“Rules of the Road” Navigation with Logic, Theorem Proving and a Large Ontology

James Timberlake
Naval Postgraduate School
Monterey, CA
james.timberlake@nps.edu

Adam Pease
Naval Postgraduate School
Monterey, CA
adam.pease@nps.edu
Corresponding Author

For (CSCI-RTCS)

***Abstract*—The deployment of maritime autonomous surface ship (MASS) is accelerating across various domains, driven by an expanding range of use cases. In non-military applications, MASS are utilized for oceanographic research, marine resource exploration, environmental monitoring, search and rescue operations, and commercial shipping [1]. In the military domain, MASS are employed for missions such as intelligence gathering, surveillance, deception, protection, and offensive operations targeting maritime infrastructure. A critical concern shared across these domains is the ability of an autonomous vessel on the high seas to adhere to the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs), or the “Rules of the Road” [2]. Our work employs typed first order logic with arithmetic and a large ontology along with automated theorem proving to create a decision support system for navigation that is compliant with the COLREGs.**

***Index Terms*—COLREGs, ontology, logic, decision support system, theorem proving**

I. Introduction

We used the Suggested Upper Merged Ontology (SUMO) [3], [4], and associated inference technology [5]–[7] to develop a decision-aid framework for autonomous vessels that ensure COLREGs-compliant navigation.

SUMO is a comprehensive upper-level ontology and collection of domain ontologies that provides a framework for integrating knowledge across domains. It captures foundational concepts such as time, space, events, and objects, and offers a platform for structuring domain-specific data. SUMO employs higher-order logic [8], enabling knowledge engineers to declare constraints on actions through formal rules [3]. These declarative rules allow for validation through theorem proving [9], creating systems that can reason and justify decisions autonomously. By embedding decision-making processes within the logic itself, SUMO eliminates the need for external procedural implementation, streamlining the development of decision-support systems. SUMO’s large inventory of concepts and definitions provides for reuse, lessening the burden of formalizing concepts. By creating new content within the context of a large ontology, it ensures that changes in scope or context of the project do not expose hidden assumptions that would break existing models [10]. SUMO will be explained further in section II-C.

The primary objective is to formalize the decision-making logic necessary for autonomous vessels to identify and navigate close-quarters scenarios in compliance with applicable COLREGs rules. By representing these rules in a formal logic, the research aims to bridge the gap between theoretical maritime regulations and practical autonomous navigation, enabling MASS to reason about and execute collision-avoidance maneuvers with transparency and traceability.

This research relies on several key assumptions:

- **Situational Awareness**: The autonomous vessel is assumed to have situational awareness (SA) of its own position, course, speed, and that of surrounding vessels in its vicinity.
- **Sensor Inputs**: It is presumed that the vessel’s radar and sensor systems provide accurate real-time inputs, such as closest point of approach (CPA), course over ground (COG), speed over ground (SOG), range and relative bearings of nearby vessels.
- **Focus on Decision-Making**: The scope of this research is confined to formalizing the decision-making process using ontology and automated reasoning tools, specifically for identifying COLREGs scenarios and determining compliant actions. Path replanning and maneuver execution are outside the scope of this work.

If integrated with onboard radar systems, the ontology and reasoning engine could evaluate navigational scenarios in real-time, offering several benefits:

- **Rule Applicability Assessment**: The system identifies which COLREGs rule applies to a given situation, reducing ambiguity for operators.
- **Collision Avoidance Recommendations**: By analyzing vessel dynamics and relative positioning, the system suggests actionable maneuvers, such as altering course or speed, to avoid collisions.
- **Enhanced Situational Awareness**: The tool provides clear, rule-based insights to operators, enabling more informed and confident decision-making.
- **Training**: This tool, coupled and integrated with simulation software, could provide direct Rules of the Road training for maritime navigators.

Unlike traditional procedural software testing, which relies on partial coverage through test cases, formal logic allows for rigorous, static validation, ensuring that the reasoning processes are provably error-free. If a logical contradiction is found, theorem proving provides a proof that precisely describes the inconsistency. This guarantees a higher degree of reliability and robustness, essential for the high-stakes environment of maritime navigation.

