is responsible for monitoring and, if necessary, firefighting within that compartment. This agent has control over fire detection devices (such as infrared cameras, smoke detectors, and flame detectors) as well as firefighting equipment (CO2 and foam generators) located in the assigned room. Additionally, this agent has control over elements like light and sound alarms (in the case of vessels with human crew), air and fuel valves, voltage switches, and fire doors. All these elements are depicted as classes in Figure 7. At the highest level, class *Elements of environment* is defined, which includes all previously mentioned elements. Like any class diagram, inheritance rules for properties apply in this schema as well. Consequently, all environmental elements inherit all properties from their root class. The properties are as follows:

- Unique identifier of the element (*ID*)
- Identifier of the ship compartment where the element is located (compartment_ID)
- Coordinates of the element's location within the assigned ship compartment (X_coordinate, Y_coordinate)

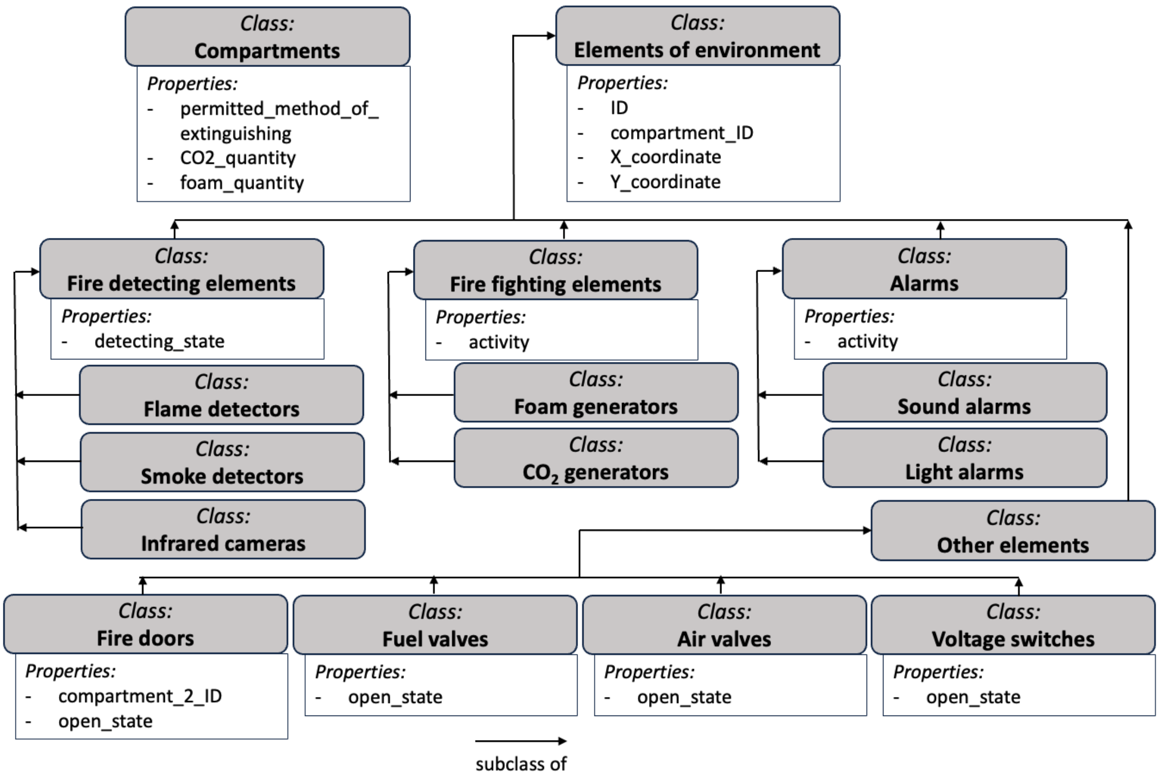

Figure 7. Part of the ship firefighting ontology

Each environmental element, in addition to these properties, possesses specific attributes related to the functionality of a given element. For instance, fire detection elements have the property *detecting_state* providing the agent with information about the current detection status. Firefighting elements have the *activity* property, which the agent uses to initiate the fire suppression process and similarly halt it once the fire is extinguished. Fuel and air valves, voltage switches, and fire doors feature the *open_state* property through which the agent reads whether the element is open or closed. By altering this property, the agent can interact within the environment. Fire doors have an additional property compartment_2_ID, which specifies the compartment to which the door leads.

