

# Agent Based Onboard Firefighting System

Dean Sumić<sup>a</sup>, Lada Maleš<sup>b</sup>, Marko Rosić<sup>c</sup>

This paper presents a model of agent-based architecture for fighting fires on ships. The introduction of agent technology in firefighting decision-making is a step towards safe autonomous vessels.

The human factor can be excluded through the introduction of agent-based technology for the detection and extinguishing of fires onboard ships. The aim is to reduce the number of injuries and deaths, and minimize loss of ships and cargo. Another advantage of agent-based technology is its easy interoperability with other automated onboard systems. The presented model was implemented on a prototype in a simulation environment. The results of the experiment conducted on the implemented prototype are also presented.

## KEY WORDS

- ~ Fires on ships
- ~ Agents
- ~ Autonomous vessels

a. University of Split, Faculty of Maritime Studies, Split, Croatia

e-mail: dean.sumic@pfst.hr

b. University of Split, Faculty of Humanities and Social Sciences, Split, Croatia

e-mail: lada.males@ffst.hr

c. University of Split, Faculty of Science, Split, Croatia

e-mail: marko.rosic@pmfst.hr

doi: 10.7225/toms.v10.n01.007

This work is licensed under 

Received on: Feb 23, 2020 / Revised on: Apr 2, 2021 / Accepted on: Apr 17, 2021 /

Published: Apr 20, 2021

## 1. INTRODUCTION

The number of maritime incidents is still high and technology currently used in all ship subsystems has not succeeded in significantly reducing the risk of accidents. An important reason is that the ship is a system consisting of elements one of which are humans (Bielić, Vidan, & Mohović, 2010). Fires on board are one of the basic classes of distress situations on board. They are one of the leading causes of loss of ships, cargo or injuries to people. The ship's engine room is a place where fuel, oils, high voltage, pressurized steam, boilers with burners, internal combustion engines, electricity generators, oil and fuel separators, and incinerators are located and as such it represents the greatest fire hazard onboard. When fires onboard are analyzed, people emerge as risk factors, sometimes as fire causes and sometimes as firefighting entities.

At the moment, a built-in fixed fire system or systems are manually operated by human beings in case of fire. This requires both crew training and their mutual communication, which is especially important in incident situations. Modern ship crews form a multicultural environment, as they have different native languages and frequently encounter language difficulties.

After an alarm is sounded, the crew must check the status of the fire and determine whether the fire has really occurred or not. If the fire detection alarm was correct, after onsite assessment of the fire, the crew sound the fire alarm, arousing the fire brigade into action. Crew from the fire department need to be fully equipped before springing into action. This includes dressing up, putting on or taking off the breathing apparatus, shoes, clothes, gloves, helmets and various tools. The equipped firefighting team must then meet at the agreed collection station (Muster Station). The ship usually has more than one such station and each crew member should know where to come depending on his assignment. In practice, the crew is divided into different

teams such as the incident team, the bridge team or the engine room team. The speed of their response to a fire alarm depends on the training and level of training of crew members, which is often a weak link in real-time response and decision-making (Wikman et al., 2017) (Zhang, Zhan, & Tan, 2009).

The crew must be capable of making decisions and acting in the event of any other distress situation that may arise as a result of heavy weather, equipment failure, piracy attack or human error. Though methods of action are strictly defined, errors still occur that result in either wrong decisions or untimely reactions, which can cause loss of life and property. The human factor is often the cause of ship fires. People can also be injured either in fires or while fighting them. Likewise, the crew may be insecure in its actions due to inappropriate decisions, violation of normal procedures, lack of practice and fatigue (Akyuz, 2017).

This opens the space for the implementation of agent technology with the intention to ensure response in line with the prescribed procedures, while eliminating the risks associated with people, fatigue, untimely action or communication problems. It also eliminates the problem of potentially irrational reactions and unreasonable decision-making by individuals who have found themselves in a life-threatening situation and succumbed to panic.

In this paradigm, the agent does not change the legally prescribed procedures of action, but guarantees their timely application. Therefore, the agent is a reactive agent that gets information about the environment from sensors, and takes action on the basis of built-in knowledge (prescribed operating procedures) by activating the appropriate elements of the ship's fire system. Such an agent is always alert, monitoring the state of the assigned environment and strictly following the prescribed procedures in the event of an incident in the shortest time possible. This agent, in conjunction with additional sensors, can also reduce the number of false positive alarms and thus further increase safety levels.

This approach and the introduction of agent-based decision-making technology in the detection and firefighting process is also a contribution to the construction of autonomous ships. The trend of crew reduction has long been present in the maritime industry and with the introduction of MASS (Maritime Autonomous Surface ships) it gained new momentum, increasing the possibility of unmanned ships in the near future. However, these ships will not sail completely independently for some time, but will have MASS operators on shore as supervisors, remote control system managers, voyage planners or simply decision makers (Ramos, Utne, Vinnem, & Mosleh, 2018) who will be active operators whenever the STCW (Standard Training for Watchkeeping) legislation needs to be changed. Although this is positive from the point of view of employing people and opening new high-tech positions, these operators are in turn becoming a new human risk factor, and research is focused on the development of risk analysis models for MASS operators

where fuzzy logic can be used to avoid collisions (Wu, Yip, Yan, & Soares, 2019). A systematic approach whereby ships would become MASS in some studies requires a program that would verify the required level of safety onboard ships, as well as the introduction of STPA (Systems-Theoretical Process Analysis) to identify risks that may occur on ships as a result of deck officers' or engineers' decisions, as scenarios that can be evaluated using the OCA System (Online Consequence Analysis System) (Rokseth, Utne, & Vinnem, 2017).