II. Background

A. COLREGs

The crux of almost every close-quarters situation between two vessels at sea is that of COLREGs. The International Maritime Organization (IMO) formalized The Rules of the Road in 1972 and they came into effect in 1977, and since then there have been over 60 amendments to these rules. The United States adopted the International Rules in 1977 and later developed the Inland Rules in 1981; this research will focus on the International Rules [2]. COLREGs consist of 38 rules split between five sections (A-E): General, Steering and Sailing Rules, Lights and Shapes, Sound and Light Signals, and Exemptions. This research will focus on a decision aid for portions of Part B, Steering and Sailing Rules. COLREGs are built around two vessels interacting in a one-on-one encounter, leaving ambiguity for a situation involving more than two vessels. This responsibility normally falls on the Ship's Master to safely apply the rules to avoid collisions. Part of this research will be to both aid in the COLREGs-compliant decision making as well as investigate how an ontology can adjudicate complex and abstract situations.

B. Ontology Languages

"An ontology defines a set of representational primitives with which to model a domain of knowledge" [11]. The primitives in this case are the relationships between vessels on the high seas that can be modeled to represent the knowledge of COLREGs. The current research in this field primarily utilizes the Semantic Web standards such as Resource Description Framework (RDF) and Web Ontology Language (OWL) to assist autonomous vessels. These standards are effective for certain applications but employ a limited logic (description logic) that is less expressive than well-known formalisms like first order logic. An expressive logic can better represent the definitions of abstract concepts such as time, space, events, and objects whereas the Semantic Web operates at a more granular level of machine-readable metadata. This research aims to improve on prior work by presenting a novel approach to collision avoidance decision aids for autonomous vessels by use of SUMO.

C. Ontologies and MASS

Prior work in this area has typically provided a set of concepts with a type structure and relationships. That approach does not fully capture rules that define the concepts. If the logic used is not as expressive as first order logic, definitions of terms must be either expressed informally as English comments, omitted entirely and left to intuition about the term names, or encoded in a separate procedural language that is not subject to static proof of consistency.

The research described in [12] proposes an ontology-based approach to enhance MASS navigation safety and collision avoidance decision-making. The authors construct an ontology model for navigation scenarios, integrating structured representations of scene elements, such as autonomous ships, environmental factors, and obstacles. The model captures entity attributes and relationships to facilitate semantic understanding of navigation scenes. [13] explores the use of ontologies as a foundational framework to enhance autonomous ship functionality, focusing on semantic data integration, improved SA, and effective decision-making. [14] explores the integration of ontologies within the 4D/RCS (Real-Time Control System) architecture to enhance decision-making in autonomous vehicles and improve path planning and collision avoidance. The study introduces an ontology-based model to represent environmental objects and their attributes, such as rigidity, movability, and crushability. These attributes support the system in collision damage assessment, guiding path replanning, and real-time decision making. The work in [15] explores the application of ontology-based methods to enhance semi-automatic communication in collision avoidance scenarios at sea. The authors developed ontologies tailored to navigational and communication processes to facilitate automated generation and interpretation of messages, incorporating elements of IMO's Standard Marine Communication Phrases.

D. SUMO and Reasoning

The SUMO[1] is a comprehensive ontology that includes an domain-independent upper ontology and dozens of domain-specific ontologies that are all mutually compatible. Originally proposed by the IEEE-sanctioned Standard Upper Ontology (SUO) Working Group, SUMO was developed to address the need for a shared, formal representation of knowledge across diverse domains, including engineering, philosophy, artificial intelligence (AI), and information science [3].

One of the core original purposes of SUMO was to provide a formalized foundation for domain-specific extensions. These extensions enable the ontology to adapt to specialized fields while maintaining interoperability with its general structure. SUMO has since been extended for applications in biology, transportation, military logistics, medicine and dozens of other domains [3].

The development and application of SUMO is supported by the Sigma Knowledge Engineering Environment (SigmaKEE), a platform designed to facilitate ontology browsing, editing, and reasoning [16]. SigmaKEE integrates with external resources such as WordNet [17] to enhance its natural language processing capabilities, enabling semantic interpretation and linguistic reasoning.

SigmaKEE automatically converts SUMO into the Thousands of Problems for Theorem Provers (TPTP) family of logical languages, making it compatible with standard automated

[1]https://www.ontologyportal.org

reasoning tools. These transformations, explained in [5], [18], [19], enable SUMO to support logical verification, decision support, and system validation, expanding its applicability to domains requiring rigorous formal reasoning.