An example that describes an ontology representation is given in the following text. In the example the ontology has three conceptual classes: *Compartments*, *Foam generators*, *Infrared cameras*. Additionally, the number of properties is limited. All classes possess an identification property, *ID*, while classes *Foam generators* and *Infrared cameras* have the property of being associated with an assigned ship compartment i.e., *compartment_ID*. The class *Infrared cameras* has the property *detecting_state*, while the class *Foam generators* has the property *activity* so that fire protection objects that are instances of these classes can communicate alarming states through these properties and receive orders to initiate fire suppression procedures.

The ontology is presented using a combination of RDF/RDF Schema/OWL. The ontology header, containing ontology name and the locations of the definitions of the used formats is given in Figure 8. Figure 9 shows class declarations, while Figure 10 shows class property declarations.

```
@prefix : <//ship_fire_protection#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

# Ontology declaration
<http://www.example.com/ship_fire_protection> rdf:type owl:Ontology ;
   rdfs:label "Ship fire protection" .
```

Figure 8. Ontology header

```
# Classes
:Compartments rdf:type owl:Class ;
   rdfs:label "Compartments" .

:FoamGenerators rdf:type owl:Class ;
   rdfs:label "Foam generators" .

:InfraredCameras rdf:type owl:Class ;
   rdfs:label "Infrared cameras" .
```

Figure 9. Classes declaration

To simplify, property value constraints are not depicted in the class property declarations in Figure 10. For instance, the property *detecting_state* of the *Infrared cameras* class could be defined to only accept values such as "*fire_in_compartment*," "*no_fire_in_compartment*," and "*raised_level_of_vigilance*". However, in this example, such constraints have not been implemented for the sake of simplicity. In the property declarations in the example presented, the properties can be set only to permitted values.

```
# Data Properties
:ID rdf:type owl:DatatypeProperty ;
   rdfs:label "ID" ;
   rdfs:domain :Compartments ,
            :InfraredCameras ,
            :FoamGenerators ;
   rdfs:range:integer .

:compartment_ID rdf:type owl:DatatypeProperty ;
   rdfs:label "compartment ID" ;
   rdfs:domain :InfraredCameras ,
            :FoamGenerators ;
   rdfs:range:integer .

:detecting_state rdf:type owl:DatatypeProperty ;
   rdfs:label "Fire detection status" ;
   rdfs:domain :InfraredCameras ;
   rdfs:range:string .

:activity rdf:type owl:DatatypeProperty ;
   rdfs:label "Activity" ;
   rdfs:domain :FoamGenerators ;
   rdfs:range:string .
```

Figure 10. Class property definitions

Class and property declarations are not sufficient to express all the knowledge agents need to access ontologies. Inference rules need to be defined as well. An example of a simple inference rule is the response to a signal indicating that an infrared camera has detected fire in a ship compartment. This example describes only one firefighting resource, namely foam. Therefore, following the receipt of the relevant infrared camera signal, the foam generator in the compartment where the fire was detected needs to be activated.

All communication of the firefighting agent with its environment occurs through the properties of the assigned fire protection elements. Therefore, the inference rule (and consequently, the action rule) needs to be defined using class properties. In this case, the following steps need to be taken:

1. Within the loop, read the *detecting_state* properties of *Infrared cameras*.
2. At the moment when the property detecting_state of an *Infrared camera* equals the value "*fire in compartment*", initiate the following action:
   2.1. For the *Foam generator* whose property *compartment_ID* matches the property *compartment_ID* of the *Infrared camera* that detected the fire, set the property *activity* to "*active*".

In a real-world scenario, the inference rule would be much more complex. The inclusion of more elements within the inference rule naturally leads to an increase in the complexity of the rule itself. Upon receiving a warning signal, other fire detection elements are also checked to confirm the presence of fire. After confirming the fire, many more complex activities are undertaken, involving almost all the previously defined elements of the ship's fire protection environment (fire doors are closed, alarms are activated, fire suppression procedures are initiated depending on the characteristics of the compartment where the fire occurred). The definition of the inference rule described above is given in Figure 11.

```
[ rdf:type owl:Class ;
  owl:equivalentClass
    [ rdf:type owl:Restriction ;
      owl:onProperty :fire_detection_status ;
      owl:hasValue :fire_in_compartment ;
      owl:onClass :InfraredCameras
    ] ,
    [ rdf:type owl:Restriction ;
      owl:onProperty :compartment_ID ;
      owl:hasValue ?compartment_ID ;
      owl:onClass :InfraredCameras
    ] ,
    [ rdf:type owl:Restriction ;
      owl:onProperty :activity ;
      owl:hasValue :active ;
      owl:onClass :FoamGenerators ;
      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ]
  ->
  [ rdf:type owl:Restriction ;
    owl:onProperty :compartment_ID ;
    owl:hasValue ?compartment_ID ;
    owl:onClass :FoamGenerators
  ] .
```