## 2. AGENTS

Wooldridge and Jennings define agents as entities that have the following characteristics (Wooldridge & Jennings, 1995):

- **Autonomy:** Agents act without direct user intervention or computer system components and have control over their actions and internal status,
- **Ability to communicate** with other agents or with the user,
- **Reactivity:** Agents perceive their environment (which can be a real world, collections of other agents, the Internet or some other environment, in this case ship systems) and react to changes,
- **Proactivity:** In order to accomplish their task, agents are not only responsive to changes in the environment but are also capable of taking initiative by initiating changes in the environment.

A software agent is a computer similar to an autonomous robot. Program agents can be designed to be stand-alone or work in collaboration with other agents or humans. In their interactions with humans, agents may possess qualities characteristic of humans, such as understand human language and speech, have a personality, or humanoid embodiment. Such entities are an evolutionary step forward compared to conventional computer programs. They can be activated and run independently and do not require input or interaction with a human user. These agents can also run, monitor, and terminate other programs, agents, or devices in their environment. The key elements that distinguish agents from conventional programs are their ability to respond to changes in the environment, autonomy, goal orientation, and persistence (Franklin & Graesser, 1997).

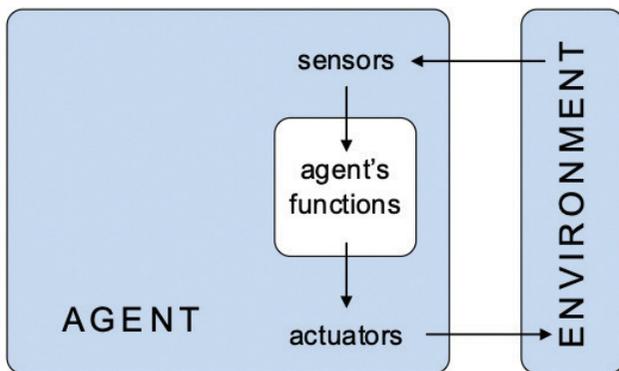
A characteristic that distinguishes agents from a multitude of other artificial systems, including computer programs and electromechanical robots, is the fact that agents are capable of adopting principles of behavior to some extent (Steels, 1995). Basically, we can perceive an agent through his perception of the environment through sensors and action in and on such environment through an effector (Russell & Norvig, 2020). A schematic representation of this idea is given in Figure 1. The term perception is used for all perceptual inputs of an agent over time. The sequence of observations is the entire history of everything the agent has ever observed. The choice of action taken by an

agent at any given time may depend on the last observation or on a whole sequence of observations. If the choice of agent action for each possible sequence of observations can be determined, then its functionality is fully defined.

Mathematically speaking, it can be said that the behavior of an agent is described by an agent function that maps any sequence of observations into appropriate action. If  $P^*$  is the set of all sequences of observations of arbitrary length, and  $A$  the set of actions that an agent can perform, the agent function can be written as:

$$f: P^* \rightarrow A \quad (1)$$

Agent function is an abstract concept that can contain various decision-making principles such as, among other things, calculating the usefulness of individual options, deduction based on logical rules and fuzzy logic (Šalamon, 2011).



**Figure 1.**  
Schematic representation of the agent.

Russel and Norvig listed five basic types of agents by the degree of expressed intelligent behavior and ability (Russell & Norvig, 2020):

- Simple Reflex Agents that select action on the basis of a current observation, ignoring the history of observation,
- Model-Based Reflex Agents that select the action based on the history of observations,
- Goal-Based Agents who choose the action to achieve a specific goal,
- Utility-Based Agents that select the action that will achieve the maximum degree of success and
- all these agents can also be Learning Agents who analyze the experience to choose the action.

The same authors (Russell and Norvig, 2020) emphasize the importance of environmental analysis when designing agent systems. Significantly less attention is paid to the nature

of the environment in which the designed solution operates. It turns out, however, that the characteristics of the environment are one of the key elements for choosing the right solution model. There are several aspects that affect the characteristics of the environment. The nature of the problem, the scope of available information and knowledge about the environment, and changes occurring in the environment through the action of the agent are only some of the elements that determine the nature of the environment. Understanding the characteristics of the environment is one of the first issues that need to be tackled to deal with the problem of the agent paradigm in the field of artificial intelligence.

Although agents can act individually to solve a particular problem, agent systems frequently consist of several agents. Most real-world problems require or involve multiple agents to account for the decentralized nature of the problem, provide multiple perspectives, or present competing interests (Panait & Luke, 2005). Systems involving multiple autonomous agents with the ability to interact and communicate are called Multi-Agent Systems (MAS) (Jennings & Wooldridge, 1998).

The basic characteristics of multi-agent systems are as follows:

- each agent has incomplete information or problem-solving capabilities and, based on these limitations, his own view of the problem,
- there is no global control of the system,
- data is decentralized,
- calculations are asynchronous.

