COLREGs-based Situation Awareness for Marine Vessels - a Discrete Event Systems Approach

Hansen, Peter Nicholas; Papageorgiou, Dimitrios; Blanke, Mogens; Galeazzi, Roberto; Lûtzen, M.; Mogensen, J. ; Bennedsen, M. ; Hansen, D.

Published in:
IFAC-PapersOnLine

Link to article, DOI:
10.1016/j.ifacol.2020.12.1453

Publication date:
2021

Document Version
Peer reviewed version

COLREGs-based Situation Awareness for Marine Vessels - a Discrete Event Systems Approach ⋆

P. N. Hansen * **D. Papageorgiou** * **M. Blanke** * **R. Galeazzi** * **M. Lützen** ** **J. Mogensen** ** **M. Bennedsen** ** **D. Hansen** **

* *Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, 2800 Lyngby, Denmark, (e-mail: pnha,dimpa,mb,rg@elektro.dtu.dk).*
** *Svendborg International Maritime Academy, 5700 Svendborg, Denmark, (e-mail: mal,jmo,mrb,doh@simac.dk).*

Abstract: Autonomy at different levels is envisaged to provide decision support, to enable navigation with temporally unattended bridge or have the navigator placed remotely, being able to take command if required. For this purpose, methods for computer-based situation awareness are needed to avoid risks of collision. Correct interpretation of a situation is crucial, and all navigation decisions must be based on the COLREGs. This paper presents a discrete-event-systems-based framework that facilitates autonomous interpretation of the situation in which the own ship is. This can be used for COLREGs-compliant decision planning when all vessels navigate according to the rules. The proposed framework comprises a set of coupled finite-state deterministic automata and segregates situation understanding from anticipation. The suggested formalism is analysed with respect to avoidance of deadlocks and shows how synchronisation of vessel-specific automata modules is achieved. Simulations illustrate the concept using realistic scenarios.

Keywords: Autonomous vessels, situation assessment, autonomous situation awareness, discrete-event systems, finite-state automata, COLREGs representation, anticipation.

1. INTRODUCTION

In order to maintain safe navigation of marine vessels, it is vital that a correct situation assessment and interpretation is made. In case that another vessel is likely to come closer to own ship than a limit set for closest point of approach, the situation is said to poses a risk of collision and the International Regulations for Preventing Collisions at Sea (COLREGs) (IMO, 1972) rules apply.

Navigators rely primarily on Radar and Electronic Chart Display and Information System (ECDIS), supplemented by visual information. Their training and experience enables accurate assessment of the current situation, but experience shows that other duties on the bridge or navigation in heavily trafficked areas, have caused incidents. With the advent of highly automated vessels, where electro optical sensors are available and automated object detection and classification is done by computer algorithms, decision support could be made available to assess and suggest handling of immediate and future risks. The process of assessment includes to order all vessels that could pose a risk on an awareness list, which is sorted according to decreasing risk. Using this list, the navigator takes course of action in compliance to the COLREGs.

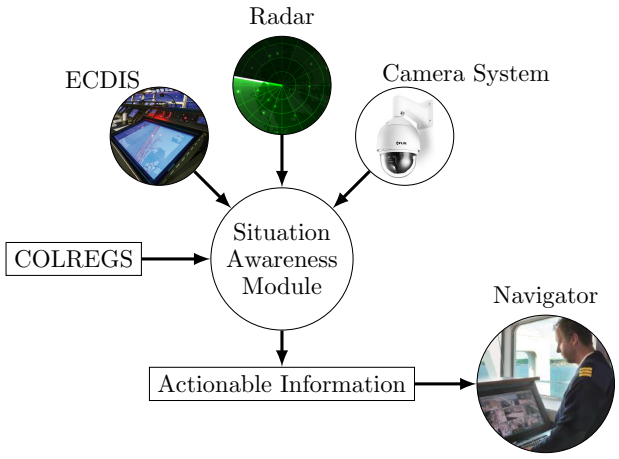


Fig. 1. Autonomous situation awareness module

With the aim of achieving a possibility of temporarily unattended bridge, the navigational process needs be implemented in software, such that a computerized system can oversee the navigation. For such a system to be useful it would need to guarantee the same reliability as an experienced navigator, while also being COLREGs-compliant. Figure 1 illustrates how such a system could be realised; a vision system comprising an array of cameras

⋆ This research is sponsored by the The Danish Maritime Fund, Orients Fund and the Lauritzen Foundation through the ASAN project: Autonomous Situation Awareness for Navigation, and by the Autonomy part of the ShippingLab project, which is sponsored by the Danish Innovation Fund and the aforementioned Foundations.

performs the visual look-out and combines the information with radar readings and ECDIS data. This "perception" system is interfaced to an autonomous *situation awareness* module, which in turn, generates *actionable information*, i.e. information based on rules that can be acted upon. Automated aggregation of the perceived information and its processing into actionable information, would leave the navigator free to focus on higher level tasks, i.e. decision making, based on computer assisted situation awareness.