Figure 11. Inference rule definition

The ontology presented for the given example corresponds to the middle layer of the three-layer ontology model depicted in Figure 4, i.e., the ontology definition layer. One level below, at the ontology language level, the RDF and OWL have been used. An example of the top layer, the data layer represented by the ontology, is given in Figure 12. It shows the initial state of class instances from the previous example. Here, only one ship compartment referred to as SC001 is defined, containing infrared camera IC1 and foam generator FG1. In the initial state, the infrared camera does not detect fire, and the foam generator remains inactive.

```
# Individuals
:compartment1 rdf:type :Compartments ;
   :hasID SC001 .

:foamGenerator1 rdf:type :Foam_generators ;
   :hasID SG1 ;
   :hasCompartmentID SC001 ;
   :hasActivity "inactive" .

:infraredCamera1 rdf:type :Infrared_cameras ;
   :hasID IC1 ;
   :hasCompartmentID SC001 ;
   :hasDetectingState "no fire" .
```

Figure 12. Example of data presented by the ontology

6. CONCLUSION

The integration of autonomous ships into the maritime industry requires an interdisciplinary approach that addresses technical challenges, safety, security, environmental impact, ethical concerns, and socio-economic issues. An ontological information infrastructure can help overcome these barriers and unlock the full potential of autonomous navigation. By ensuring seamless data integration, promoting interoperability, and improving the decision-making processes, the ontological framework ensures safer, more efficient, and environmentally friendly autonomous maritime navigation. In this paper, the ontology of a ship firefighting system has been presented. It represents only a part of the set of ontologies that need to be developed for a complete ontological coverage of autonomous ship functionalities. A corresponding ontology needs to be developed for each ship subsystem. Methods of integrating such ontologies are an integral part of the three-layer ontology model. Through this integration, the collection of all developed ontologies can be considered a unified ontology for autonomous vessels. Only in such environment can the full potential of semantic interoperability between autonomous ship subsystems and interoperability between autonomous ships and external objects within the maritime traffic ecosystem be realized. Continuous technological development, collaboration between industry stakeholders, researchers, and decision-makers is crucial to establishing standards and best practices for ontological information infrastructures.

CONFLICT OF INTEREST

The authors declared no potential conflicts of interest with respect to the research, authorship and publication of this article.

REFERENCES

Allal, A. A., Mansouri, K., Youssfi, M. & Qbadou, M. Toward energy saving and environmental protection by implementation of autonomous ship. In: 19th IEEE Mediterranean Electrotechnical Conference (MELECON), 2018, Marrakech, Morocco. IEEE, pp. 177-180. Available at: http://dx.doi.org/10.1109/MELCON.2018.8379089.

Beckett, D. & Berners-Lee, T. 2008. Turtle -- Terse RDF Triple Language. Available at: https://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/.

Galieriková, A. 2019. The human factor and maritime safety. Transportation Research Procedia, 40, 1319-1326. Available at: http://dx.doi.org/10.1016/j.trpro.2019.07.183.

Gomez-Perez, A. & Corcho, O. 2002. Ontology languages for the Semantic Web. IEEE Intelligent Systems, 17, pp. 54-60. Available at: http://dx.doi.org/10.1109/5254.988453.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. Knowledge acquisition, 5, pp. 199-220. Available at: http://dx.doi.org/https://doi.org/10.1006/knac.1993.1008.

Gu, Z. & Zhang, S. 2021. DL-Lite Full: A Sub-language of OWL 2 Full for Powerful Meta-modeling. In: Martin Homola, Vladislav Ryzhikov & Schmidt, R. (eds.) Proceedings of the 34th International Workshop on Description Logics. Bratislava, Slovakia.

