International Journal of Ocean Law, Maritime Security and Space Law
Volume 1 Issue 1, 2026

Marine MAS: A multi-agent framework to support autonomous maritime system design, modeling, and assessment

Meharine. J

Department of Politics, Philosophy and Religion,

Article Info
Received: 23-12-2025 Revised: 02-01-2026 Accepted: 10-01-2026 Published: 20-01-2026

ABSTRACT
Autonomous shipping brings a great opportunity to improve the maritime traffic safety, reliability, and to reduce the cost, whose prototypes are expected to be industrialized in coming years. Current studies and projects focus on proof development of autonomous vessels, evaluation of autonomous vessel safety, and formulation of guidance for autonomous shipping paradigm. In system level, autonomous shipping is anticipated to realize advanced intelligence and autonomy in terms of planning, situation awareness and shipping safety, decision-making, etc. In this study, being aligned with the aforementioned aspects, a multi-agent system (MAS) framework, called "MarineMAS," is proposed to aid the autonomous shipping modelling and evaluation. We emphasize the design perspective that especially focuses on critical knowledge transfer to make transition towards autonomous shipping, the core components to achieve system intelligence, and the potential technological set and experience gained to support efficient MAS modelling. A prototype system has been imple-mented from scratch to demonstrate the key features of the proposed MarineMAS frame-work. Through characterizing the multi-agent definitions and incorporating the corresponding evaluation logics, MarineMAS can be further developed and applied for various applications in autonomous shipping evaluation and management scenarios.

Introduction

In the marine industry, autonomous shipping is a new paradigm for maritime transportation that will challenge the conventional methods used in the current system. This calls for innovative ways to the design, planning, testing, and assessment of unmanned ships and their systems (NFAS 2017). Unmanned surface vehicles (USVs), also known as autonomous surface vehicles (ASVs) or autonomous surface crafts (ASCs), have long been proposed and demonstrated by a number of research labs and companies prior to the recent introduction of autonomous merchant vessels (Liu et al. 2016; Manley 2008). Autonomous shipping is anticipated to provide significant advantages such as increased flexibility and eco-friendliness, improved operational dependability and accuracy, reduced development and operating costs, and improved crew and traffic safety. Currently, some theoretical studies (Burmeister et al. 2014b; Wölbel, Montewka, and Kujala 2017; Man et al. 2015) have concluded that the development of autonomous ships can reduce navigational accidents by eliminating operational errors caused by human factors like fatigue or overtired operation. Numerous R&D initiatives are being launched worldwide to provide proof of the autonomous shipping idea. A proposal for an autonomous ship is being developed and verified in Europe via the MUNIN project, which is co-funded by the European Commission under its Seventh Framework Program (Porathe 2013; Burmeister et al. 2014a). The goal of the ENABLE project is to demonstrate, confirm, and certify autonomous vehicle safety in Europe ("NAVTOR Takes Maritime Lead for EU" 2017). Supported by MacGregor, a division of Cargotec, the association of Finnish Marine Industries, and the Finnish financing organization TEKES, "One Sea" seeks to allow entirely remote-controlled boats in the Baltic Sea ("Project Aims for World First" 2017). The goal of the Rolls-Royce-led AAWA project is to establish Finland's first commercial ship operators ("AAWA project" 2017). The Maritime Safety Administration (MSA) of China funds the Unmanned Multifunctional Maritime Ships Research and Development Project, which aims to achieve motorized multipoint coverage, intelligent cruise and rescue, and all-day networked sea supervision ("China Completes Unmanned" 2017). Japan launches a collaborative endeavor to create a technological idea for an autonomous maritime transport system ("MOL Launches" 2017). An experimental 132-foot autonomous battleship was created by the US military ("The US Navy's new autonomous" 2016). The majority of these initiatives include strong cooperation between academic institutions, businesses, and governmental organizations. The autonomous shipping guidelines are being developed by several organizations and forums ("Guidelines for Autonomous Shipping" 2017; LR specifies "autonomy levels" 2016), definitions (NFAS 2017), and pre-analysis (Blanke, Henrique, and Bang 2017), which aims to develop consistent terminology and provide recommendations to the community of autonomous ship researchers. The Norwegian Forum for Autonomous Ships (NFAS) defines autonomous ships into four classes based on their level of autonomy (NFAS 2017). In contrast, Lloyd's Register assigns six autonomy levels (AL) to define a cyber-enabled autonomous ship, ranging from AL 0, which is "fully manned," to AL 5&6, which is "fully autonomous" (LR defines "autonomy levels" 2016). These standards address several important topics, such as data storage, cybersecurity, network and communication, human-machine interface, integrated system evaluation, and system design.

Autonomous shipping, which involves many entities and elements at a convincing size, is definitely complex system engineering as it incorporates new operational logics and models that need to be examined. It is difficult to analytically characterize every component of the system, including route planning and coordination, seaway capacity, and vessel collision risk assessment, among many others, in a variety of

complicated situations using just mathematical models. Multi-agent system (MAS) technologies have been shown to be an efficient solution for complex systems and their optimization. MAS has been used in a number of areas, including smart manufacturing systems (Guo and Zhang 2009), system control (Qiao, Liu, and Guy 2006), and microgrid management (Xiao et al. 2010). MAS has been used in the maritime traffic domain to plan and carry out counter-piracy operations (Jakob et al. 2012), estimate accident probabilities (Xiao et al. 2012), manage maritime human activities (Parrott et al. 2010), prevent vessel collisions (Pouyan and Jafarinejad 2015), and support maritime search and rescue decision-making (Yu and Xue 2010). Additionally, MAS has been used in practical industrial applications for meso- or macro-scale system modeling and assessment, which is needed to assist high-level decision-making and strategy creation (Bergenti and Vargiu 2010; Pěchouček and Maňk 2008).

In terms of investigating a timely transportation schema, autonomous shipping necessitates a software modeling process in addition to the hardware component, such as constructing autonomous shipping for testing experiments. This process is crucial for providing insightful information and direction for testbed trials or proof of system design, and it may eventually be helpful for an actual autonomous shipping system that enters services. This study proposes a MAS framework named "MarineMAS" to help with autonomous shipping system design, planning, modeling, and assessment. In order to satisfy the operating logics and crucial needs of autonomous shipping, we pay particular attention to important components, processing, and design concerns. As a result, a prototype system is put into place that addresses the fundamental procedures for autonomous traffic planning, situation awareness, and safety management. The computational architecture for large-scale simulation and design for effective modeling, including spatial indexing, are recommended based on the original system implementation. The following is a summary of this work's main contents:

For the first time in this field, a multi-agent framework called "MarineMAS" is suggested to support autonomous shipping system modeling and assessment in line with the growing needs of autonomous shipping. Critical thought on the design of the framework and key components of processing has been elucidated. In order to facilitate autonomous shipping system modeling, we define the basic operation procedure as a processing chain comprising realistic origin-destination route planning, dynamic situation awareness, and shipping decision-making (actions) with the concretized solutions. In order to define the fundamental aspects of MarineMAS, a prototype system based on Java technologies is created from scratch. The demonstrated findings are given in an understandable manner to explain the crucial processing components. We have included the essential algorithms for route planning, situational awareness, risk avoidance measures, etc. into the prototype system. However, in this study, we primarily concentrate on framework architecture, key processing, logics, and rationalities to allow autonomous shipping assessment rather than proposing particular algorithms. ● The knowledge gained from the prototype system implementation has been presented, including the logical processing flow, the prospective technology set to accomplish scalable MAS modeling, and spatial indexing (proposed for the first time) to realize spatial query in constant time. In order to overcome comparable difficulties in their MAS implementation, interested readers should consult these understandings. Our focus in this study is to present the MarineMAS framework design and provide an overview of the information acquired during the creation of our prototype. The remainder of the document is structured as follows: The conceptual design of the Marine MAS framework is covered in the "Framework Conceptual Design" section, with a focus on crucial elements of knowledge transfer, origin-destination route planning, situation awareness, and coordination for autonomous shipping. We also provide an overview of the possible modifications on particular algorithms for autonomous shipping adaptation. The technical specifics of the prototype system are provided in the "Prototype System Implementation" section, which also discusses the user interface and elaborates on system implementation. The suggested remedy and possible future research are summarized in the Conclusion Section, which wraps up the work.

