TRANSNAV
http://www.transnav.eu
the International Journal
on Marine Navigation
and Safety of Sea Transportation
Volume 7
Number 2
June 2013

DOI: 10.12716/1001.07.02.01

Towards Safe Navigation by Formalizing Navigation Rules

A. Kreutzmann, D. Wolter, F. Dylla & J.H. Lee
Cognitive Systems Group, University of Bremen, Bremen, Germany

ABSTRACT: One crucial aspect of safe navigation is to obey all navigation regulations applicable, in particular the collision regulations issued by the International Maritime Organization (IMO Colregs). Therefore, decision support systems for navigation need to respect Colregs and this feature should be verifiably correct. We tackle compliancy of navigation regulations from a perspective of software verification. One common approach is to use formal logic, but it requires to bridge a wide gap between navigation concepts and simple logic. We introduce a novel domain specification language based on a spatio-temporal logic that allows us to overcome this gap. We are able to capture complex navigation concepts in an easily comprehensible representation that can directly be utilized by various bridge systems and that allows for software verification.

1 INTRODUCTION

Navigation regulations such as the official collision regulations of the International Maritime Organization (IMO Colregs) are an essential instrument for safety in navigation. Some situations may require further rules and general recommendations may be implemented to foster sensible navigation behavior, e.g., with respect to fuel efficiency. All these regulations need to be obeyed—which can be a very demanding task in complex situations. By augmenting bridge systems such as ECDIS and autopilots to understand navigation regulations we can support crews, reducing the risk of regulation violations. To start with, this requires an implementation of navigation regulations that is known to be correct.

We argue for a declarative, logic-based approach to represent navigation regulations. Logics offer precise semantics for reasoning and they build a common basis for software verification. The use of such formal methods during software development is a common requirement for higher standards of safety-critical software. However, logics are usually based on primitive concepts and it requires overly complex statements to represent everyday concepts such as "oncoming traffic". Trying to formalize a non-trivial set of navigation regulations with a simple logic inevitably leads to incomprehensible formalizations that are error-prone to align with navigation software, rendering effectiveness of the overall approach questionable.

The contribution of this paper is to show how the opposition of primitive concepts in logic on the one-hand and abstract concepts in navigation regulations on the other hand can be overcome. To this end, we develop an abstract logic, a so-called qualitative spatio-temporal logic, which can adequately represent navigation concepts. They allow comprehensible representations specifically suited for navigation problems. Qualitative spatial logics as studied in the field of Artificial Intelligence (AI) are acknowledged

for their ability to grasp concepts of human cognition. We thus can connect formal logic to concepts of human cognition, obtaining formalizations with precise logic semantics that can be understood and even adapted by navigators, not only by computer science experts. These formalizations are universal in the sense that the very same representation can be used in a variety of tasks: to display regulation violations in ECDIS, to enforce rule-compliant path-planning in autopilots, and above all to support the software development process by verification.

This paper is organized as follows. We give references to related work, then we present our qualitative spatial logic. We outline how navigation formalizations in this logic can be integrated with various navigation tasks using logic-based software tools. Finally, we show how logic reasoning developed for our logic can be employed in verification and to reveal problems with software or with the regulations themselves. The paper concludes by a discussion and outlook section.

2 BACKGROUND

Sophisticated bridge system can be considered as decision support systems (DSS) as they aim to support crew in navigation decisions. Various computer science techniques have been applied to devise such systems. Smierzchalski & Michalewicz (2000) and Szłapczynski (2010) demonstrate how evolutionary algorithms can be applied for collision free navigation even in case of multi-ship encounter. A related approach has been pursued by Mohamed-Seghir (2012) using a combination of branch-and-bound and genetic algorithms. Both approaches aim to determine a cost-optimal path, but they cannot guarantee to respect official regulations, i.e., it can be illegal and even dangerous to follow a path computed. It is thus necessary to integrate a representation of Colregs in order to obtain decision support that complies to official regulations. As reported by Pietrzykowski & Uriasz (2010), various approaches to represent knowledge contained in navigation regulations like Colregs have been applied. Their approach aims at combining different techniques, but does not handle situations in which multiple vessels are mutually subject to regulations at the same time. By contrast, Banas & Breitsprecher (2011) argue for the use of logic rule-based systems as framework for representing navigation regulations. They claim logic to provide the best means to tackle requirements on a DSS for navigation identified, namely reproducible and verifiable results, integration of informal knowledge, easy update or extension of knowledge, regulation prioritization, and comprehensibility of the representation. Indeed, the use of formal methods based on logic is a common means to foster reliability of safety-critical software like a DSS for navigation. We adopt the motivation of Banas & Breitsprecher to employ logic for formalization. Our primary focus is to adequately capture the complex spatio-temporal concepts involved in navigation regulations. We improve on previous work by devising an advanced logic framework that incorporates sophisticated spatial reasoning. This allows us to better meet the aforementioned criteria for bridge systems, in particular with respect to the safety-critical aspect of verifiability of the software and with respect to comprehensibility of the representation.

Developing a formalization of Colregs one has to face several design criteria which are somewhat competing. Any formalization of Colregs has to bridge the gap from the official regulations denoted in natural language to a verified formal framework on which the system is based. A common approach is to develop a domain language which abstracts from the formal framework and offers concepts and techniques close to the application domain. Of course, the mapping from domain language to the formal system must be transparent and verifiable itself to avoid introducing errors in the translation. To this end, our approach utilizes a formal framework that already incorporates many abstract concepts necessary to represent navigation regulations. This allows us to obtain a transparent mapping from domain language to underlying logic. Moreover, bridge systems can benefit from the underlying formal framework, given the framework provides sophisticated reasoning mechanisms that are capable of tackling navigation tasks. As we demonstrate, reasoning methods of qualitative spatio-temporal logics are well-suited to meet this goal.