We employ the Vampire theorem prover [20] for reasoning with SUMO. Vampire is an automated theorem prover (ATP) designed to automate the proof-search process for logical theories. It supports first-order logic (FOL) as well as several different logics beyond first order. It is typically the winner in most of the divisions it competes in at Computer Automated Systems Competition (CASC) [20]. Vampire supports a range of reasoning tasks, including theorem proving, model generation, and satisfiability checking, making it particularly suitable for applications in formal verification, artificial intelligence, and knowledge representation [21].

III. Implementation

The formalization process for the COLREGs is structured into three key components. First, foundational definitions and attributes are established, providing the essential building blocks for the system. Next, specific rules from COLREGs, such as *head-on*, *crossing*, and *overtaking* situations, are translated into formal logic to enable systematic reasoning. Finally, helper functions and auxiliary mechanisms are introduced to facilitate the application of these rules in dynamic and complex maritime scenarios.

Building on this principle, it is assumed that the vessels in the generated scenarios are *mutually visible at time X* and *concurrently translocating at time X*. These assumptions are represented in the code snippets by the predicates:

```
(mutuallyVisibleAtTime ?VES1 ?VES2 ?TIMENOW)
(concurrentTranslocationAtTime
    ?VES1 ?VES2 ?TIMENOW)
```

These assertions are currently hard-coded to establish that, at time `?TIMENOW`, `?VES1` and `?VES2` are visible to one another and underway.

A. Radar Inputs

The reasoning software is presumed to receive a range of inputs from an onboard radar system, enabling real-time decision-making. These inputs pertain to both the user vessel (denoted with the attribute `OwnVessel`) and the other vessel involved in the interaction (denoted with the attribute `TargetVessel`). Since maritime conditions are dynamic, all input data is tied to specific vessels and evaluated as snapshots in time. The formal definitions of various terms summarized below can be found in [22].

The radar inputs include:

- **SOG**: The speed of the vessel in relation to the ground.
- **COG**: The course the vessel is following with respect to true north.
- **CPA**: The minimum distance between two vessels if they continue on their current courses and speeds.
- **CPAt**: The time until CPA (CPAt) is the time remaining until the vessels reach their closest point. It is common for radar systems to display a negative value for CPAt if it as already passed.
- **Relative Bearing**: The angle between the two vessels, relative to own vessel.
- **Range**: The distance between the two vessels.

The user input includes:

- **Captain's Thresholds for CPA and CPAt**: Defined as `cpaThresh` and `cpaTThresh`, these parameters determine what constitutes a close-quarters situation based on the Captain's judgment. For example, a `cpaThresh` and `cpaTThresh` of 2nm and 10 minutes respectively would indicate that risk of collision exists if there's another vessel with a CPA of less than 2nm in less than 10 minutes.

These inputs form the foundational dataset for the reasoning engine, enabling it to assess compliance with COLREGs and to recommend or execute appropriate maneuvers.

B. Vessel Types

The types of vessels that are defined in COLREGs that this work considers are power-driven vessels (PDVs), Sailing Vessel, Vessel Engaged in Fishing, Vessel Constrained by Draft, Vessel Restricted in Her Ability to Maneuver, and Vessel Not Under Command. The relationship between these vessels and who has the right-of-way is mostly defined by Rule 18. The exception to Rule 18 is Rule 13 (overtaking), and the handling of that will be discussed shortly.

In SUMO, there is a class `WaterVehicle`; all the vehicle types mentioned above are instances of this class and the different vehicle types are modeled as attributes. In SUMO, entities may not have different class membership during their lifetimes. The vehicle types above are not their own class because a vessel could at one point in time be a PDV, but as soon as it puts up a sail and turns off its engine it becomes a Sailing Vessel. At all times, these vessels are instances of the class of water vehicles, and, at a certain time they have only one vessel type `Attribute`.