Situation awareness enhancement for safe navigation at sea, in the air or on ground has been addressed in several works in the past. A method for determining and visualizing information necessary for ship collision avoidance, taking into account weather conditions, was presented in (Szlapczynski and Krata, 2018). The Closest Point of Approach (CPA) between own ship and target vessels, as well as Distance to CPA (DCPA), were used as criteria for risk calculation in a simulator presented in (Hasegawa et al., 2012). The severity of the situation was assessed by examining only the vessel posing the largest collision risk. Threat assessment and integration of the COLREGs based on heading and speed estimation to general collision avoidance strategies was discussed in (Campbell et al., 2012). A multi-objective particle swarm optimization method and its hierarchical variation were presented in (Hu et al., 2017, 2019) for facilitating collision avoidance. The relevant COLREGs were interpreted as mathematical inequalities in the context of the optimisation problem. A review for detection and resolution of eminent trajectory conflicts of aircraft was provided in (Kuchar and Yang, 2000), withy focus on state estimation and propagation for assessing the current situation. Situation awareness and COLREGs-based interpretation was embedded in the decision making process in many of the existing studies. The authors in (Johansen et al., 2016) proposed a Model Predictive Control (MPC) approach for collision avoidance for marine surface vehicles, where the COLREGs were accounted for in the form of constraints. Neural networks were used in (Puente et al., 1992) to define a supervisory logic that combines perceived information from the surroundings of a vessel to fuse basic behaviours that correspond to the COLREGs-described situations and produce collision avoidance actions. Discrete-Event Systems (DES) frameworks for decision making support have also been in focus in relation to collision avoidance for ground and aerial vehicles. Supervisory control was used in (Dallal et al., 2013) to ensure collision avoidance for controlled and uncontrolled vehicles. A similar direction but based on labelled Petri nets was taken in (Wan et al., 2018). Formation control of surface marine vehicles based on DES-logic with embedded collision avoidance features was pursued in (Moreno-Salinas et al., 2018). Parallel automata were found effective for supervision in real-time control in (Blanke et al., 1997), which also suggested implementation using meta-layer principles and reflection (Lunau, 1997) and pointed to the widely recognized fact that correct software implementation is of particular importance when implementing supervision systems. Software architectures were focused on as being essential for supervision and control of robotic systems in (Coste-Maniere and Simmons, 2000) and meta-level methods were used to model control and supervision in (Perronne et al., 2006). Methods for implementation in safety critical applications were reported in (Cuer et al., 2018) for autonomous vehicles and (Rawlings et al., 2020) presented symbolic model checking of supervisory control of labelled transition systems, which presented an essential step toward safe and reliable implementation.

Although many of the previous studies present solutions that facilitate collision avoidance, their approaches rely on directly translating sensory information into manoeuvring actions, i.e. they do not explicitly address the understanding of current situation based on interpretation of perceived information. A multi-layered framework for autonomous situation awareness for navigation at sea based on the DES theory was presented in (Papageorgiou et al., 2019). This approach was able to interpret incoming sensor information and provide an understanding of single-ship encounter situations based on COLREGs. This paper builds on these earlier results and presents an extension that facilitates anticipation of the future evolution of the given situation and ability to provide actionable information to the navigator. The architecture argues for separation of situation awareness into three modules: (*perception*, *understanding* and *anticipation*), with a separate module to coordinate data flow. A parallel architecture allows for tracking and handling of a large number of objects and vessels, and is only limited by the computational resources of the hardware platform. Moreover, the modular design of the framework allows for flexibility for future expansion of the situation awareness component.

The remainder of the paper is structured as follows: Section 2 gives an overview of an autonomous sea-faring system for marine surface vessels and describes a typical situation-assessment/decision-making cycle. Section 3 details the DES-based framework for autonomous situation awareness. The applicability of the proposed approach is demonstrated in a simulation of a real-case scenario in Section 4, where comments are made on the results. Finally, conclusions are drawn in Section 5 and elements of future work are presented.

2. NAVIGATION PROCESS DESCRIPTION

The autonomous navigational process can be divided into two parts. The first part is the situation awareness (marked by the blue box in Figure 2), which relates to the gathering of information about the current state of the own vessel, environment, nearby objects, vessel, etc., and processing this into "actionable information". The second part is the decision making (marked by the orange box in Figure 2), which relates to the decision made based on the actionable information.