Interaction in a multi-agent system occurs when two or more agents are brought into dynamic interaction through a set of mutual actions. Interactions develop from a series of actions the consequences of which affect the future behavior of agents. During interactions, agents are in contact with each other, whether that contact is direct or takes place through another agent or through environment (Ferber, 1999).

Agent collaboration can be defined as the successful action of one agent (or group of agents) aided by the action of another agent (or group of agents). The cooperation of agents in a multi-agent system presupposes compatible or common goals. In this case, agents may cooperate in an attempt to achieve a goal that would be difficult to realize individually (Cavezzali, Girotti, & Rabino, 2003). This is an example of positive interaction in which the addition of a new agent allows a marked increase in the level of performance of the group or the avoidance and/or resolution of potential or actual conflicts. The type of cooperation depends on the degree of interdependence of their activities, as well as on whether the capabilities of individual agents are sufficient to achieve the goals, or whether there are sufficient resources in the environment.

The agent paradigm can be applied very effectively in combination with another new, exciting and advanced technology,

which is rapidly entering all aspects of human life and work - the Internet of Things (IoT). Physical objects, fully integrated into the information highway, that receive an abundance of identifying, historical, and descriptive data through their sensors, as well as data on their position and environment (Cooper & James, 2009). By adding intelligence, possessed by agents, to everyday objects connected to the Internet, the concept of the Internet of Things takes on a new, higher dimension, and the Internet of Things becomes an Internet agent (Yu, Shen, & Leung, 2013) (Fortino, Gravina, Russo, & Savaglio, 2017). For the purposes of this paper, i.e. the use of agents as firefighting elements onboard ships, it is important to emphasize that the Internet of Things technology can be used in a local network that does not necessarily have an Internet connection, and therefore in the ship's information infrastructure.

### 3. MODEL OF FIREFIGHTING AGENT-BASED SHIP ARCHITECTURE

In this model the ship is considered to be a set of rooms containing elements relevant for fire detection and extinguishing (Sumić, 2021). These elements are called environmental elements. An agent is assigned to each room and it receives an image of the world (ship) through readings of the properties of the elements of the environment, for example by reading the state of various detectors. All agents operate within the environment using elements that have the ability to change that environment, such as various valves or generators of extinguishing agents. Each environmental element belongs to an environmental element class, and each environmental element class has associated properties characteristic of that class.

The model of firefighting agent-based ship architecture is defined as follows. First, the environment is defined by defining the space that the environment takes up, i.e. the environment consists of spatial units that constitute that environment.

The set  $PJO = \{ pjo_1, pjo_2, \dots, pjo_m \}$ ,  $m \in N$  is called *the set of basic spatial units of the environment*. The elements  $pjo_i$ ,  $1 \leq i \leq m$  are called *the basic spatial units of the environment*.

A basic spatial unit is therefore the smallest spatial unit that we can address, and the environment space is defined as sets created from basic spatial units. During model development, this approach was chosen to increase the accuracy of detection of locations where a disturbance in the environment occurred. This approach allows us to identify the location of a fire inside a room or the location of an injured person inside a room, as opposed to a system where location is identified at room level. The set of basic spatial units of the  $PJO$  must therefore be distributed across the rooms in the environment, i.e. the basic spatial units of the environment are grouped into the rooms of that environment.

Let  $PJO$  be a set of basic spatial units of the environment. Each partition of the  $PJO$  set is called *the set of environment rooms* and is denoted by  $P(PJO)$ . Each element of the set  $P(PJO)$  is called *the environment room*.

The environment rooms contain elements belonging to different classes, such as *Valves, Doors, Sound Alarms, Light Alarms, Smoke Detectors, Infrared Cameras, CO<sub>2</sub> Generators and Foam Generators*. The class of environment elements will be denoted by  $Cee$ .  $CEE = \{ Cee_1, Cee_2, \dots, Cee_n \}$ ,  $n \in N$ , is a set of classes of environment elements. Element  $e \in Cee$  is called a *Cee class element of the environment*.

Certain properties must be attributed to the elements of the environment and the allowed values of such properties must be defined. Therefore, the structure property-values is defined in order to assign the appropriate values to each type of environment element. In order to associate relevant properties and values with the appropriate types of environment elements, structure property-values is defined above the class of environment elements. For example, the Doors and the Smoke Detector are both environment elements but each of them requires different properties to be observed.

*The property-values structure over Cee class environment elements* is each a subset of the set

$$\{ (a,b,c) \mid a \in Cee; b,c \in string[30] \} \quad (2)$$

where  $string[30]$  is a set of all strings with the maximum length of 30 characters.

The structure property-values over the  $Cee$  class of environment elements is denoted by  $SV(Cee)$ . The first component of an ordered triple  $(a,b,c)$  from the property-value structure is called *the target element of the environment*, the second component is called *the property name*, while the third component is called *the property value*. For  $e \in Cee$  those elements of the set  $SV(Cee)$  for which the first component is the element  $e$  will be called *the properties of element e*.

Table 1 shows an example of one interpretation of the structure property-values over the classes of environment elements from the set  $\{Valves, Doors, Sound Alarms, Light Alarms, Smoke Detectors, Infrared Cameras, CO_2 Generators, Foam Generators\}$ .

Just as it is necessary to assign certain values to the elements of the environment, it is also necessary to assign appropriate properties to the rooms and their values, such as, for example, identifiers of fire-extinguishing means allowed to be used in a given room and similar properties. Therefore, the structure property-values over the set of environment rooms has to be defined.