An PDV is represented in SUO-KIF as:

```
(=>
  (and
    (instance ?VES1 WaterVehicle)
    (attribute ?VES1 PowerDrivenVessel))
  (not
    (or
      (attribute ?VES1 NotUnderCommand)
      (attribute ?VES1 RestdInAbilToManeuver)
      (attribute ?VES1 ConstrainedByDraft)
      (attribute ?VES1 VesselEngagedInFishing)
      (attribute ?VES1 SailingVessel))))
```

This can be read in plain English as, *if there is an instance of a Water Vehicle, and that vehicle has the attribute of a PDV, then it does not have the attribute of a vessel not under command, restricted in her ability to maneuver, constrained by draft, engaged in fishing, or sailing*.

C. Hierarchy

Rule 18 establishes the hierarchy between vessels and the only exception to this rule is overtaking. For example, a PDV is to keep out of the way of a Sailing Vessel, however, if the Sailing Vessel is overtaking the PDV, the roles reverse and the PDV is to maintain course and speed and the Sailing Vessel shall give-way and pass at a safe distance.

Below is the formalization for a Sailing Vessel in SUO-KIF. A similar definition exists for all vessel types. Some term names are shortened due to space constraints.

```
(=>
  (and
    (instance ?VES1 WaterVehicle)
    (attribute ?VES1 SailingVessel)
    (instance ?VES2 WaterVehicle)
    (mutuallyVisAtTime ?VES1 ?VES2 ?TIMENOW)
    (or
      (attribute ?VES2 NotUnderCommand)
      (attribute ?VES2 RestdInAbilToManeuver)
      (attribute ?VES2 ConstrainedByDraft)
      (attribute ?VES2 VesselEngagedFishing))
    (not
      (overtakingSitn ?VES2 ?VES1 ?TIMENOW)))
  (and
    (giveWayVessel ?VES1 ?VES2 ?TIMENOW)
    (standOnVessel ?VES2 ?VES1 ?TIMENOW)))

(=>
  (and
    (instance ?VES1 WaterVehicle)
    (attribute ?VES1 SailingVessel)
    (instance ?VES2 WaterVehicle)
    (attribute ?VES2 PowerDrivenVessel)
    (mutuallyVisAtTime ?VES1 ?VES2 ?TIMENOW)
    (not
      (overtakingSitn ?VES1 ?VES2 ?TIMENOW)))
  (and
    (giveWayVessel ?VES2 ?VES1 ?TIMENOW)
    (standOnVessel ?VES1 ?VES2 ?TIMENOW)))
```

In an effort to not explain code line-by-line, this section handles a hierarchy situation for a sailing vessel, where either the target vessel is not overtaking the sailing vessel, or the Sailing Vessel is not overtaking the target vessel. By explicitly indicating an overtaking situation does not exist, the reasoner will land on a refutation for who gives-way to who in accordance with Rule 18 (hierarchy).

D. Conclusive Terms

Several terms used throughout these axioms are considered conclusive, meaning they provide the operator (or route-planning software of a MASS) with the information needed to make a decision. These terms represent key questions, such as *am I the give-way vessel?* or *What is my next obligatory action?*

The conclusive terms include:

- **Obligatory Next Action**: Specifies the baseline action a vessel is expected to take in accordance with COLREGs. Examples include:
  - Alter course to starboard
  - Slow down
  - Back-down engines
  - Pass port-to-port
  - Pass at a safe distance
  - Maintain course and speed
- **Give-way Vessel**: Identifies the vessel required to take proactive measures to avoid a collision, as per Rule 16.
- **Stand-on Vessel**: Identifies the vessel expected to maintain its course and speed, as per Rule 17, unless circumstances necessitate otherwise.

Each rule is articulated through multiple definitions to account for all possible variations of the scenario. For instance, an overtaking situation can manifest as *I am overtaking another vessel* or *I am being overtaken by another vessel.* Additionally, it is essential to explicitly define the negation of each rule, such as *I am NOT overtaking them* or *I am NOT being overtaken by them.* Formalizing these negations is critical to ensure the reasoning engine can evaluate scenarios comprehensively and determine when a rule does not apply.

It should be mentioned that classical logic does not have the default method of negation by failure, such as is found in the logic programming language Prolog. In Prolog and similar languages, if a system cannot conclude X then it will conclude (not X). This is expedient, since proving that something is not true is not guaranteed to terminate in first order logic. By taking this shortcut, Prolog optimizes performance, but this is at the potential cost of reaching unjustified conclusions.