Situation awareness comprises three elements: *perception*, *understanding* and *anticipation* (Endsley, 1995) as illustrated in Figure 2. Perception is the collection of sensory information, which for the navigation process consists of: radar data, ECDIS data, weather information, visual lookout and own ship manoeuvrability (propulsion system state). This data is fused together to form a coherent and consistent image of the current situation. Understanding is the process of interpreting the fused information provided by the perception module, and based on this assess the situation at a given instance in time. Anticipation is the expected development of the current situation generated by the understanding module and the environment in

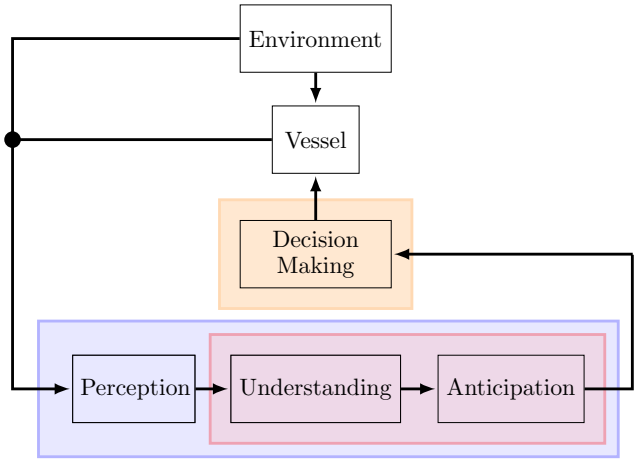


Fig. 2. Block diagram illustrating the placement of the situation awareness (blue) and decision making (orange) modules w.r.t. navigational process. The red box shows the two parts of the situation awareness module, on which this study focuses.

which the system is. For a marine vessel the anticipation will be based on the rules at sea and the environment in which the situation evolves, e.g. applying COLREGs can determine the expected future behaviour of the different vessels in a given situation.

The focus of this paper is the implementation and architecture of the last two steps in the situation awareness module (bound by the red box in Figure 2). The following two assumptions are adopted for the rest of the analysis:

Assumption 1. A perception module that provides detailed information about the detection and classification of objects based on the sensor readings and ship instrumentation is available.

Assumption 2. All vessels obey COLREGs.

3. AUTONOMOUS SITUATION AWARENESS FRAMEWORK

The proposed framework for autonomous situation awareness is based on the tools provided by DES theory and comprises different types of *automata*. The reader is referred to (Cassandras and Lafortune, 2008, ch. 2) for a brief introduction in the basic elements of DES theory relating to framework description. Four different automata types are used, corresponding to the segregated functionalities of the framework:

- *Coordinator automaton*: It manages the entire situation awareness process and coordinates its different submodules.
- *Perception automaton*: It serves as a relay between the perception module (instrumentation and sensor fusion) and the coordinator automaton.
- *Understanding automaton*: It manages the appropriate assessment steps that need be taken for each detected object and interprets the associated perception input.
- *Anticipation automaton*: It correlates the assessments made for detected objects and predicts the evolution of the current situation in the near future based on the COLREGs.

The design and properties of the individual automata are elaborated in the following subsection.

3.1 Coordinator automaton

The coordinator automaton is the main coordinating module of the framework and acts as an intermediate layer that connects the other three types of automata to each other. It receives sequences of events from the other automata and generates appropriate events that trigger transitions into different steps of the navigation process.

The sequences reported to the coordinator may concern the detection of an object or information regarding the type of object, its size, relative bearing etc. These specific type of reports can be sent by either the perception automaton or an understanding automaton, both of them being elaborated later in this section. In the first case the coordinator checks whether the detected object is already being tracked and if so, it updates its properties (type, size, associated collision risk etc.). This information comes from own ship's sensory equipment and the perception module and is stored in a database, which can be accessed by the coordinator. If the object is detected for the first time, then the coordinator generates an instance of the understanding and the anticipation automaton, dedicated to evaluate the newly detected object and project the action of the own ship, respectively. In case the incoming report is sent by a understanding automaton, the coordinator updates the awareness list if necessary and generates an appropriate event that will trigger a transition in the corresponding anticipation automaton.

The coordinator automaton is defined as the five-tuple

$$G_c \triangleq (\mathcal{C}, E_c, f_c, C_1, C_1) \ . \tag{1}$$

where the states $C_i \in \mathcal{C}$ are listed below:

C_1: Wait for new reports (initial and marked state).
C_2: Check type of new report.
C_3: Check if detected object already being tracked.
C_4: Check if object type has changed.
C_5: Update object type.
C_6: Generate new pair of understanding and anticipation automata instances associated with the unique ID of the tracked object, and appends the object to the awareness list.
C_7: Calculate and compare associated risk to highest recorded risk. Update awareness list.
C_8: Assess the type of own ship.
C_9: Generate triggering event for anticipation automaton.

The event set E_c, i.e. the set of events that may trigger transitions between two modes of the coordinator is detailed in Appendix A. The state transition diagram of the coordinator automaton is shown in Figure 3 and it can completely define the extended state transition function $f_c : \mathcal{C} \times E_c^* \to \mathcal{C}$. The languages generated and marked by G_c are defined as

$$\mathcal{L}(G_c) \triangleq \{s \in E_c^* \ : \ f_c(C_1, s) \text{ is defined}\}$$
$$\mathcal{L}_m(G_c) \triangleq \{s \in E_c^* \ : \ f_c(C_1, s) = C_1\}$$

where s is a string of events in the event set of the automaton.