Framework conceptual design

Autonomous shipping transportation systems can be modeled with multiple static and dynamic agents, such as vessel agents and shore station agents, along with the definition of their behaviors, in order to investigate maritime traffic activities with advanced working modes and to pre-evaluate specific unexplored scenarios for creating insights on system safety, efficiency, and reliability. By mimicking their real-world counterparts' actual behaviors—such as sailing, planning, danger perception, and risk avoidance acts of an autonomous ship—each kind of these agents is autonomous to achieve their own goals. In order to investigate the system mechanism for autonomous shipping, the interaction between agents (vessel–vessel, vessel–shore station, etc.) attempts to simulate the practical coordination between various agent entities to accomplish smart navigation both individually and collectively. As seen in Figure 1, MarineMAS has a hierarchical layered design that logically arranges the functions of three layers: the "knowl-edge layer," the "MAS layer," and the "analytics layer." This allows for system modeling and makes it easier to evaluate pre-defined test cases and scenarios or create operational guidelines and decision-making.

Once the MAS is established, the knowledge layer enables realistic agent-behavior modeling by providing crucial prior information for agents to employ during runtime. In the system, it is guarantyd that level autonomous shipping would be gradually transferred from the current maritime system rather than completely redesigned, which means that the majority of the established rules and regulations, well-defined sailing seaways, navigation guidelines, Electronic Navigational Charts (ENC) data, and environment GIS information are still applicable for autonomous shipping, among other things. These data are the essential input data for useful MAS modeling, or the supporting information required for navigation. Furthermore, the large-scale traffic AIS and radar data may be used to extract useful facts and pattern information, according to recent research (Pallotta, Vespe, and Bryan 2013). Waterway patterns extracted from AIS data, for example, show how boats sail in line with traffic

regulations, planning, and demand. These patterns may be used as a reference for building actual seaway routes that approximate those practical cases. Additionally, various traffic patterns, vessel behavior, and risk models are described using AIS data (Silveira, Teixeira, and Guedes Soares 2013). To define agents' behavior (route planning, situation awareness, and coordination) and to construct agents' basic knowledge base, MarineMAS uses both standardized navigation data and traffic realities/patterns derived via an offline learning process.

The specification of agents and their relationships, runtime processing logics, modeling methods, and the actions and interactions of the agents in the simulation scenarios are all provided by the MAS modeling layer. Online situation awareness, route planning, and coordinating action for collision avoidance are the key components of an intelligent traffic system. An autonomous vessel must anticipate traffic in the area and evaluate any hazards beforehand, then take appropriate

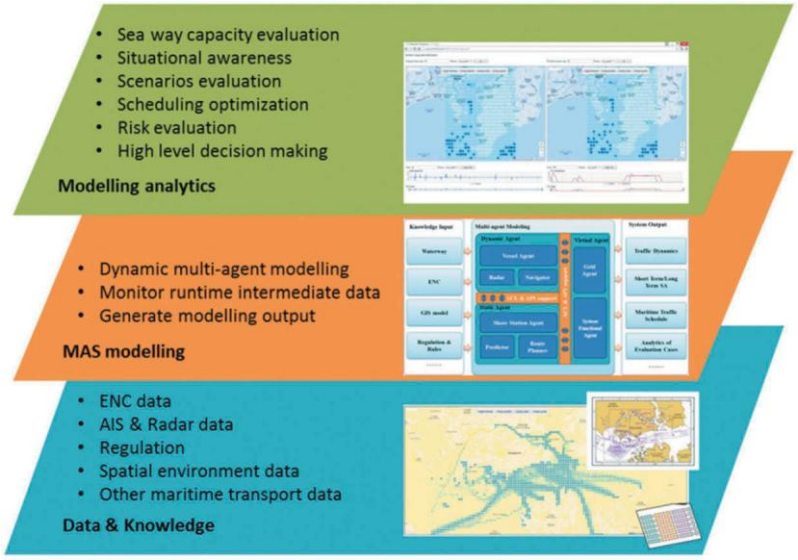


Figure 1. Hierarchy layered design of MarineMAS framework.

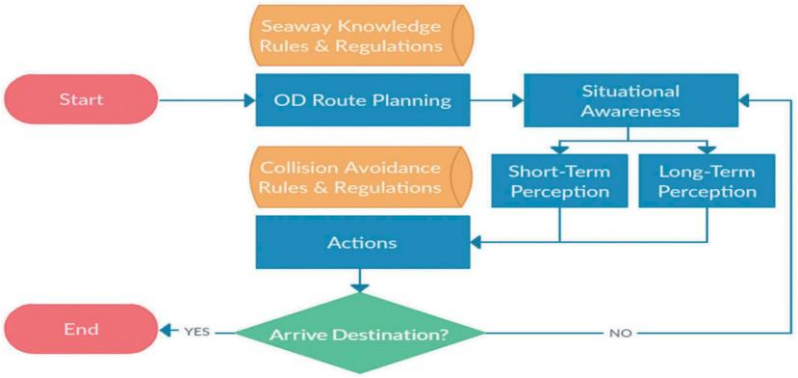


Figure 2. The fundamental operating logic of an autonomous vessel' sailing model.

activities. Figure 2 shows the basic operational logic for an autonomous ship agent, which includes the following sub-steps:

Step 1: Origin-Destination (OD) route planning: During runtime simulation, an autonomous ship agent should follow the reference navigation path that is generated by OD route planning. According to LR's definition, an OD route may be created by the Shore Control Center (SCC) (AL2-4) or allocated directly by the autonomous shipping system (AL5&6);

Step 2: Situation Awareness (SA): SA is an online predictive state perception during navigation that depends on the capacity to anticipate future traffic situations and possible hazards beforehand. SA time win-dow may be projected into the future in either a short-term or long-term manner. It is expected that an autonomous ship would be able to detect nearby traffic and predict how it will change in the near future (short term). An autonomous ship may achieve knowledge of traffic distribution, such as density/hotspot, for hours ahead with assistance from SCC or marine traffic authorities. Step 3: actions based on SA perception, such as steering rearrangement, route modification, or collision avoidance. A repeating processing loop is created by steps two and three until the simulated vessel arrives at its destination. It should be mentioned that this elementary technique is always used for numerous boats in a specified test case or scenario in any sophisticated navigation assessment setups,

such as for safety evaluation or seaway capacity evaluation. It may provide significant analytical findings that are in line with the simulation goal by modifying the modeling parameters used in each stage of this essential procedure.