**Table 1.**

Classes of environment elements with corresponding structures property - value.

Environment element class name	Property name	Allowed property values
<b>Valves</b>	valve_type	„CO <sub>2</sub> “, „foam“
	coordinates	ID of the basic spatial unit of the environment in which the element is located
	openness_state	„open“, „closed“
<b>Doors</b>	coordinates	ID of the door interconnecting two rooms
	openness_state	„open“, „closed“
<b>Sound Alarm</b>	coordinates	ID of the basic spatial unit of the environment in which the element is located
	activity	„alarm active“, alarm inactive“
<b>Light Alarm</b>	coordinates	ID of the basic spatial unit of the environment in which the element is located
	activity	„alarm active“, alarm inactive“
<b>Smoke Detector</b>	coordinates	ID of the basic spatial unit of the environment in which the element is located
	detection_state	smoke level detection
<b>IR Camera</b>	coordinates	ID of the basic spatial unit of the environment in which the element is located
	fire_detection_state	„fire in space“, „without fire in space“, „increased level of caution“
	person_presence_detection_state	number of detected persons in space, „caution – indefinite state“
<b>CO<sub>2</sub> Generator</b>	activity	„in the process of generating“, „inactive“
<b>Foam Generator</b>	activity	„in the process of generating“, „inactive“

The structure property – values over the set of environment rooms  $P(PJO)$  is each subset of the set

$$\{ (a,b,c) \mid a \in P(PJO); b,c \in string[30] \} \quad (3)$$

where  $string[30]$  is a set of all strings with the maximum length of 30 characters.

Structure property - values over the set of environment rooms  $P(PJO)$  is denoted by  $SV(P(PJO))$ . The first component of the ordered triple  $(a,b,c)$  is called *the target room*, the second component is called *the property name*, while the third component is called *the property value*. Table 2 shows an example of one interpretation of the structure property - values over a set of environment rooms.

**Table 2.**

Example of interpretation of the structure property - values over a set of environment.

	Property name	Allowed property values
<b>Room</b>	room_ID	Room Identification Number
	fire_detection_state	„fire in space“, „without fire in space“, „raised level of caution“
	allowed_extinguishing_method	"CO <sub>2</sub> ", "foam", "foam & CO <sub>2</sub> "
	people_presence	number of people in room
	CO <sub>2</sub> _level	number of CO <sub>2</sub> generators

One of the properties of the room is the presence of people. Prior to initiating the firefighting procedure, the appropriate method of extinguishing needs to be determined, taking into account whether there are people in the area under fire or not. Also, in some cases it is not even allowed to start the extinguishing process before the evacuation of all persons from the area engulfed in flames. The value of this property changes in real time depending on the change in the value of the corresponding detectors.

The following is a definition of the set of all elements of the environment.

The set  $EE(CEE) = \{e \in Cee : Cee \in CEE\}$  will be called *the set of all environment elements from the CEE classes of environment elements*.

The  $EE(CEE)$  set contains all the elements of the environment regardless of which class of elements of the environment they belong to. The elements of the set  $EE(CEE)$  now need to be associated with the premises of the environment, i.e. we need to define where each of the elements of the environment is located.

Function  $D_{eo} : EE(CEE) \rightarrow P(PJO)$  is called *the function of assignment of environment elements to environment rooms*.

Function  $D_{eo}$  assigns an environment element e.g. *Smoke Detectors* or *Infrared Cameras* to rooms where those elements are located.

Having provided the above definitions, the fire environment of the ship can now be defined:

The firefighting environment of the ship is 5-tuple.

$$O = (P(PJO), EE(CEE), SV(P(PJO)), SV(CEE), D_{eo}) \quad (4)$$

where:

- $P(PJO)$  – set of environment rooms,
- $EE(CEE)$  - set of all environment elements from the  $CEE$  classes of environment elements,
- $SV(P(PJO))$  - structure property - values over the set of environment rooms  $P(PJO)$
- $SV(CEE)$  - structure property - values over  $CEE$  classes of environment elements,
- $D_{eo}$  - function of assignment of environment elements to environment rooms.

Now an agent needs to be introduced into the model:

Agent  $a$  is an ordered triple  $(K, Ac, Z)$  where:

- $K$  - agent's knowledge of the environment,
- $Ac$  - actions by which the agent can act within the environment and

- $Z$  - rules based on which the agent decides which action to take in the environment.

To simplify the model, we limit it to the case where each environment room will be assigned to one agent.  $k$  is the cardinal number of the set  $P(PJO)$ . We define the set  $A$  of all agents, i.e.  $A = \{a_1, a_2, \dots, a_k\}$  where  $a_i$  is the agent,  $1 \leq i \leq k$ .

Now it is necessary to define a function that will assign an agent to each space.

Each bijection  $D_a : A \rightarrow P(PJO)$  is called *the function of assigning agents to environment spaces*.