Hogg, T. & Ghosh, S. 2016. Autonomous merchant vessels: examination of factors that impact the effective implementation of unmanned ships. Australian Journal of Maritime & Ocean Affairs, 8, pp. 206-222. Available at: http://dx.doi.org/10.1080/18366503.2016.1229244.

IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, 1991. IEEE Std 610, pp. 1-217. Available at: http://dx.doi.org/10.1109/IEEESTD.1991.106963.

Im, I., Shin, D. & Jeong, J. 2018. Components for Smart Autonomous Ship Architecture Based on Intelligent Information Technology. Procedia Computer Science, 134, pp. 91-98. Available at: http://dx.doi.org/10.1016/j.procs.2018.07.148.

IMO 2021. Outcome of the regulatory Scoping Exercise for the use of Maritime Autonomous Surface Ships (MASS). In: IMO (ed.) MSC.1/Circ.1638. London: International Maritime Organisation.

Kim, M., Joung, T.-H., Jeong, B. & Park, H.-S. 2020. Autonomous shipping and its impact on regulations, technologies, and industries. Journal of International Maritime Safety, Environmental Affairs, and Shipping, 4, pp. 17-25. Available at: http://dx.doi.org/10.1080/25725084.2020.1779427.

Kurt, I. & Aymelek, M. 2022. Operational and economic advantages of autonomous ships and their perceived impacts on port operations. Maritime Economics & Logistics, 24, pp. 302-326. Available at: http://dx.doi.org/10.1057/s41278-022-00213-1.

Males, L., Sumic, D. & Rosic, M. 2022. Applications of Multi-Agent Systems in Unmanned Surface Vessels. Electronics, 11. Available at: http://dx.doi.org/10.3390/electronics11193182.

Males, L., Sumic, D., & Rosic, M. . Toward interoperability of multi-agent systems on autonomous ships. In: Telmoudi, A. J., ed. 9th International Conference on Control, Decision and Information Technologies (CoDIT) 3-6 July 2023 2023 Rome, Italy. IEEE, pp. 422-426. Available at: http://dx.doi.org/10.1109/CoDIT58514.2023.10284374.

Mishra, R. B. & Kumar, S. 2011. Semantic web reasoners and languages. Artificial Intelligence Review, 35, pp. 339-368. Available at: http://dx.doi.org/10.1007/s10462-010-9197-3.

Munim, Z. H., Saha, R., Schøyen, H., Ng, A. K. Y., & Notteboom, T. E. 2022. Autonomous ships for container shipping in the Arctic routes. Journal of Marine Science and Technology, 27, pp. 320-334. Available at: http://dx.doi.org/10.1007/s00773-021-00836-8.

Pérez, J., Arenas, M. & Gutierrez, C. 2009. Semantics and complexity of SPARQL. ACM Trans. Database Syst., 34, Article 16. Available at: http://dx.doi.org/10.1145/1567274.1567278.

Rødseth, Ø. J. From concept to reality: Unmanned merchant ship research in Norway. In: 2017 IEEE Underwater Technology (UT), 21-24 Feb. 2017, Busan, Korea (South). IEEE, pp. 1-10. Available at: http://dx.doi.org/10.1109/UT.2017.7890328.

Sumić, D., Maleš, L., & Rosić, M. 2021. Agent Based Onboard Firefighting System. Transactions on Maritime Science, 10, pp. 101-111. Available at: http://dx.doi.org/10.7225/toms.v10.n01.007.

Sumic, D., Males, L., & Rosic, M. 2021. An Agent-Based Ship Firefighting Model. Journal of Marine Science and Engineering, 9. Available at: http://dx.doi.org/10.3390/jmse9080902.

Swartout, B., Patil, R., Knight, K. & Russ, T. Toward distributed use of large-scale ontologies. In: Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems, 1996, p. 25. Available at:

Ter Horst, H. J. 2005. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics, 3, pp. 79-115. Available at: http://dx.doi.org/10.1016/j.websem.2005.06.001.

Udrea, O., Recupero, D. R. & Subrahmanian, V. S. 2010. Annotated RDF. ACM Trans. Comput. Logic, 11, Article 10. Available at: http://dx.doi.org/10.1145/1656242.1656245.

Ziajka-Poznańska, E. & Montewka, J. 2021. Costs and Benefits of Autonomous Shipping—A Literature Review. Applied Sciences, 11, p. 4553. Available at: http://dx.doi.org/10.3390/app11104553.