The analytics layer gathers the MAS modeling output and applies all modeling-related processing and analytics for the intended issue. The modeling analytics in MarineMAS focus on the meso- or macro-system level. Route planning and scheduling optimization, risk mitigation strategy evaluation, seaway capacity assessment under specific safety requirements, situation awareness behavior with respect to varying information access (limited by radar sensing range or rely on the data accessible about the neighboring traffic data, such as the future trajectory of surrounding vessels), and other high-level decision-making and navigation strategies are some examples of modeling use cases for an autonomous shipping system. As long as autonomous ships operate in partial automation mode with remote monitoring (AL2-4), for instance, MAS can facilitate situation/scenario-based evaluation. For instance, in the event that a ship's connection to SCC is lost, the impact of a communication break and the effect of fault tolerance and redundancy strategies can be evaluated in the model for creating the navigation safety guidelines in various scenarios. Since OD route planning, dynamical situation awareness, and the related actions are essential operating components for any complex evaluation case, each of these elements will be explained in the discussion that follows, along with information on the necessary data and knowledge input and the corresponding processing operations.

OD route planning

Even though autonomous shipping uses novel operating modes and sophisticated sensing and control technology, the majority of current transportation regulations and the knowledge gathered from long-standing traffic operations are anticipated to be applied to autonomous shipping. Based on a vessel's past AIS data, we provide various trajectory records in Figure 3, which describe



Figure 3. The AIS data records collected for a vessel.

the routes that the ship often travels. The voyage knowledge (waterway) for the same or comparable vessel may essentially be used as a guide for future trip route planning. When combined with other navigational data, such as ENC and GIS data, the large-scale real-world maritime traffic datasets reflect fine-grained traffic patterns and rules derived from maritime practice, which can be used for practical route planning. Together, these datasets form the knowledge base of an autonomous ship agent.

Sea route patterns for various kinds of boats may be derived from the maritime traffic AIS (or radar) data. Figure 4, for instance, shows the waterway patterns of six different kinds of boats (based on half-year AIS data acquired for Singapore waters). The waterway patterns exhibit clear variability, indicating the variety of maritime journey design for different marine transportation needs. In order to create a more realistic OD path, an autonomous vessel might refer to comparable types of boats (or vessels with similar features) based on practical transportation need. We will describe how to apply the waterway pattern to route planning techniques for OD route generation in the discussion that follows.

An autonomous ship's journey trip begins with route planning from origin to destination, which is often organized by SCC (i.e., route planner). A* (Hart, Nilsson, and Raphael 1968) and Theta* (Daniel et al. 2007), two well-known route finding algorithms that have been used and validated in actual transportation route planning practice, are suitable choices for this work. These algorithms use grid-based searching to determine the best path between the origin and

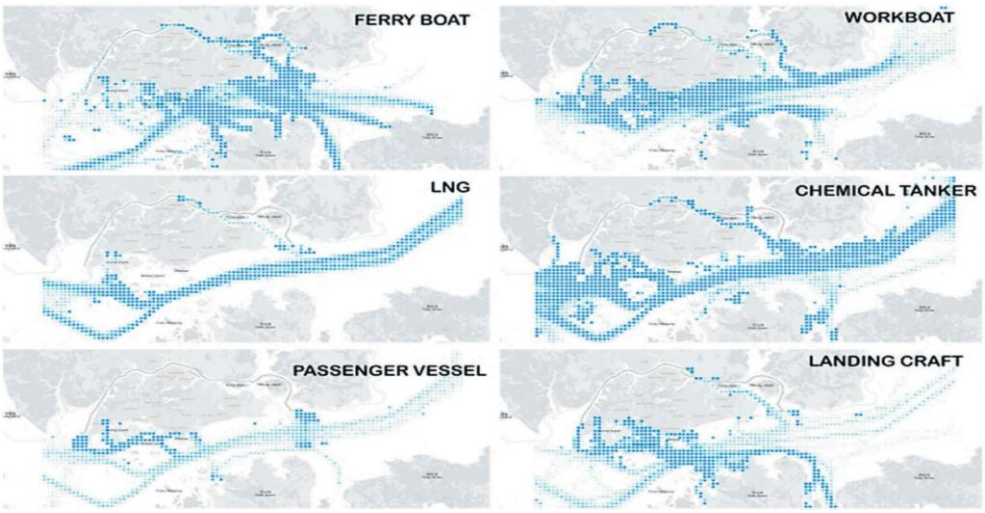


Figure 4. Waterway patterns of different vessel types.

destination, avoiding the island or land region in the process. Because Theta* uses any-angle route planning, it performs better. The following design factors are crucial for autonomous shipping in order to provide a feasible and secure path for an autonomous ship. The main considerations are as follows: (1) incorporate the waterway pattern to border the reasonable sea way area for route generation; (2) enable non-uniform cost value for individual grids, where the grid cost can be interpreted as traffic density or environmental factors that are relevant to sailing safety; (3) incorporate the maritime traffic policy or regulation modeling, such as modeling the unidirectional and bidirectional waterways or the restricted areas (land and island); (4) take into account the maneuverability of the vessel, such as angular limits for the route turning section.

Several OD route planning algorithms have been applied in our prototype system; some of the outcomes are shown in Figure 5. Figure 5 (i) and (ii) show the route planning outcomes generated by A* and Theta*, respectively, given the identical origin and destination. It is evident that Theta* produces a trajectory that is more realistic than A*. In accordance with the previously indicated design concerns, we further improve the Theta*. Grid-based cost map modeling is made possible by the enhanced Theta* method. The planned route might avoid the hotspot region to reach the target, as shown in Figure 5 (iii), where the heat map is used to depict the traffic density or environmental conditions. As shown in Figure 5 (iv), the enhanced Theta* algorithm can generate the route within the waterway pattern while avoiding the hotspots when the waterway constraint is applied. When a trip voyage is started, the route planning is either carried out by SCC or obtained via the autonomous shipping on-board system (AL5&6).

Situation awareness (SA)

Situation awareness becomes a continuously changing perspective of the changing surroundings and surrounding traffic after a vessel's destination is determined and it begins sailing on a predetermined path. Situation awareness is handled by human workers on board in conventional marine transportation systems, while in autonomous shipping, it is either totally controlled by the autonomous shipping system (AL5&6) or handled by a SCA (AL2-4). The formal definition of situation awareness, according to earlier research (Endsley 1988, 2016; Ottesen 2018), is "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley 1988). This term may be translated as: gathering and projecting existing traffic and environmental data to anticipate future conditions. The trajectories of nearby boats in the near future are the most significant navigation states that should be anticipated. While the current and slightly earlier sailing status (i.e., trajectory) serves as a crucial input for future trajectory prediction, the majority of current approaches take into account the collision risk by measuring the spatial–temporal relationship between potential future trajectories of approaching vessels. For