The coordinator automaton G_c is non-blocking since from each mode C_i there exists a path in the transition diagram

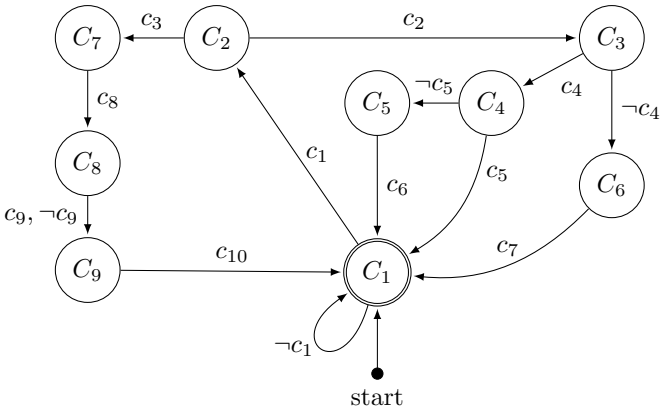


Fig. 3. State transition diagram of the coordinator automaton.

to C_1, i.e all its states are *coaccessible* to C_1. This can be easily confirmed by inspecting the reachability matrix of G_c. It should be noted that state C_8, although trivial, is needed for future extensions of the framework, where non power-driven vessels will also be considered as own ship type and therefore different COLREGs have to be considered.

3.2 Perception automaton

The perception automaton handles the information regarding detection and classification of objects and their properties and conveys it to the coordinator. This information is obtained from a database of detected objects, which is populated by the perception module (sensor fusion, detection and classification systems). The perception automaton is defined by the five-tuple

$$G_p \triangleq (\mathcal{P}, E_p, f_p, P_1, P_1) \ . \tag{2}$$

where the states $P_i \in \mathcal{P}$ are listed below

P_1: Scan for objects (initial and marked state).
P_2: Report object detection to the coordinator.

Given the possible outcomes listed below, i.e. the different cases of the process described by G_p,

p_1 : Object is detected.
p_2 : Object detection is reported to the coordinator.

the event set of G_p is defined by $E_p \triangleq \{p_1, \neg p_1, p_2\}$. The transition diagram of G_p is illustrated in Figure 4. The languages generated and marked by G_p are defined as

$$\mathcal{L}(G_p) \triangleq \left\{ s \in E_p^* \ : \ f_p(P_1, s) \text{ is defined} \right\}$$
$$\mathcal{L}_m(G_p) \triangleq \left\{ s \in \mathcal{L}(G_p) \ : \ f_p(P_1, s) = P_1 \right\} \ .$$

It is easy to see from its state transition diagram that G_p is coaccessible and, therefore, non-blocking.

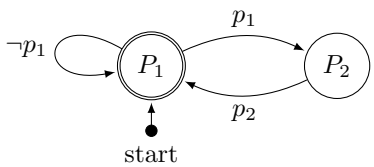


Fig. 4. State transition diagram of the perception automaton.

3.3 Understanding automaton

Several instances of the understanding automaton can be generated by the coordinator, each instance assessing the situation with respect to one specific detected object. If the object violates the predefined limits for CPA and Time for Closest Point of Approach (TCPA) d_{req}, t_{req}, respectively, the object is close enough to potentially pose a collision risk, and then more detailed assessment has to be made. If the object is land or a buoy, then its shape, type and position need to be validated, whereby own vessel's position and course can be validated. On the other hand, if the object is a vessel, then its type (sail boat or power driven vessel) and bearing relatively to the own ship (port, starboard, etc.) has to be checked along with the CPA and TCPA, so as to evaluate whether the own ship has to stand on or give way according to the COLREGs. If the CPA is smaller than a safety distance limit d_{act} and the TCPA is smaller than a predefined time action limit t_{act}, then an action has to be taken. The discrete states $U_i \ : \ \bigcup_{i=1}^{13} \{U_i\} \triangleq \mathcal{U}$ concerning the assessment of detected objects are listed below:

U_1: Compare CPA to d_{req} (initial and marked state).
U_2: Compare TCPA to t_{req}.
U_3: Evaluate object type.
U_4: Calculate own ship bearing relatively to target vessel.
U_5: Assess target vessel type.
U_6: Compare TCPA to Time to Next Waypoint (TTW).
U_7: Check if next waypoint is towards target vessel.
U_8: Compare CPA to distance action limits d_{act}.
U_9: Compare TCPA to time action limits t_{act}.
U_{10}: Validate geometry of detected object using ECDIS.
U_{11}: Evaluate own ship position uncertainty.
U_{12}: Compare next waypoint to detected object's position.
U_{13}: Report events sequence to coordinator (marked state).