IV. Evaluation

The evaluation of the developed ontology and reasoning framework follows a structured and iterative process to ensure logical consistency and validate scenario-based rule compliance. The steps undertaken are outlined below:

1) **Base Ontology Conversion:**
   - Convert the foundational ontologies into typed first-order form (TFF) format.
   - This initial TFF file does not contain any scenario-specific data but includes the defined axioms and the minimal SUMO ontology necessary for reasoning.
2) **Scenario Generation:**
   - Construct an itemized database of test scenarios covering a range of probable maritime situations.
   - Format the necessary scenario-specific parameters into TFF syntax.
   - Generate a corresponding visual representation of each scenario for verification and interpretability.
3) **Scenario Integration:**
   - Append the scenario-specific TFF data to the base TFF file.
4) **Consistency Checking:**
   - Execute Vampire on the combined TFF file, with no conjecture, to check for logical consistency.
5) **Conjecture Execution:**
   - Append one conjecture at a time to the scenario.
   - Run Vampire for each query individually.
   - Catalog the output, including Vampire refutation outcomes and execution times.

6) **Iteration Across Scenarios:**
   - Repeat steps 3–5 for all test scenarios to ensure comprehensive evaluation.
7) **Result Compilation:**
   - Aggregate all recorded results into a structured `.csv` file.
   - The compiled results serve as the basis for performance evaluation, statistical analysis, and validation of the reasoning framework.

V. Model Consistency and Satisfiability

Ensuring the validity of the model requires evaluating both its consistency and satisfiability. A set of axioms is satisfiable if there exists at least one interpretation in which all statements hold true [23]. Conversely, a model is considered consistent if no contradiction can be derived from its axioms. While these concepts are closely related, they are distinct: a consistent model may still be unsatisfiable, and proving satisfiability is often significantly more challenging than demonstrating consistency.

A. What Can be Said: Local Consistency

This research establishes that the model exhibits *local consistency*—meaning within the scope of tested conjectures, no contradictions arose. Specifically, every conjecture formulated as a theorem to be proven yielded the expected results, with Vampire providing refutations and correct logical consequences. This indicates that the subset of axioms involved in each proof was internally consistent.

Furthermore, at no point did the model produce a counter-satisfiable result. That is, no refutation was found that negated a previously established theorem. This reinforces the notion that the axioms, at least within the bounds of individual proofs, maintain local consistency.

The following conjectures were applied to each scenario:

1) **Crossing Situation:** Is there an `X` and `Y` in a crossing situation?
2) **Head-on Situation:** Is there an `X` and `Y` in a head-on situation?
3) **Overtaking Situation:** Is there an `X` overtaking a `Y`?
4) **Give-way Vessel:** Is there an `X` that is required to give-way to `Y`?
5) **Stand-on Vessel:** Is there an `X` that is required to stand-on to `Y`?
6) **Own Vessel's Obligation:** What is the obligatory next action of own vessel?
7) **Target Vessel's Obligation:** What is the obligatory next action of the target vessel?

It should be noted that not every conjecture will yield an answer for every scenario. Only one of the first three conjectures (1–3) may return a result, as a scenario can only represent one type of situation at a time. COLREGs does discuss special situations with more than one vessel, however, that is beyond the scope of this work. The remaining conjectures (4–7) generally return results for all applicable scenarios, provided the situation necessitates a give-way or stand-on action.

B. Example Crossing Situation: Give-way

The scenario shown in Figure 1 represents another type of crossing situation between Blue and Red. The parameters defining the scenario are as follows:

- **Blue:** COG = 293°, SOG = 16 knots
- **Red:** COG = 245°, SOG = 6 knots
- **Relative Bearing of Red from Blue:** 45°
- **CPA:** 2 nm
- **CPAt:** 18 minutes
- **Range:** 4 nm

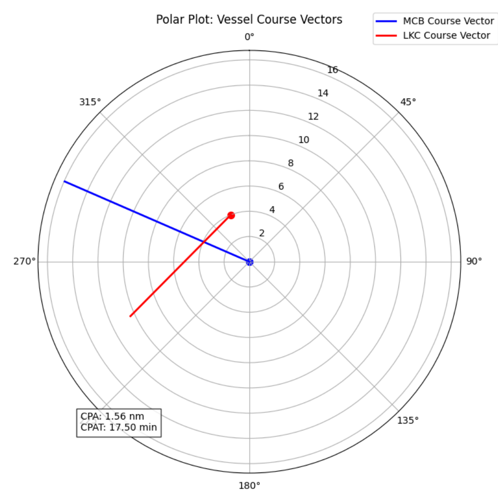

Fig. 1. Example of a Crossing Situation as the Give-way.