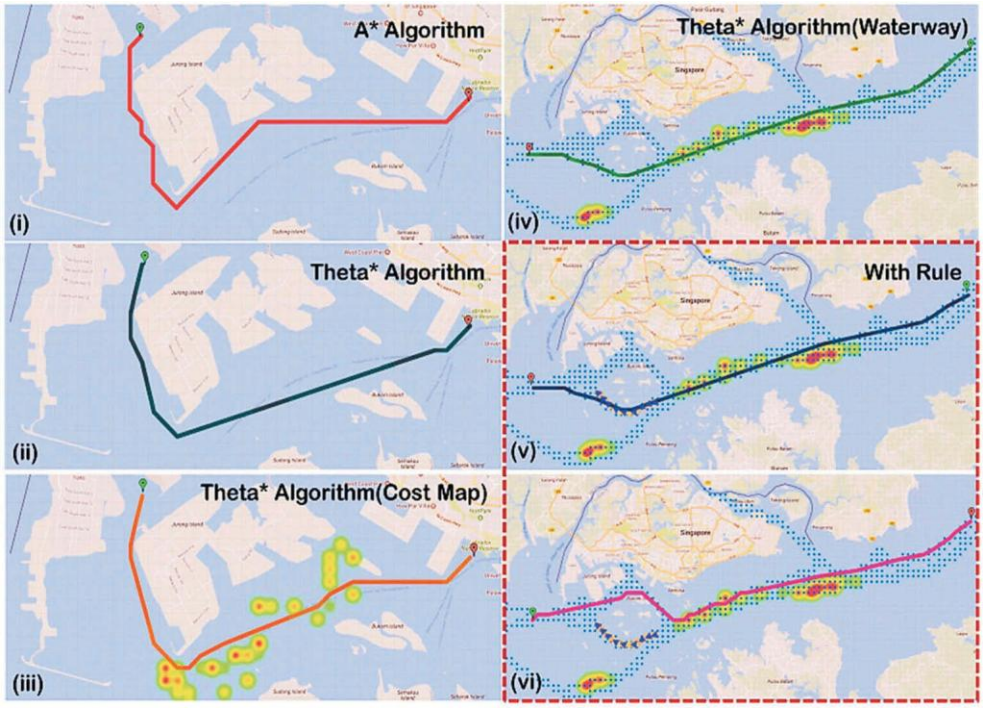


Figure 5. OD route planning in autonomous shipping.

For example, the linear regression model is used to generate vessel trajectories for traditional collision risk metrics like the closest point of approach (TCPA) and the closest point of approach (DCPA) (Bole, Wall, and Norris 2014) (Mou, van der Tak, and Ligteringen 2010; Chin and Debnath 2009; Hsu et al. 2009). Vessels' Speed over Ground (SOG), a constant Course over Ground (COG), and location (latitude and longitude) linked with a series of timestamps are often used sailing state characteristics.

As shown in Figure 6, the most widely used trajectory prediction techniques now are knowledge-based forecasting, non-linear projecting, and linear regression. The majority of prediction techniques operate on the fundamental premise that the previous state is thought to be preserved within a certain time frame. Since the regular waterway pattern may be utilized to amend the incorrect prediction by considering the probable state change, particularly for the transition between straight and turning parts of a route, the knowledge-based forecasting technique somewhat avoids this assumption. Forecasting uncertainties may be further decreased for autonomous ships by using additional state sensing data for trajectory prediction and by including anticipated future actions, such as maneuvering control in the near future, to improve forecasting accuracy. Recent research has shown that previous knowledge of upcoming maneuvering operations may improve forecasting accuracy (Borkowski 2017). It is anticipated that vessels' future trajectories will become more predictable in autonomous shipping as more state and possible future control signal data are integrated beforehand. The risk assessment (including collision risk) may provide significantly more accurate findings when trajectory prediction is improved.

By combining the possible positions of boats in the future, traffic hotspots may be forecasted, provided that the forecasting time frame is expanded to a few hours ahead. Traffic hotspots may be characterized as either a danger hotspot (defined with regard to traffic speed and density, etc.) or a traffic density hotspot (defined in terms of vessels' locations), depending on the particular definition. The route planning approach may utilize these traffic hotspots as input to reschedule traffic and reduce the hotspots, which are crucial SA data. We have included the hotspot prediction capability into our prototype system, and Figure 7 shows the predicting outcomes. For the hotspot prediction, an error-tolerant probabilistic forecasting approach is taken into consideration since a longer prediction window indicates more uncertainty (Xiao et al. 2017). The possible range that individual boats may sail within is shown by the circle in Figure 7 (i); the hotspots may be represented by the aggregating findings. In order to observe just the dynamic flow, the hotspots created by the stationary vessels are eliminated in Figure 7 (ii), which displays the hotspot findings using a heatmap. Regarding autonomous ships, traffic hotspots will also be more predictable due to the availability of more sensor data and projected future maneuvering information.

Action and decision making

The next step is to take steps to reduce the risk and release the traffic hotspots when the possible collision risk and hotspots are identified. The system intelligence for autonomous shipping is introduced by SA and prompt actions taken jointly. Figure 8 shows an encompassing diagram of the "SA phase" and "action phase." As said, there are two process branching based on the SA phase's hotspot detection and risk perception. Higher attention is given to short-term trajectory modification, which addresses the identified collision risk that is likely to happen soon. The short-term planning in Figure 8 focuses on avoiding collisions, which may be modeled as an optimization process to reduce the risk between any paired boats (keep in mind that there may be other objectives, such as efficiency for such optimization). Hotspot mitigation is often achieved over longer time windows by arranging traffic at the aggregate level, which is a lower priority operation for an autonomous ship.

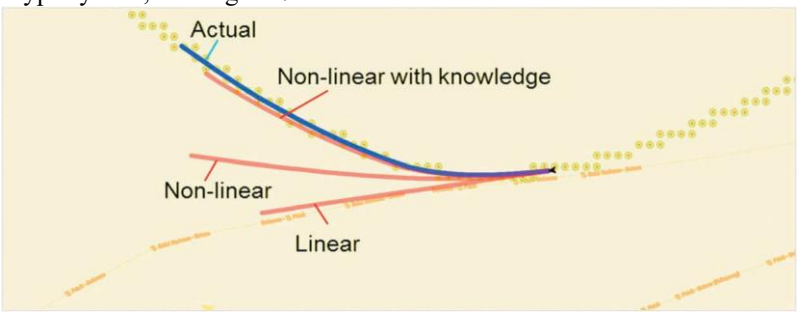


Figure 6. Predictive projection of vessel trajectory.

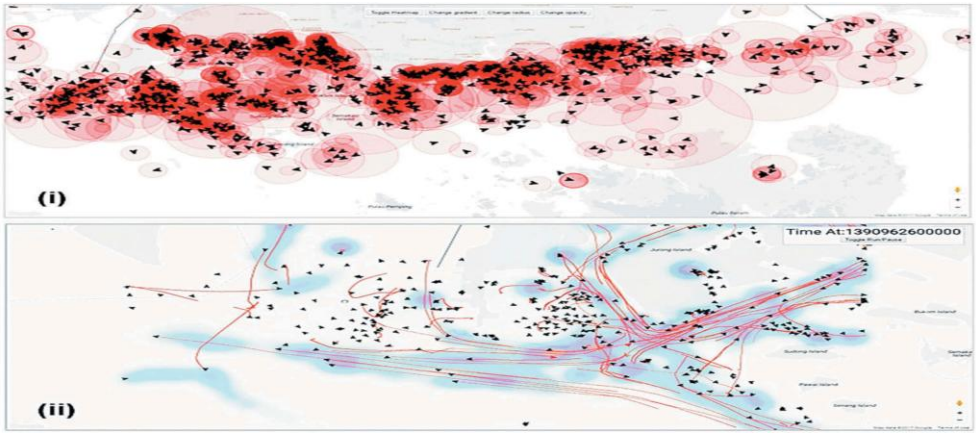


Figure 7. Hotspot and traffic congestion forecasting.