The event set E_u associated to the discrete states set $\mathcal{U}$ is presented in detail in Appendix A. The extended transition function $f_u : \mathcal{U} \times E_u^* \to \mathcal{U}$ is fully described by the state transition diagram in Figure 5. Based on the foregoing description, the understanding automaton G_d is defined as the five-tuple

$$G_u \triangleq (\mathcal{U}, E_u, f_u, U_1, \{U_1, U_{13}\}) \ . \tag{3}$$

The set of marked states for G_u is chosen to contain only the states U_1, U_{13} because they signify the end of a full assessment cycle and the reporting to the coordinator, respectively. The languages generated and marked by G_u are defined as

$$\mathcal{L}(G_u) \triangleq \{ s \in E_u^* \ : \ f_u(U_1, s) \text{ is defined} \}$$
$$\mathcal{L}_m(G_u) \triangleq \{ s \in \mathcal{L}(G_u) \ : \ f_u(U_1, s) \in \{U_1, U_{13}\} \} \ .$$

Similarly to the coordinator automaton, G_u is also non-blocking since from any mode U_i there exists a path in the transition diagram to U_1 or U_{13}.

Remark 1. The only possible events that can trigger the transition from U_4 to U_5 are; u_4, u_5, u_6, u_7, which are mutually exclusive. Moreover, it is guaranteed that one, and only one, of the events will occur, e.g. $u_4 \Rightarrow \neg u_5 \wedge \neg u_6 \wedge \neg u_7$.

Remark 2. Although the transition from state U_5 to U_6 does not depend on whether the target vessel is power

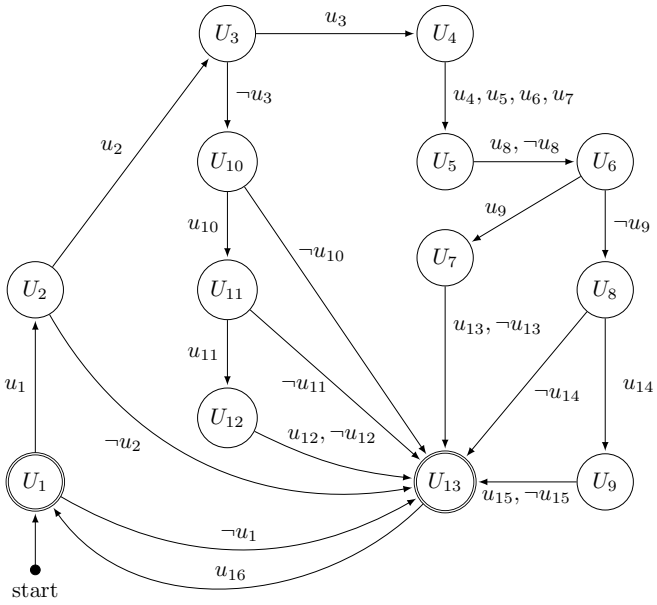


Fig. 5. State transition diagram of the understanding automaton.

driven or not, different COLREGs rules apply for non-power driven vessels, hence the need for the distinction between the two types.

3.4 Anticipation automaton

The anticipation automaton G_a is defined as the five-tuple

$$G_a \triangleq (\mathcal{A}, E_a, f_a, A_1, A_1). \tag{4}$$

with the discrete states $A_i : \bigcup_{i=1}^{2} \{A_i\} \triangleq \mathcal{A}$ defined as:

A_1: Vessel expected to stand on (initial & marked state).
A_2: Vessel expected to give way.

The event set E_a governing the transitions between the modes A_i are defined as $E_a \triangleq \bigcup_{i=1}^{31} \{a_i\}$, where the events a_i are generated by the coordinator by concatenating the words received by the corresponding understanding automaton and those of the coordinator itself. For instance, the events a_1, a_5, a_7, a_8 are defined as

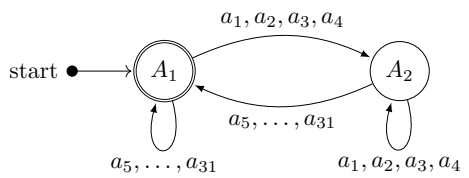


Fig. 6. State transition diagram of the anticipation automaton.

$$a_1 = \overbrace{u_1 u_2 u_3 u_7 u_8 \neg u_9 u_{14} u_{15}}^{\text{Report from Understanding automata}} \underbrace{c_1 c_3 c_8 c_9}_{\text{Appended word from Coordinator}} \rightarrow \text{“Give Way”}$$

$$a_5 = u_1 u_2 u_3 \overbrace{u_4}^{\text{Object is overtaking}} \underbrace{u_8}_{\text{Target is power-driven}} \neg u_9 u_{14} u_{15} c_1 \underbrace{c_3}_{\text{Report sent from an understanding automaton}} c_8 \overbrace{c_9}^{\text{Own ship is power-driven}} \rightarrow \text{“Stand On”}$$

$$a_7 = u_1 u_2 u_3 \overbrace{u_6}^{\text{Object is crossing from port}} \underbrace{\neg u_8}_{\text{Target is not power-driven}} \neg u_9 u_{14} u_{15} c_1 c_3 c_8 \overbrace{c_9}^{\text{Own ship is power-driven}} \rightarrow \text{“Give way”}$$

$$a_8 = u_1 u_2 u_3 \overbrace{u_7}^{\text{Object is crossing from starboard}} \underbrace{\neg u_8}_{\text{Target is not power-driven}} \neg u_9 u_{14} u_{15} c_1 c_3 c_8 \overbrace{c_9}^{\text{Own ship is power-driven}} \rightarrow \text{“Give way”}$$