The relative bearing of 45° and aspect angle indicate that Red is approaching Blue from its starboard side. This scenario aligns with a crossing situation where the vessel on the starboard side (Red) is the stand-on vessel, while the vessel on the port side (Blue) is the give-way vessel.

Although from a birds-eye view this can be considered the same type of situation as the previous crossing example. However, in first-person point of view, specifically from the perspective of the Blue vessel, this is different in that own vessel is now to give-way instead of stand-on.

C. Summary of Validating Requirements

The final test set included 288 distinct scenarios, with the distribution of scenario types summarized in Table I. The distribution was not perfectly balanced across different scenario types.

TABLE I
SUMMARY OF SCENARIO TESTS

Situation Type	Count	Situation Variation	Count
Crossing Situations	88	Blue Stand On	38
		Red Stand On	50
Overtaking Situations	68	Blue Stand On	3
		Red Stand On	65
Head On Situations	16		
No Situation Found	116		
Total Scenarios	288		
Average Time to Refutation	0.531s		

The model demonstrated a high level of accuracy, with 171 of the 172 scenarios that resulted in a refutation yielding the expected results—representing a 99.4% true positive rate. Only one scenario produced an unexpected outcome, corresponding to a false positive rate of 0.58%. The cause of this anomaly is currently unknown.

It is important to note that the evaluation of results involves a degree of subjectivity. Unlike traditional classification tasks with an absolute ground truth, maritime scenarios often involve context-dependent judgment. The model's results are closely tied to the author's interpretation of COLREGs, which may differ from another mariner's perspective. This subjectivity is especially relevant in scenarios where no situation was identified. While many of these outcomes were objectively correct—such as scenarios with a negative CPAt, indicating that the vessels had already passed their closest point of approach—others could be open to interpretation depending on the evaluator's experience and judgment.

This work aims to reduce such ambiguity by formalizing COLREGs in a machine-interpretable framework. However, this does not imply universal consensus among maritime professionals regarding all outcomes.

VI. CONCLUSION

The validation testing demonstrated that the model effectively adheres to the Rules of the Road. No evidence of logical contradictions was observed, confirming local consistency within the tested scenarios. While the model successfully identified crossing, overtaking, and head-on situations, it also correctly determined when no formal COLREGs situation existed.

This research demonstrates the feasibility and effectiveness of formalizing COLREGs within an ontology-based framework to support autonomous vessel decision-making. By leveraging the Vampire ATP and the SUMO ontology, the model successfully identifies and classifies maritime situations with a high degree of accuracy, while adhering to the logical principles of COLREGs. The findings of this study lay the foundation for future advancements in transparent autonomous maritime navigation, contributing to safer and more efficient operations at sea.

REFERENCES

[1] P. Yang, J. Xue, and H. Hu, "A bibliometric analysis and overall review of the new technology and development of unmanned surface vessels," *Journal of Marine Science and Engineering*, vol. 12, no. 1, 2024. [Online]. Available: https://www.mdpi.com/2077-1312/12/1/146

[2] United States Coast Guard, *Rules of the Road*. Washington, D.C.: United States Department of Transportation, United States Coast Guard, 1992.

[3] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, ser. FOIS '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 2–9. [Online]. Available: https://doi.org/10.1145/505168.505170

[4] A. Pease, *Ontology: A Practical Guide*. Articulate Software Press, 2011. [Online]. Available: https://books.google.com/books?id=Mvs3twAACAAJ

[5] A. Pease and S. Schulz, "Knowledge engineering for large ontologies with sigma kee 3.0," in *Automated Reasoning*, S. Demri, D. Kapur, and C. Weidenbach, Eds. Cham: Springer International Publishing, 2014, pp. 519–525.