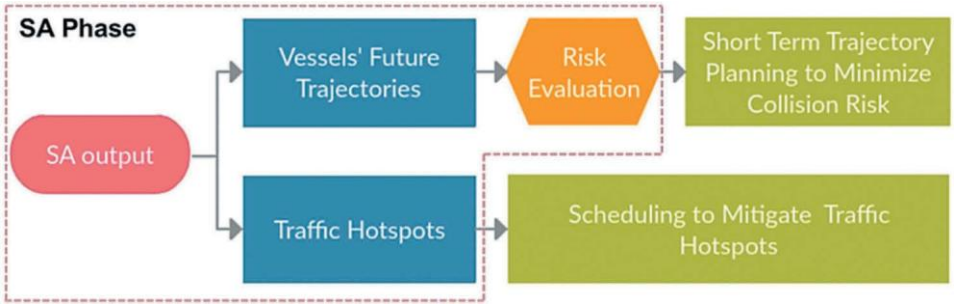


Figure 8. Actions taken to mitigate risk and traffic hotspots with SA as input.

Hotspot mitigation requires comprehensive traffic data, which is often managed by marine traffic agencies. It should be mentioned that short route adjustment operates by organizing a group of boats in order to facilitate cooperative planning among a list of potential vessels. The objective of such cooperative planning in terms of short-term trajectory planning is to provide sailing trajectories for approaching boats that are free of collisions. Cooperative route planning has been shown to be a successful way for a group of robots to prevent collisions (Frese and Beyerer 2011). Autonomous shipping management may use a similar strategy. Tree search (Frese and Beyerer 2010), elastic band (Hilgert and Hirsch 2003), and mixed integer linear programming (Borrelli et al. 2006) are some of the prominent methods.

Reinforcement learning (Wang and Phillips 2013), prioritized planning (Cap et al. 2015), and other optimization techniques (Teng, Lau, and Kumar 2017). Depending on how the autonomous shipping system is actually implemented, coordination may be achieved at the SSC or immediately amongst nearby boats. Temporal planning, or timing the traffic flow, is a better choice for hotspot reduction. It may be able to increase traffic throughput while maintaining safety constraints by improving the traffic distribution.

The SA states of vessels in the near future and a risk assessment procedure are the main foundations of the cooperative planning algorithm. When a collision occurrence is identified, the boats' behavior must adhere to the specified rule (e.g., which vessel should, in certain circumstances, yield to another). Generally speaking, boats with higher risk scores have to act to lessen the likelihood of a collision. One of the boats with a greater risk (or in accordance with the regulations) must re-plan its future trajectory for collision avoidance once the modeling procedure has verified that collisions have happened within a certain range. The TCPA and DCPA measures serve as the foundation for the commonly used risk score computation, and other additional metrics have been put out in more recent studies. Regardless of the risk metric used, one of the challenges is figuring out the appropriate collision alarm threshold. This can be learnt from the real-world functioning of the current traffic system, but system simulation may also provide insights by modifying the threshold value.

In summary, the fundamental operation for vessel movement modeling is an integrated process of "OD route planning," "SA for per-ception of future risk and traffic states," and "decision-making to mitigate the risk." While adjusting to autonomous shipping, the particular procedures in each sub-step may change, such as by adding additional sensing and control data to remove the uncertainties.

Agents defined in MarineMAS

Multiple static, dynamic, and virtual agents have been defined in MarineMAS, as shown in Figure 9. Based on their real behavior inside the system, each of these actors is free to pursue their own objectives. Navigational activities for sailing, planning, SA, and scheduling are the goals of the interaction between agents (shore station–vessel, vessel–vessel, and virtual agent–physical agent). The MarineMAS framework defines the following categories of agents:

Vessel agents may be classified as both dynamic and population agents. Agents that are transportable inside the system are referred to as dynamic agents. Within the modeling area, which is limited by established canal limits, vessel agents are allowed to move. For necessary navigational activities, each vessel may communicate with nearby vessels and shore station agents. Every vessel agent has a navigator to assist in modifying the moving trajectory and a radar module to detect nearby traffic. ● Radar is a functional agent constructed with a vessel agent that aids in situation awareness by obtaining information about nearby traffic, such as the quantity and distance of nearby vessels. Another functional agent that the vessel agent refers to for trajectory creation is the navigator. The short-term route planning algorithms are carried out by the navigator. During the runtime simulation, the associated vessel agent receives the trajectory coordinates for movement operations. One kind of static agent that is responsible for long-range route planning between traveling origin and destination is the shore station agent. Shore stations in autonomous shipping can be divided into two groups: SSC, which functions as a shipping operator and oversees a fleet of ships (in terms of remote monitoring and planning); and maritime traffic management agency/authority, which has access to global maritime traffic data and coordinates the traffic in a specific water area. In our first implementation of the prototype system, we do not distinguish the shore station agent. The shore station agent calls the predictor to forecast future marine traffic conditions.

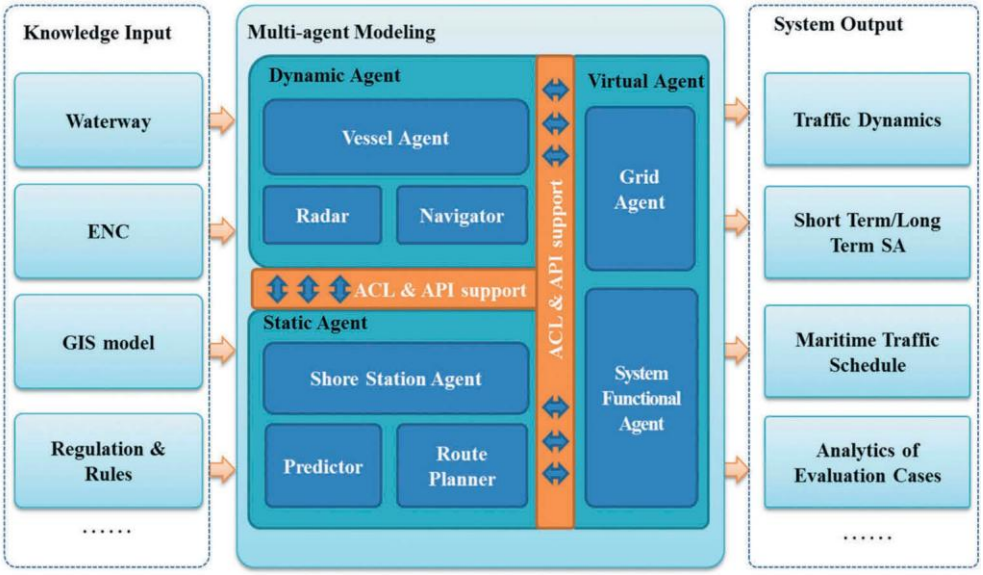


Figure 9. MarineMAS – definition of agents and system input and output.

Forecasting findings are helpful in determining what has to be done ahead of time based on the impression of future traffic. With the input of origin and destination, the route planner is in charge of creating long-range sailing routes. To accomplish optimal seaway design, historical traffic facts and maritime regulations are used. The system's grid agent is a kind of visual agent that manages all grid-based functions, including updating and retrieving the vessel list in individual grids. Better computing efficiency and geographical indexing are the primary goals of including grid agents (further information will be provided in the implementation section). The data persistence of the system intermediate and output outcomes are handled by the system functional agent. Other kinds of agents may query and get certain common shared data structures that are stored with the system functional agent.