Similarly to the previous automata, the extended transition function $f_a : \mathcal{A} \times E_a^* \rightarrow \mathcal{A}$ is fully described by the state transition diagram in Figure 6. The languages generated and marked by the anticipation automaton G_a are defined as

$$\mathcal{L}(G_a) \triangleq \{s \in E_a^* \ : \ f_a(A_1, s) \text{ is defined}\}$$
$$\mathcal{L}_m(G_a) \triangleq \{s \in \mathcal{L}(G_a) \ : \ f_a(A_1, s) = A_1\} \ .$$

The anticipation automaton G_a is non-blocking since both its states are coaccessible to A_1

Remark 3. An instance of the anticipation automaton exists for each other vessel in the area of awareness. The automaton describes the rules that apply to own ship and the other vessel. The required behaviour of either vessel will be “stand on” or “give way” if there is a risk of collision. The set of anticipations will describe the expected evolution of the situation, taking both TCPAs and required behaviour into consideration.

The entire situation awareness framework can be represented by the parallel composition of the aforementioned automata. Since each of them is non-blocking and they do not have any common event, it follows that their parallel composition is also non-blocking. The architecture of the proposed framework is illustrated in Figure 7.

4. SIMULATION RESULTS

The proposed framework was implemented and tested in simulation environment using Python.

4.1 Simulation scenarios

The framework is tested against a simple scenario with three vessels. The purpose of the simulation is not to investigate the resolutions of possibly complex situations but to demonstrate the effectiveness of the framework in evaluating an encounter of more than two vessels and project a possible evolution of the situation. Given are the initial positions and speeds of own ship and target vessels and based on this information, the framework provides an assessment of the situation in the form of anticipated behaviour tables. The following assumptions are made:

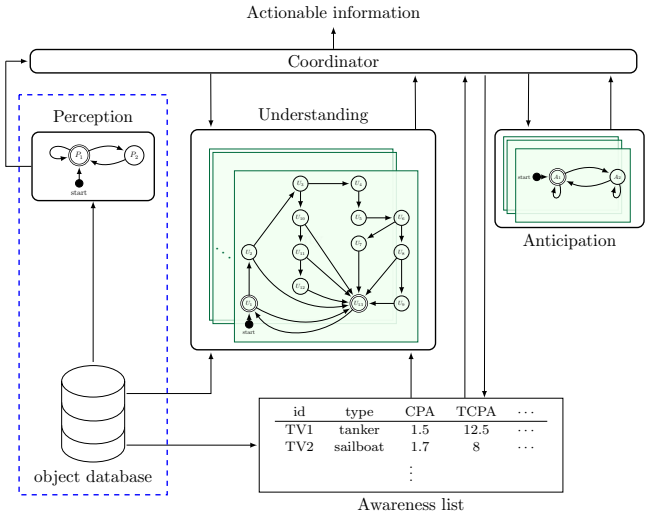
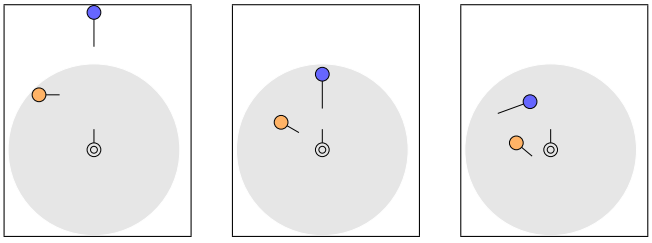


Fig. 7. Block diagram illustrating the architecture of the DES framework for situation awareness.

Assumption 3. All vessels are within the detection range of the sensory system on-board Own Ship

Assumption 4. All vessels are power-driven which is detectable by the perception system.

Figure 8 illustrates the given scenario. The Own Ship (OS) is denoted by two homocentric circles, while the target vessels by the coloured circles. The gray disk around OS denotes the awareness zone (where $TCPA \leq t_{req}$).



(a) $t = t_0$ (b) $t = t_1$ (c) $t = t_2$

Fig. 8. Simulation scenario: Port crossing (orange Target Vessel 1 (TV1)) and head-on (blue Target Vessel 2 (TV2)) in three different time instances.