[6] L. Kovács and A. Voronkov, "First-order theorem proving and vampire," in *Proceedings of the 25th International Conference on Computer Aided Verification*, ser. CAV 2013, vol. 8044. New York, NY, USA: Springer-Verlag New York, Inc., 2013, pp. 1–35.

[7] S. Schulz, "E – A Brainiac Theorem Prover," *Journal of AI Communications*, vol. 15, no. 2/3, pp. 111–126, 2002.

[8] C. Benzmüller and A. Pease, "Progress in automating higher-order ontology reasoning," in *Workshop on Practical Aspects of Automated Reasoning (PAAR-2010)*, B. Konev, R. Schmidt, and S. Schulz, Eds. Edinburgh, UK: CEUR Workshop Proceedings, 2010.

[9] A. Pease and S. Schulz, "Contradiction detection and repair in a large theory," in *The International FLAIRS Conference Proceedings*, 2022.

[10] A. Pease and C. Benzmüller, "Ontology Archaeology: Mining a Decade of Effort on the Suggested Upper Merged Ontology," *The ECAI-10 Workshop on Automated Reasoning about Context and Ontology Evolution*, August 2010.

[11] M. T. Ozsu and L. Liu, Eds., *Encyclopedia of Database Systems*. New York: Springer, 2009.

[12] A. Lan-xuan, Z. Xin-yu, D. Zhi-guo, and W. Cheng-bo, "Research on ontology-based situation understanding and decision-making approach for mass," in *IOP Conference Series: Materials Science and Engineering*, vol. 929, 2020, p. 012029.

[13] D. Sumić, L. Maleš, T. Marušić, and M. Rosić, "Ontology-based information infrastructure for autonomous ships," *Transactions on Maritime Science*, vol. 13, no. 1, 2024.

[14] R. Provine, C. Schlenoff, S. Balakirsky, S. Smith, and M. Uschold, "Ontology-based methods for enhancing autonomous vehicle path planning," *Robotics and Autonomous Systems*, vol. 49, no. 1, pp. 123–133, 2004, knowledge Engineering and Ontologies for Autonomous Systems 2004 AAAI Spring Symposium. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889004001423

[15] P. Hatlas-Sowinska and M. Wielgosz, "Ontology-based approach in solving collision situations at sea," *Ocean Engineering*, vol. 260, p. 111941, 2022.

[16] A. Pease, *The Sigma Manual: A Guide for Users and Developers of the SUMO Toolset*, 2024, accessed: January 14, 2025. [Online]. Available: https://github.com/ontologyportal/sigmakee/blob/master/doc/manual/SigmaManual.pdf

[17] C. Fellbaum, "Semantic network of english verbs," in *WordNet: An Electronic Lexical Database*, ser. Language, Speech, and Communication, C. Fellbaum, Ed. MIT Press, 1998, pp. 69–104. [Online]. Available: http://books.google.com.hk/books?id=Rehu8OOzMIMC

[18] A. Pease, "Converting the suggested upper merged ontology to typed first-order form," 2023. [Online]. Available: https://arxiv.org/abs/2303.04148

[19] C. Benzmüller and A. Pease, "Higher-Order Aspects and Context in SUMO," in *Special issue on Reasoning with context in the Semantic Web*, I. J. V. Jos Lehmann and A. Bundy, Eds. Science, Services and Agents on the World Wide Web, 2012, vol. 12-13.

[20] G. Sutcliffe and M. Desharnais, "The CADE-29 Automated Theorem Proving System Competition (CASC-29)," *AI Commun.*, vol. 37, no. 4, p. 485–503, Jan. 2024. [Online]. Available: https://doi.org/10.3233/AIC-230325

[21] V. D. Team, "Vampire: Theorem prover," https://vprover.github.io/, 2025, accessed: 2025-01-03.

[22] N. Bowditch, *American Practical Navigator: An Epitome of Navigation*, 2019th ed. Springfield, Virginia, USA: National Geospatial-Intelligence Agency, 2019, vol. II, Pub. No. 9, Prepared and

published by the National Geospatial-Intelligence Agency. [Online]. Available: https://msi.nga.mil/Publications/APN
[23] G. Sutcliffe, “The TPTP Language,” 2025, accessed: January 14, 2025. [Online]. Available: https://tptp.org/UserDocs/TPTPLanguage/TPTPLanguage.shtml