Radar aids in obtaining the list of nearby vessel agents and distance metric for collision risk assessment, while the vessel agent initiates the navigator's short-term route planning procedure to provide a collision-free trajectory for each simulation cycle. The route planner at the shore station agent is responsible for long-term OD route planning. To facilitate route development, the route planner requires the pre-dictor's assessment of future traffic. For shared information access and data persistence modeling, both static and dynamic agents may communicate with system functional agents. Each vessel will get the global traffic data and long-range route planning produced by the shore station agent for long-range route generation navigation. In order to prevent collisions, the nearby vessel agents exchange maneuvering

information either directly or via a shore station agent.

Prototype system implementation

The first implementation of a prototype system based on the MarineMAS framework concept is covered in this part. After experimenting with a few of the MAS toolkits, we have decided to create our own prototype system from scratch, despite the fact that there are several open source toolkits available to enable MAS development. The user interface of a MAS built using Repast Simphony toolkits is shown in Figure 10. Based on the preliminary development trials, we decided to develop an original solution primarily because of the following factors: (i) the majority of current MAS modeling toolkits are made for generic purposes rather than specifically for maritime traffic system simulation; MatSim is made for large-scale agent-based transport simulation but primarily focuses on land traffic modeling; (ii) many open source toolkits adopt outdated technologies that do not benefit from the latest technological advancements, particularly with regard to large-scale modeling and in-memory computing capability; and (iii) it is also fairly difficult to modify the built-in modules, visualization, and user interfaces with new logics.

The J2EE framework is used in the development of the prototype system, which integrates cluster-supporting in-memory caching and concurrent computing libraries. In accordance with the J2EE norm, the prototype system features a layered design. The basic functions like query and update that communicate with the underlying database or in-memory caching are defined by the data access layer. When the system boots up, the commonly used data will be put into an in-memory data structure for performance reasons.

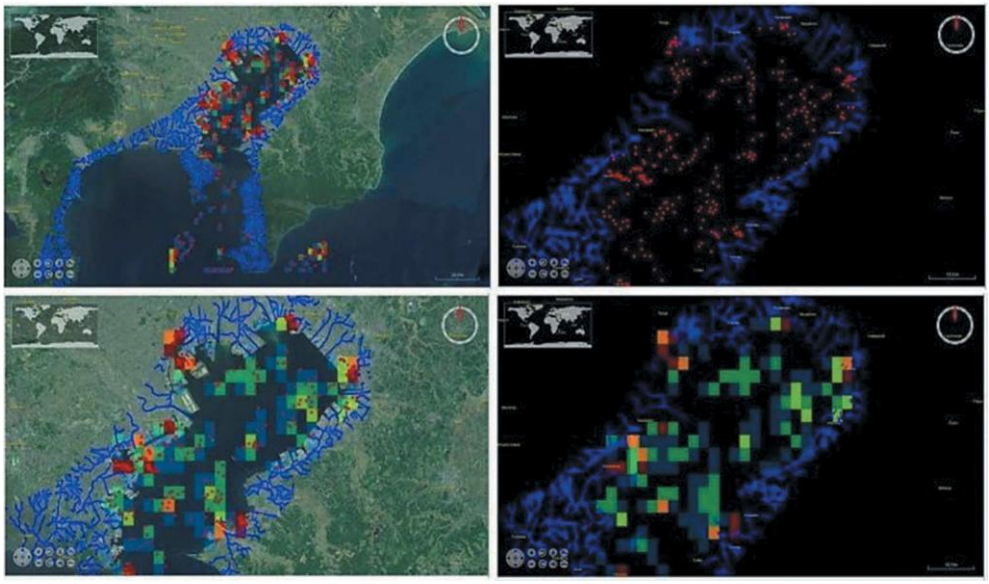


Figure 10. MAS implementation trial based on RePast.

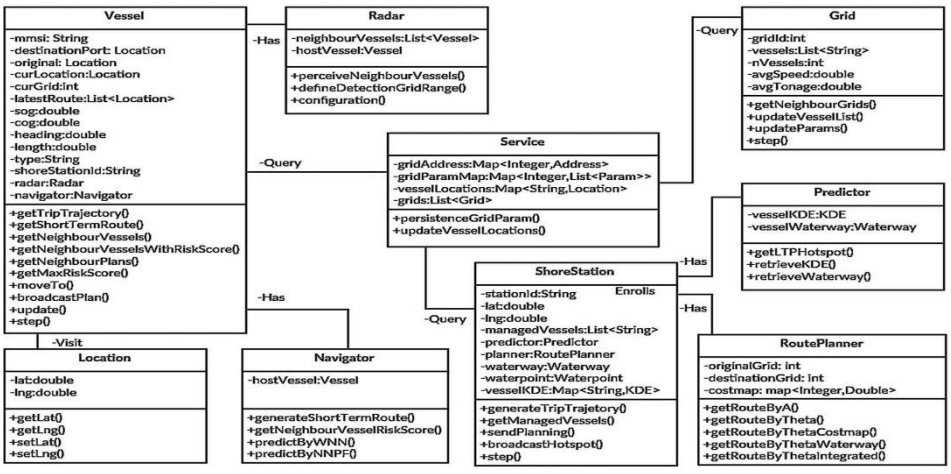


Figure 11. The UML diagram of key agent related classes in the prototype system.

up. In essence, the modeling data may be divided into two categories: supporting data to maintain the modeling configuration settings, indexes, and states for effective operation, and core data for agent-based modeling, such as

knowledge of maritime traffic and GIS data. The higher level processing logic encapsulation (planning, SA, and actions) in the function layer, also known as the service layer, may call the fundamental DAO methods in the data access layer if any data operations are needed. In order to achieve system scalability, this layer manages the cluster characteristics and in-memory data grid (across the cluster) if the simulation is operating across a cluster, apart from the fundamental aspects of MAS modeling. The user interface for dynamic modeling animation, analytic result illusion, and other system configuration information presentation is provided by the visualization layer. Although the prototype system is designed as a web-based B/S application, its modularized architecture facilitates its conversion to a desktop version. We summarize three crucial design factors—in-memory caching, concurrent computing mode, and spatial indexing—with the first system implementation to allow realtime computing and modeling efficiency.

Regarding in-memory caching, frequently used run-time data are put into memory at system startup and are often kept there. In the meanwhile, the majority of processing result data is initially stored in memory before being periodically emptied onto hard drive storage (such as a database on a hard drive). In order to lower the IO overhead for efficiency considerations, the dumping cycle is meant to be much bigger than the simulation cycle (i.e., the simulation step of agents). The in-memory data caching solution in our test enables seamless modeling and animation on the user interface (UI) for hundreds of boats, but it lags when directly reading and updating realtime data via a database. To help with in-memory caching across a multi-server cluster, several toolkits are available. For large-scale modeling, in-memory caching is practically needed.