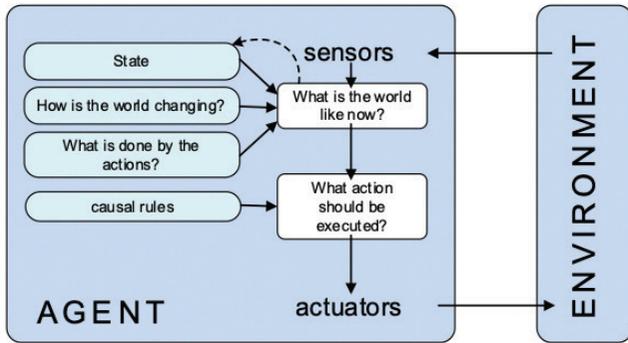
Function  $D_a$  assigns to each environment room from  $P(PJO)$  one agent tasked with fire protection. The environment of this agent is the environment space assigned to that agent. If we assume that the necessary infrastructure is provided and the agent is capable of monitoring the state of all elements of the environment in the room assigned to it in real time, the agent's knowledge  $K$  of the environment can be identified with the properties of all elements of the environment located in the room.

It is similar with the agent's actions  $Ac$ . Assuming the necessary infrastructure is provided and the agent communicates with the elements of the environment within the room, the actions of the agent represent changes in those states of the elements (i.e. formally the values of their properties) that have the ability to change the state of the environment. For example, if there is a valve in the room to which the agent is assigned, the agent's actions include the possibility to act on the property *openness\_state* of the observed valve, which is certainly an action by which the agent can act on his environment.

The manner in which the rules  $Z$  based on which the agent decides which actions to take within the environment will be determined, depends on the interpretation of the model. These rules include the state of the environment elements of the room to which the agent is assigned. For example, these rules can be defined by production rules, such as the following production rule:

```
if smoke_detector.detection_state = "smoke in the room" then
  {if (ir_camera.fire_detection_state = "fire in space" or
  ir_camera.fire_detection_state = "raised level of caution")
  then {
    sound_alarm.activity = "active"
    light_alarm.activity = "active"
    ...}}
```

This behavior of a model-based reactive agent scheme is shown in Figure 2.



**Figure 2.**  
Schematic representation of a model-based reactive agent.

It is now possible to define a ship's firefighting agent architecture.

The firefighting agent architecture of the ship is 7-tuple

$$(P(PJO), EE(CEE), SV(P(PJO)), SV(CEE), D_{eo}, A, D_a) \quad (5)$$

where:

- $P(PJO)$  - set of environment rooms,
- $EE(CEE)$  - set of all environment elements from the  $CEE$  classes of environment elements,
- $SV(P(PJO))$  - structure property - values over the set of environment rooms  $P(PJO)$ ,
- $SV(CEE)$  - structure property - values over the classes of environment elements  $CEE$ ,
- $D_{eo}$  - function of assignment of environment elements to environment rooms,
- $A = \{a_1, a_2, \dots, a_m\}$  - set of agents,
- $D_a$  - function of assigning agents to environment rooms.

#### 4. THE EFFECTIVENESS OF THE SHIP FIREFIGHTING AGENT ARCHITECTURE MODEL

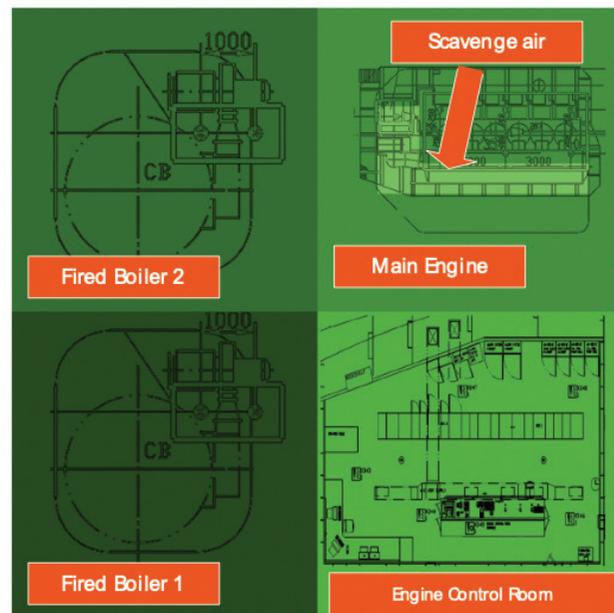
Most real systems are inherently variable, interconnected, and complex. It is difficult to predict the behavior of systems that are subject to either variability, interconnection, or complexity, and it is almost impossible to do the same for systems that are potentially subject to all of the above. Simulation models, however, can explicitly represent the variability, interconnectivity, and complexity of a system. As a result, it is possible to predict system performance by simulation, compare alternative performances, as well as to determine the effect of alternative rules on system performance. Therefore, simulation

is often a convenient approach to system modeling. In order to experimentally confirm the effectiveness of the designed model of onboard firefighting agent architecture, a software simulation environment based on a defined model was developed.

For simulation purposes, the layout of a real ship was used. The following rooms were selected:

- fired boiler room 1 and fired boiler room 2. These rooms are redundant.
- main engine room,
- engine control room,
- main engine scavenge air space.

Figure 3 shows the layout of selected rooms in the simulation environment. Each room was assigned an agent tasked with autonomous fire protection of the room. The model was implemented in a discrete simulation where dynamic steps were programmed in the time units of the simulation that correspond to the real time relationships required to perform a particular action. In each time unit of the simulation, each agent detected possible fire occurrences in its designated room, checked whether the fire was real or whether the false state of the elements of the smoke or flame detector environment triggered a false fire alarm that alerted the room agent. Based on the fire detection status decision, the room agent made extinguishing decisions depending on the room properties, including the permitted extinguishing method ( $CO_2$  or foam in the model), human presence,  $CO_2$  level, foam level and maximum foam filling period for the room.