4.2 Simulation results

- $t = t_0$: TV1 in orange is detected to be crossing from port, therefore OS stands on (rules 2 and 15). TV2 in blue is on head-on course but the t_{req} is not violated since it is outside the awareness zone, so OS stands on.
- $t = t_1$: TV2 enters the awareness zone and OS needs to give way (rule 14). The situation is interpreted as OS (and TV2) being expected to give way. However, the condition $TCPA < t_{act}$ is not yet violated with respect to TV2, hence OS will not make any manoeuvre yet.
- $t = t_2$: TV2 turns starboard due to its predefined route and there is no collision risk with OS (CPA changes). Therefore, OS is expected to stand on.

For each of the scenarios, the different automata in the simulation will produce a set of strings, which are interpreted by the anticipation automata for each of the vessels. The reported set of strings w.r.t. TV1 during simulation r_1, r_2, r_3, are shown in Equations 5, 6 & 7.

$$\text{At } t = t_0 : r_1 = u_1 u_2 u_3 \overbrace{u_6}^{\substack{\text{Object on}\\\text{port side}}} \underbrace{u_8}_{\text{Target is power-driven}} \neg u_9 u_{14} u_{15} c_1 \underbrace{c_3}_{\substack{\text{Report from a}\\\text{Understanding automata}}} c_8 \overbrace{c_9}^{\substack{\text{Own ship is}\\\text{power-driven}}} \tag{5}$$

$$\text{At } t = t_1 : r_2 = u_1 u_2 u_3 u_6 u_8 \neg u_9 u_{14} u_{15} c_1 c_3 c_8 c_9 \tag{6}$$

$$\text{At } t = t_2 : r_3 = \underbrace{\neg u_1}_{\text{CPA}>d_{req}} c_1 c_3 c_8 c_9 \tag{7}$$

The reported set of strings w.r.t. TV2 during simulation r_4, r_5, r_6, are shown in Equations 8, 9 & 10.

$$\text{At } t = t_0 : r_4 = u_1 \overbrace{\neg u_2}^{\text{TCPA}>t_{req}} c_1 c_3 c_8 c_9 \tag{8}$$

$$\text{At } t = t_1 : r_5 = u_1 \overbrace{u_2}^{\text{TCPA}<t_{req}} u_3 \underbrace{u_5}_{\text{Object is head-on}} u_8 \neg u_9 u_{14} u_{15} c_1 c_3 c_8 c_9 \tag{9}$$

$$\text{At } t = t_2 : r_6 = \neg u_1 c_1 c_3 c_8 c_9 \tag{10}$$

The situation and anticipated action of OS at each time instance of the simulation, are shown in Table 1.

Table 1. Scenario action and situation table

Time	$t = t_0$		$t = t_1$		$t = t_2$	
-	TV1	TV2	TV1	TV2	TV1	TV2
Situation	Port cross.	-	Port cross.	Head-on	-	-
Action	Stand On	Stand On	Stand On	Give Way	Stand On	Stand On

5. CONCLUSIONS AND FUTURE WORK

The paper proposed a framework for situation awareness using a distinct separation between understanding, anticipation and decision making. The framework consists of multiple automata connected through a coordinator automaton, which is responsible for communication between instances of underlying automata. The coordinator can generate new instances of automata, allowing the framework to handle multi-object scenarios. The usability of the framework was demonstrated by simulating scenarios with high risk of collision, where the anticipated COLREGs-compliant action of own ship changed over time. The paper demonstrated how autonomous situation awareness was achieved through appropriate monitoring of events and state representation in the anticipation automata.

Future extensions of the work will include scenarios with higher complexity, multi-vessel autonomous decision making and path planning. Validation of the reliability of the proposed assessment framework will be conducted for situations used in Master Mariner exams.

Appendix A. DESCRIPTION OF EVENT SETS

The coordinator event set is defined as $E_c \triangleq \{c_i, \neg c_i\}$ where

c_1 : A new report is available.
c_2 : Report been sent by perception automaton.
c_3 : Report sent by understanding automaton.
c_4 : Detected object currently tracked.
c_5 : Different object type from that on awareness list.

c_6 : Objected type updated on awareness list.
c_7 : Relevant understanding automaton instances generated, object appended to awareness list.
c_8 : Object poses the highest collision risk.
c_9 : The own ship is a power-driven vessel.
c_{10} : Event associated to perception and understanding reports was sent to anticipation automaton.

The understanding automaton event set is defined as $E_u \triangleq \bigcup_{i=0}^{16} \{u_i, \neg u_i\}$ where

u_1 : Object violates CPA awareness level (CPA$< d_{req}$).
u_2 : Object violates TCPA awareness level (TCPA$< t_{req}$).
u_3 : Detected object is a vessel.
u_4 : Detected object overtaking.
u_5 : Detected object is head-on towards own ship.
u_6 : Detected object is on port side.
u_7 : Detected object is on starboard side.
u_8 : Detected object is power-driven.
u_9 : TCPA larger than TTW.
u_{10} : Object geometry matches ECDIS description.
u_{11} : Own ship position certainty factor above threshold.
u_{12} : Conflict between route and land/buoy position.
u_{13} : Next waypoint on the same side as object.
u_{14} : CPA smaller than distance action limits d_{act}.
u_{15} : TCPA smaller than time action limits t_{act}.
u_{16} : Report sent to coordinator.