Additionally, concurrent computing works well with MAS to describe the behavior and interactions of several autonomous agents. These days, the majority of popular servers are multi-core, multi-threaded, and in order to optimize system capacity, jobs are thought to execute concurrently. Meanwhile, certain cutting-edge solutions have been put forward to make effective use of multi-core servers' computational capacity and to obtain faster throughput, lower overhead, and lower latency for the development of highly concurrent systems. For example, Akka is a framework based on actor models that helps with the design of

creation of programs with high concurrency. "Akka" offers a considerably lighter resource unit called "actor" and well-designed message transmission between actors as compared to thread-based concurrency. As a result, each agent may be linked to a launched actor for the relevant activities and processing duties. While method call and direct access are used for functional integral entities between vessel, radar, and navigator as well as between shore station agent, predictor, and planner, communication between different agents (station–vessel and vessel–vessel) is implemented through sending and receiving messages

In order to get much greater computing efficiency, the whole prototype system depends on a grid-based architecture, which is another crucial design factor. As stated in Section II, we construct a form of visual agent called a grid agent to manage the grid-based waterway pattern, maintain the required grid-related characteristics, and manage any grid-based actions like retrieving and updating the vessel list in specific grids, among other things. The discretized grid-based design aims to accomplish:

(i) Reducing the issue size to make it easier to store and use knowledge and organize GIS data and environmental factor modeling on the target zone (keep in mind that continuous will significantly increase the data amount and make it harder to articulate the modeling context).

(ii) Facilitating effective OD route planning, SA, and actions (keep in mind that although OD route planning is intended for grid-based map configuration, any actions over map grids, such identifying vessels and obtaining traffic or environmental information for a particular zone, are much more efficient).

(iii) The encoded grids are primarily intended for spatial indexing. Individual grids are regularly labeled with integer codes, making it easy to retrieve neighboring grids (such those within a certain distance range) of any grid by performing a simple grid code computation. In contrast to R*-tree-based indexing O (logn) (n is the total number of modeled vessels; note that the computation efficiency of R*-tree establishment is O(nlogn) and it is necessary to dynamically update the indexing during run-time) or no indexing O(n2) to identify surrounding vessels and other GIS modeling information, discretized spatial structuring allows for any information access of neighboring grids in constant time with O(1) complexity, which is far more efficient. It is important to note that some procedures must be carried out, such as updating the locations of vessels and querying nearby boats.

Without effective spatial indexing, each simulation step will be very sluggish.

Figure 11 shows the MAS prototype system's classes UML diagram, which only includes the essential agent classes (the picture does not include the supporting classes for certain methods and algorithms). The aforementioned agent description defines the primary traits and procedures based on the characteristics and behaviors of specific agents. In order to accomplish data retrieval and other tasks via direct access, the vessel agent has the reference of radar and navigator and vise versa. The shore station agent's interaction with the predictor and route planner uses a similar concept. Messaging and some shared states enable cooperation between vessel agents as well as between vessel agents and shore station agents. In order to provide easy adaptation to other algorithms by overriding the processing feature, agents' characteristics and behaviors are encapsulated in appropriate granularity and offered as modularized interfaces. OD route planning, situational awareness (forecasting), risk assessment, and vessel movement in step function are among the elementary activities that we initially create and build in our implementation. We gradually add additional features and intelligence to the system since it is unlikely to directly include all the characteristics required to construct a complex and fully working system from the outset. The following two phases are intended for the implementation of the prototype system:

Stage 1: Simulation and modeling are done using the current marine traffic information. The sailing trajectory for each vessel is created using AIS data. Vessel movement is carried out in a step function, and the trace locations of the vessels on a route are interpolated to subsequent time points of equal interval. The prototype system's core features, such step execution, agent

communication, data retrieval, and data persistence, are validated as part of this stage's basic implementation. Grid-based traffic analytics (traffic dynamics including density, hotspot, and vessel number) may be used with the system put in place at this stage. In order to enable collision risk analytics, verify the visualization user interface, and facilitate traffic dynamics analysis, the virtual agent–system functional agent may record grid-based traffic dynamics.

Stage 2: Each vessel's sea path is created by generating its trajectories based on the waterway layout. "Shore Station Agent" plans each vessel's whole course based on the origin and destination of the route. The revised Theta* algorithm, which may avoid the island or land region to create the route with the lowest cost in terms of a given angular limitation for various kinds of boats (based on vessels' maneuvering characteristics), is the primary basis for the whole trajectory planning process. In order to simulate safe maritime sailing, SA and collision avoidance activities are performed at this step. Several techniques for cooperative route planning, include decision trees, elastic bands, integer linear programming, prioritized planning, and reinforcement learning

may be used to simulate how boats avoid collisions. Each marine agent's "Navigator" is equipped with these technologies. Both the cooperative vessel motion dynamics via interaction between various vessels and the whole route planning through interaction between "Vessel Agent" and "Shore Station Agent" provide the system intelligence. Each vessel agent should detect the traffic state, anticipate the potential risk (SA), and dynamically plan the collision-free sailing course (actions) during this implementation stage. Eventually, this will be able to mimic the behavior of an autonomous ship.

As some results are shown in Section II, we have implemented the majority of algorithms for OD route planning, situational awareness (perception and forecasting), and risk appraisal up to this point. Stage 1 has been successfully implemented. Figures 12 and 13 show the modeling and analytics user interfaces, respectively. Additionally

In order to fully actualize the capabilities specified in Stage 2 implementation, efforts will concentrate on creating a cluster-enabled backend. Additionally, the MarineMAS framework will be used to facilitate scenario and strategy-based evaluation for autonomous shipping systems.

Conclusion

In conclusion, autonomous shipping systems require more than just shipping building and sensing and control technologies from a hardware perspective. At the system level, autonomous shipping necessitates an appropriate software modeling process to enable the design and assessment of real-world practices such as planning, scheduling, and cooperation of autonomous ships for realistic traffic operations. To support the modeling and assessment of autonomous maritime systems, a MAS is suggested. In this article, we primarily address framework architecture and the necessary elements and justifications to allow autonomous

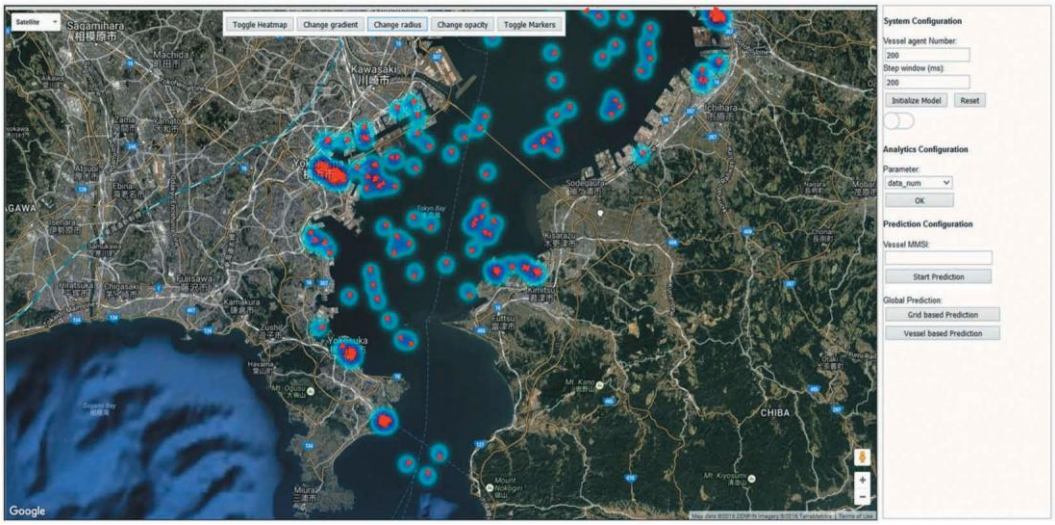


Figure 12. The modelling UI of the prototype system.