**Figure 3.**  
Layout of selected rooms in the simulation environment.

Figure 4 shows which elements of the environment were assigned to the simulation. Elements presented in Figure 4 correspond to the model from Chapter 3. For example, the following statements are true for the defined model:

- $Fired\ boiler\ room\ 1 \in P(PJO)$ ,
- $CEE = \{ IR\ monitor, Flame\ detector, Smoke\ detector, Foam\ generator, Door, Valve \}$ ,
- "0406"  $\in Flame\ detector$ ,
- $D_{eo}(0406) = Fired\ boiler\ room\ 2$ ,

- $D_a(Agent\ 1) = Fired\ boiler\ room\ 1$ ,
- $(Fired\ boiler\ room\ 1, allowed\ extinguishing\ method, foam) \in SV(P(PJO))$ .

The developed application allows fire to be simulated at any location and under any initial conditions of parameters shown in Table 1 and Table 2. It also allows the simulation of cases of occurrence of multiple parallel fires in multiple rooms. Figure 5 shows a time section of a fire simulation in all rooms when the alarms were activated and the extinguishing procedure initiated.

CLASSES OF ENVIRONMENT ELEMENTS								
ROOM	IR MONITOR	FLAME DETECTOR	SMOKE DETECTOR	FOAM / CO2 GENERATOR	DOOR	VALVE	GMDSS	AGENT
FIRE BOILER 1	IR 1	0404	0403	UFG90 X 2	DOOR1 DOOR2	FUEL FLAPS POWER		1
FIRE BOILER 2	IR 2	0406	0405	UFG90 X 2	DOOR2 DOOR4	FUEL FLAPS POWER		2
MAIN ENGINE	IR 3	0412	0411	UFG90 X 6	DOOR5 DOOR6	FUEL FLAPS POWER		3
ECR	IR 4		0344 0346	CO2 NOZZLE	DOOR7 DOOR8	FUEL FLAPS POWER		4
ME SCAVENGE AIR	IR 5		N/A	CO2 NOZZLE		FUEL		5

ELEMENTS OF THE ENVIRONMENT

Figure 4. Elements of the environment managed by agents within the simulation.

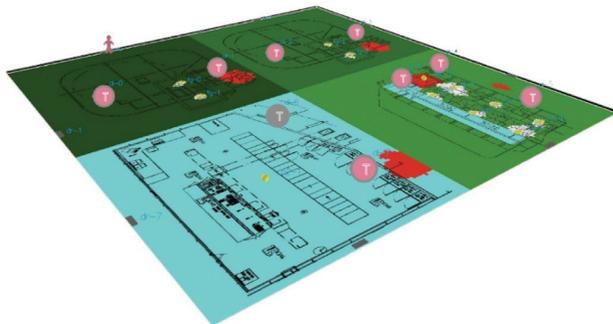


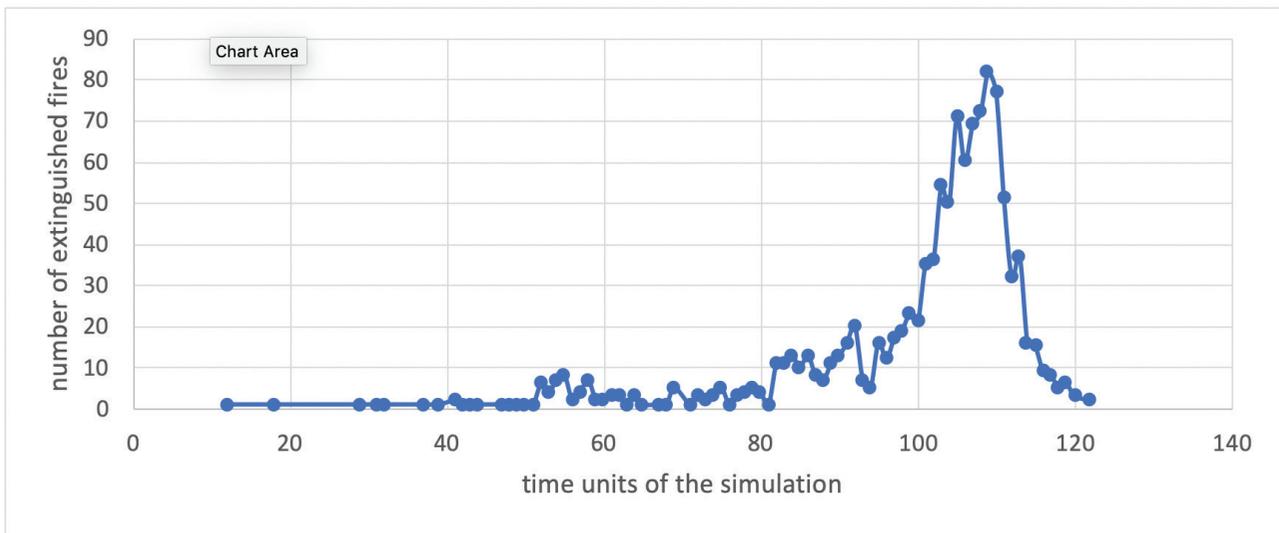
Figure 5. Example of simulation of fire in all rooms.