REFERENCES

Blanke, M., Izadi-Zamanabadi, R., Bøgh, S.A., and Lunau, C.P. (1997). Fault tolerant control - a holistic view. *Control Engineering Practice*, 5(5), 693–702. doi:10.1016/S0967-0661(97)00051-8.

Campbell, S., Naeem, W., and Irwin, G.W. (2012). A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2), 267–283.

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to discrete event systems*. Springer US.

Coste-Maniere, E. and Simmons, R. (2000). Architecture, the backbone of robotic systems. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. IEEE. doi:10.1109/robot.2000.844041.

Cuer, R., Piétrac, L., Niel, E., Diallo, S., Minoiu-Enache, N., and Dang-Van-Nhan, C. (2018). A formal framework for the safe design of the autonomous driving supervision. *Reliability Engineering & System Safety*, 174, 29–40. doi:10.1016/j.ress.2018.01.014.

Dallal, E., Colombo, A., Del Vecchio, D., and Lafortune, S. (2013). Supervisory control for collision avoidance in vehicular networks using discrete event abstractions. *Proceedings of the American Control Conference*, 6580514, 4380–4386.

Endsley, M.R. (1995). Toward a theory of situation awareness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37, 32–64.

Hasegawa, K., Fukuto, J., Miyake, R., and Yamazaki, M. (2012). An intelligent ship handling simulator with automatic collision avoidance function of target ships. *Proc. of INSLC*, 17.

Hu, L., Naeem, W., Rajabally, E., Watson, G., Mills, T., Bhuiyan, Z., Raeburn, C., Salter, I., and Pekcan, C. (2019). A multiobjective optimization approach for colregs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators. *IEEE Transactions on Intelligent Transportation Systems*.

Hu, L., Naeem, W., Rajabally, E., Watson, G., Mills, T., Bhuiyan, Z., and Salter, I. (2017). Colregs-compliant path planning for autonomous surface vehicles: A multiobjective optimization approach. *IFAC-PapersOnLine*, 50(1), 13662–13667.

IMO (1972). *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*. `http://http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx` [Accessed: September 19].

Johansen, T.A., Perez, T., and Cristofaro, A. (2016). Ship collission avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12).

Kuchar, J.K. and Yang, L.C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on intelligent transportation systems*, 1(4), 179–189.

Lunau, C.P. (1997). A reflective architecture for process control applications. In *Proc. ECOOP'97 - 11th European Conference on Object-Oriented Programming, 11th European Conference, Jyväskylä, Finland, June 9-13,*, volume 1241 of *Lecture Notes in Computer Science*, 170–189. Springer. doi:10.1007/BFb0053379.

Moreno-Salinas, D., Crasta, N., Pascoal, A.M., and Aranda, J. (2018). Formation control of surface marine vehicles for underwater target tracking using range information. *2018 13th Apca International Conference on Control and Soft Computing (controlo). Proceedings*, 201–6, 201–206.

Papageorgiou, D., Blanke, M., Lützen, M., Bennedsen, M., Mogensen, J., and Hansen, D. (2019). Parallel automaton representation of marine crafts' colregs based manoeuvering behaviours. *IFAC-PapersOnLine*, 52(21), 103–110. doi:10.1016/j.ifacol.2019.12.291.

Perronne, J.M., Thiry, L., and Thirion, B. (2006). Architectural concepts and design patterns for behavior modeling and integration. *Mathematics and Computers in Simulation*, 5-6(70), 314–329. doi:10.1016/j.matcom.2005.11.004.

Puente, E., Gachet, D., Pimentel, J., Moreno, L., and Salichs, M. (1992). A neural network supervisor for behavioral primitives of autonomous systems. In *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation*, 1105–1109. IEEE.

Rawlings, B.C., Lafortune, S., and Ydstie, B.E. (2020). Supervisory control of labeled transition systems subject to multiple reachability requirements via symbolic model checking. *IEEE Transactions on Control Systems Technology*, 28(2), 644–652. doi:10.1109/tcst.2018.2877621.

Szlapczynski, R. and Krata, P. (2018). Determining and visualizing safe motion parameters of a ship navigating in severe weather conditions. *Ocean Engineering*, 158, 263–274.

Wan, Y., Luo, J., Zhang, Q., Wu, W., Huang, Y., and Zhou, M. (2018). Controller design for avoiding collisions in automated guided vehicle systems via labeled petri nets. *IFAC-PapersOnLine*, 51(7), 139 – 144. 14th IFAC Workshop on Discrete Event Systems WODES 2018.