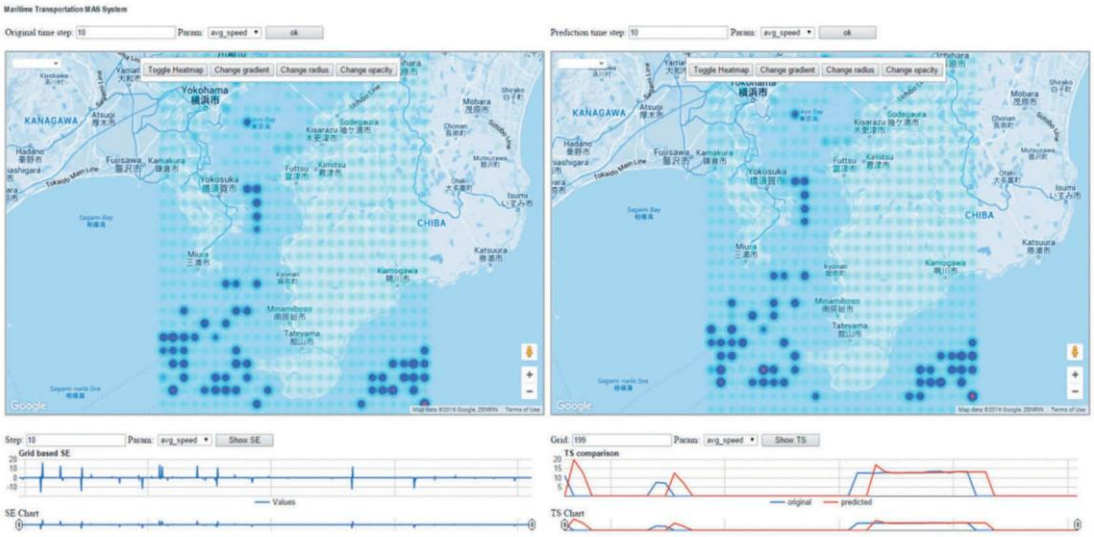


Figure 13. The analytics UI for situation awareness.

shipping evaluation. The framework adopts a hierarchy layered design to establish integration of the essential data and knowledge, MAS modelling and analytics. We clarify the essential autonomy processing logics from the aspects of OD route planning, situation awareness, and shipping decision-making for autonomous shipping modelling. Accordingly, interactions among different agents, representing shore station–vessel, vessel–vessel, and virtual agent–physical agent, are defined.

A prototype system has been implemented from scratch based on Java technologies to demonstrate the key features of MarineMAS framework. The implemented prototype system provides a development environment to accommodate route planning, situation awareness, and actions for autonomous shipping evaluation and management. The modularization and a set of universal interface defined in the prototype system enables adaption to different algorithms for simulation and modelling. The framework relies on grid-based design to introduce high computational efficiency. The experience gained from the prototype system development has been sum-marized. We believe the critical thinking on framework design and essential processing components may help the interested readers on their MAS system implementa-tion. One of our research focuses in our future explora-tions is to improve the runtime modelling efficiency through a cluster-based backend with in-memory data grid technologies. With the large-scale modelling cap-ability, MarineMAS is expected to be used for evaluating various scenarios, testbed trials, and proof of autono-mous shipping strategies and to offer valuable insight and guidelines for actual autonomous shipping system that will come into services.

References

"AAWA Project Introduces the Project's First Commercial Ship Operators [Online]." Accessed 30 December 2017. https://www.rolls-royce.com/media/press-releases/yr-2016/pr-12-04-2016-aawa-project-introduces-projects-first-commercial-operators.aspx

Bergenti, F, and E Vargiu 2010 Multi-agent systems in the industry. Three notable cases in Italy. In: CEUR workshop proceedings 11th workshop from objects to agents (WOA), vol 621. Rimini, pp 1–6

Blanke, M., M. Henrique, and J. Bang, "A Pre-Analysis on Autonomous Ships [Online]." 2017. Accessed 27 August 2018. https://www.dma.dk/Documents/Publikationer/Autonome.skibe_DTU_rapport_UK.pdf

Bole, A., A. Wall, and A. Norris. 2014. *Radar and ARPA Manual*. Oxford, United Kingdom: Butterworth-Heinemann

Borkowski, P. 2017. "The Ship Movement Trajectory Prediction Algorithm Using Navigational Data Fusion." *Sensors* 17 (6): 1–12. doi:10.3390/s17050968.

Borrelli, F., D. Subramanian, A. U. Raghunathan, and L. T. Biegler. 2006. "MILP and NLP Techniques for Centralized Trajectory Planning of Multiple Unmanned Air Vehicles." *2006 Am. Control Conf.*, Minneapolis, Minnesota, USA, 5763–5768.

Burmeister, H.-C., W. Bruhn, Ø. J. Rødseth, and T. Porathe. 2014a. "Autonomous Unmanned Merchant Vessel and Its Contribution Towards the e-Navigation Implementation: The MUNIN Perspective." *International Journal of e-Navigation and Maritime Economy* 1: 1–13. doi:10.1016/j. enavi.2014.12.002.

Burmeister, H.-C., W. C. Bruhn, Ø. J. Rødseth, and T. Porathe. 2014b. "Can Unmanned Ships Improve Navigational Safety?" *Transport Research Arena*, Paris. 1–10.

Cap, M., P. Novak, A. Kleiner, and M. Selecky. 2015. "Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots." *IEEE Transactions on Automation Science and Engineering* 12 (3): 835–849. doi:10.1109/TASE.2015.2445780.

Chin, H. C., and A. K. Debnath. 2009. "Modeling Perceived Collision Risk in Port Water Navigation." *Safety Science* 47 (10): 1410–1416. doi:10.1016/j.ssci.2009.04.004.

"China Completes Unmanned Ships R&D Project [Online]." Accessed 30 December 2017. https://www.marinelink.com/news/completes-unmanned396151.aspx

Daniel, K., A. Nash, S. Koenig, and A. Felner. 2007. "Theta*: Any-Angle Path Planning on Grids." *Journal of Artificial Intelligence Research* 39: 533–579. doi:10.1613/jair.2994.

Endsley, M. R. 1988. "Design and Evaluation for Situation Awareness Enhancement." *Proceedings of the Human Factors Society Annual Meeting* 32 (2): 97–101. doi:10.1177/154193128803200221.

Endsley, M. R. 2016. *Designing for Situation Awareness: An Approach to User-Centered Design*. 2nd ed.Boca Raton, Florida, United States: CRC Press

Frese, C., and J. Beyerer. 2010. "Planning Cooperative Motions of Cognitive Automobiles Using Tree Search Algorithms." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6359 (LNAI): 91–98.

Frese, C., and J. Beyerer, "A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles," 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, 2011, pp. 1156-1162.

"Guidelines for Autonomous Shipping [Online]." 2017. Accessed 30 December 2017. https://www.veristar.com/portal/veristarinfo/files/sites/veristarinfo/web.contents/bv-content/generalinfo/giRulesRegulations/bvRules/rulenotes/documents/641-NI_2017-12.pdf

Guo, Q., and M. Zhang. 2009. "A Novel Approach for Multi-Agent-Based Intelligent Manufacturing System." *Information Sciences* 179 (18): 3079–3090. doi:10.1016/j. ins.2009.05.009.

Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." *IEEE Transactions on Systems Science and Cybernetics* 4 (2): 100–107. doi:10.1109/TSSC.1968.300136.

Hilgert, J., K. Hirsch, T. Bertram, and M. Hiller, "Emergency path planning for autonomous vehicles using elastic band theory," Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), Kobe, Japan, 2003, pp. 1390-1395