In (Sumić, 2021), a comprehensive experiment was conducted in which all combinations of single and double fires (when a second fire occurs independently from and simultaneously with the first fire) that could occur in the modeled environment, in terms of fire location and different states of environmental elements, were simulated. Figure 6 shows the results of the experiment limited to a single room and single fire that can occur anywhere and under any initial conditions. This figure shows an example of the fired boiler room where as many as 1,152 different fires can be detected. The number of different fires that can occur in fired boiler room 1 is determined using the following parameters:

- the number of locations in the room where a fire could start (16 different states),
- the state of the power valve (open or closed - 2 different states),
- the state of the fuel valve (open or closed - 2 different states),
- the state of the flaps valve (open, closed or partially open - 3 different states),
- the state of the door (open, closed or partially open - 3 different states),
- the presence of people in the room (people in the room, no people in the room – 2 different states).

Multiplying all of these states, we found that 1,152 different fires could be simulated in the developed simulation environment.

Fires inside this room were grouped depending on the time it took to put them out. The majority of fires (exactly 726 out of 1152) were extinguished between 101st and 113th time unit of the simulation, corresponding to the real extinguishing time of 303 to 339 seconds. The maximum time required to extinguish a fire in the fired boiler room no. 1 was 366 seconds. The mean time required to put out a fire in this room was 299.7 seconds, while the median was 315 seconds.

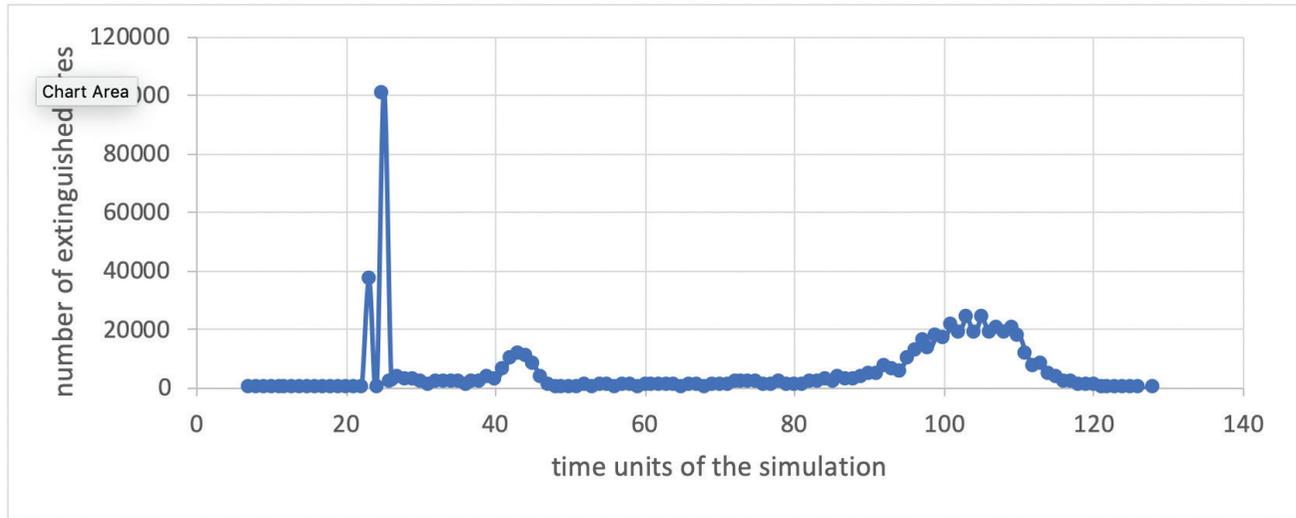


**Figure 6.**

Simulation of all fires that occurred in fired boiler room no. 1: number of extinguished fires depending on the time required to extinguish them.

If all combinations of single and double fires are taken into consideration, the number of fires increases to 650,880 different fires. The number of double fires simulated in the simulation environment was calculated in the same way as the number of single fires. In the simulation, a program was created that triggered all these fires. For each of the fires, any change in condition was recorded. Figure 7 shows the times required to extinguish all 650,880 simulated fires. The conducted experiment showed that all fires in the simulation were indeed extinguished. The first fires were extinguished in the 7<sup>th</sup> time unit of the simulation (three fires), and the last in the 128<sup>th</sup> time unit of the simulation (two fires), corresponding to the complete

extinguishing time of 21-384 seconds from fire occurrence. The discrete moments shown in the 23<sup>rd</sup> time unit of the simulation show that 37,058 fires were extinguished in that time unit, while the discrete moment in the 25<sup>th</sup> time unit of the simulation shows that that is when 101,320 fires were extinguished. These times refer to putting the fires out with CO<sub>2</sub> because the moment of extinguishing a fire with this agent is predictable considering the moment of activation. The average time from beginning to completion was 75 simulation time units, corresponding to the real time of 225 seconds. A detailed analysis of all simulated fires, as well as complete statistical data are provided in (Sumić, 2021).



**Figure 7.**  
Time required to extinguish all 650, 880 simulated fires.

## 5. CONCLUSIONS

The paper presented an agent-based onboard firefighting model and a prototype application that simulates the behavior of this model in real situations was developed. The application included all the elements of the developed model for firefighting agent architecture onboard ships. It activated the appropriate type of agents, i.e. model-based reactive agents, that were assigned to each room of the ship to monitor, detect and manage fire extinguishing. The developed application allows the setting of initial conditions of the fire environment, monitoring of the manner of fire detection, spread and extinguishing. A comprehensive experiment was conducted in which all combinations of single and double fires that could occur at any location and under any initial conditions were simulated. Thus, a total of 650,880 different fires were simulated and considered. In the simulation, each of the fires was successfully detected and extinguished. Real ship onboard systems are designed and validated by methods which must comply with the rules issued by classification societies, in keeping with designer experience which, though confirmed by calculations, is difficult to validate. This scientific research is a theoretical contribution to the development of autonomous executive systems or advisory systems the application of which affects the safety of maritime transport.

## REFERENCES

- Akyuz, E., 2017. A marine accident analysing model to evaluate potential operational causes in cargo ships. *Safety Science*, 92, pp. 17-25. Available at: <https://doi.org/10.1016/j.ssci.2016.09.010>.
- Bielić, T., Vidan, P., & Mohović, R., 2010. Complacency – Major Cause of Maritime Casualties. *Pomorstvo: Scientific Journal of Maritime Research*, 24(2), pp. 247-260. Available at: <https://hrcak.srce.hr/file/94219>.
- Cavezzali, A., Girotti, A., & Rabino, G., 2003. Multi-Agents Systems and Territory: Concepts, Methods and Applications. 43rd Congress of the European Regional Science Association: "Peripheries, Centres, and Spatial Development in the New Europe". Available at: <http://hdl.handle.net/10419/116178>.
- Cooper, J. & James, A., 2009. Challenges for Database Management in the Internet of Things. *IETE Technical Review*, 26(5), p.320. Available at: <http://dx.doi.org/10.4103/0256-4602.55275>.
- Ferber, J., 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Harlow: Addison-Wesley.
- Fortino, G. et al., 2017. Modeling and simulating Internet-of-Things systems: A hybrid agent-oriented approach. *Computing in Science & Engineering*, 19(5), pp. 68-76. Available at: <https://doi.org/10.1109/MCSE.2017.3421541>.
- Franklin, S. & Graesser, A., 1997. Is It an agent, or just a program?: A taxonomy for autonomous agents. *Lecture Notes in Computer Science*, pp.21-35. Available at: <http://dx.doi.org/10.1007/bfb0013570>.
- Jennings, N.R. & Wooldridge, M., 1998. *Applications of Intelligent Agents*. Agent

Technology, pp.3–28. Available at:

[http://dx.doi.org/10.1007/978-3-662-03678-5\\_1](http://dx.doi.org/10.1007/978-3-662-03678-5_1).

Panait, L. & Luke, S., 2005. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3), pp.387–434. Available at: <http://dx.doi.org/10.1007/s10458-005-2631-2>.

Ramos, M.A. et al., 2018. Accounting for human failure in autonomous ship operations. *Safety and Reliability – Safe Societies in a Changing World*, pp.355–363. Available at: <http://dx.doi.org/10.1201/9781351174664-45>.

Rokseth, B., Utne, I.B. & Vinnem, J.E., 2016. A systems approach to risk analysis of maritime operations. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 231(1), pp.53–68. Available at: <http://dx.doi.org/10.1177/1748006x16682606>.

Russell, S. J., & Norvig, P., 2020. *Artificial intelligence : a modern approach* (Fourth edition. ed.). Hoboken: Pearson. ISBN: 9780134610993.

Šalamon, T., 2011. *Design of agent-based models: Developing computer simulations for a better understanding of social processes*. Repin, Czech Republic: Tomáš Bruckner.

Steels, L., 1995. Building agents out of autonomous behavior systems. In L. Steels, & Brooks, R. (Ed.), *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, 1st ed., pp. 83-121. Routledge.

Sumić, D., 2021. *Agent based decision-making model of on-board distress situation* (in Croatian). (PhD Doctoral Dissertation). University of Zagreb, Zagreb. Available at: [https://www.bib.irb.hr/1111180/download/1111180.DR\\_Dean\\_Sumic.pdf](https://www.bib.irb.hr/1111180/download/1111180.DR_Dean_Sumic.pdf).

Wikman, J. et al., 2017. Study investigating cost effective measures for reducing the risk from fires on ro-ro passenger ships (FIRESAFE). Available at: <http://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-29418>.

Wooldridge, M. & Jennings, N.R., 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), pp.115–152. Available at: <http://dx.doi.org/10.1017/s0269888900008122>.

Wu, B. et al., 2019. Fuzzy logic based approach for ship-bridge collision alert system. *Ocean Engineering*, 187, p.106152. Available at: <http://dx.doi.org/10.1016/j.oceaneng.2019.106152>.

Yu, H., Shen, Z. & Leung, C., 2013. From Internet of Things to Internet of Agents. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. Available at: <http://dx.doi.org/10.1109/greencom-ithings-cpscom.2013.179>.

Zhang, Y., Zhan, Y. & Tan, Q., 2009. Studies on Human Factors in Marine Engine Accident. 2009 Second International Symposium on Knowledge Acquisition and Modeling. Available at: <http://dx.doi.org/10.1109/kam.2009.246>.