2.1 *Qualitative spatio-temporal logics*

Qualitative spatial and temporal reasoning is an established field of research dealing with representation and reasoning about spatio-temporal knowledge in an abstracted, i.e. qualitative, manner. Qualitative approaches are symbolic and symbols serve to represent concepts like “left” rather than using numerical values that measure directions. The aim of qualitative approaches is to capture the important distinctions that make a difference for a task at hand while abstracting from irrelevant details.

Qualitative spatial and temporal reasoning provides different methods of reasoning, most notably methods that can decide whether a given symbolic description of a scene is consistent, i.e., whether it can be realized by a physical configuration. For example, the three temporal statements about events A,B, and C, namely “A occurs before B”, “B occurs before C”, and “C occurs before A”, are not jointly realizable as time evolves linearly. Qualitative reasoning provides techniques to reason about various aspect of space and time (Cohn & Renz, 2007) and specialized reasoning tools are available, e.g., SparQ (Wolter & Wallgrün, 2012).

Recently, qualitative approaches have been studied in conjunction with logics, thus coining the term spatio-temporal logic. These logics are formed by “any formal language interpreted over a class of structures featuring geometrical entities or relations” (Aiello et al., 2007, Chapter 1). The logic itself is not restricted, i.e., it may be a fragment of first order logic or any higher-order logic. In this paper we are concerned with a combination of a modal logic of linear time with a qualitative approach to representing directional knowledge presented in Section 3.3.

3 FORMALIZING NAVIGATION REGULATIONS FOR USE IN BRIDGE SYSTEMS

The key question in designing an appropriate formalization is what are the individual components that make up a set of navigation regulations? Since we formalize a safety-critical system we must ensure that these components need a clear linkage to the primitives in the underlying logic.

At the core of a regulation we can identify the navigation behavior. Navigation behaviors come in two flavors. Firstly, we have navigation behaviors as instructed behaviors: the regulation defines which actions are allowed to perform. Secondly, we find navigation behaviors setting the context in which a specific regulation is applicable, for example, with respect to the vessels' relative course. While both flavors share many commonalities, there exist decisive differences. One must ensure that a context description can be evaluated at any point in time to allow instructed behaviors to be performed as soon as a regulation is applicable. If, by contrast, the context would be allowed to refer to the future, one could not tell whether one's current situation matches the context. We say that a context is a discernible navigation behavior, i.e., a pattern of actions and events that can be recognized by an observer. Analogously, instructed behaviors are restricted to only talk about future actions. In other words, regulations are of the form "if you approach the port, reduce speed" rather than "if you crashed into a quay wall, you should have reduced speed in first place". Although context and instructed behavior are distinct, we can apply a common framework of representation to both of them.

As second component of regulations we identify a valuation of liability. As soon as a regulation is applicable, its instructed behavior defines which actions are allowed. As applicability of a regulation is subject to change, we introduce the term valuation of liability to indicate whether a navigation behavior is applicable and how it relates to competing regulations. The Colregs regulations have different liability and their liability might change depending on other regulations currently applicable. For example, the regulations state that (Rule 13,d): "Any subsequent alteration of the bearing between the two [overtaking] vessels shall not make the overtaking vessel a crossing vessel within the meaning of these rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear." In this example, certain behaviors (being a crossing vessel) are temporarily forbidden while vessels are in the context of overtaking one another. While inhibiting certain behaviors can easily be formalized, a true modeling of rule precedence and conflict resolution is a challenging aspect in its own right and outside the scope of this paper. For time being, we simply say that a valuation may take either the value applicable or not applicable.

In summary, a set of navigation regulations can be formalized as a mapping from the set of navigation behaviors describing the context to a valuation of liability of navigation behaviors that state which behaviors are allowed to take place. Our terminology is close to that of rules in the classical sense of logic in computer science: an antecedence leading to consequence.

Throughout the remainder of this paper we use Colregs Rule 12,a,i (sailing vessels) as a running example to illustrate our approach. Let us start by looking at the example of how Rule 12 can be formalized in our approach shown in Figure 1.

official rule (natural language):

When two sailing vessels **are approaching one another**, so as to involve risk of collision, one of them shall keep out of the way of the other as follows:

(i) when each has the **wind on a different side**, the vessel which has the **wind on the port** side shall **keep out of the way** of the other.

formalization (modeling language):

```
(rule12_i
:context  (AND (is_sailing_vessel ?X)
               (is_sailing_vessel ?Y)
               (is_approaching ?X ?Y)
               (is_approaching ?Y ?X)
               (COULD (collide ?X ?Y))
               (wind_on ?X PORT)
               (wind_on ?Y STARBOARD))

:behavior (AND (give_way ?X)
               (keep_course ?Y)))
```

Figure 1. From Colregs (top) to regulation formalization (bottom). The formalization describes context and required behavior in a declarative manner; ?X and ?Y are variables that stand for vessels.

As can be seen, we have chosen a simple syntax using parentheses for grouping. The context and instructed behavior part of a regulation are indicated by respective labels. The formalization only explicitly states one case of having "the wind on a different side" which eases readability, as the other case is symmetrical and achieved by swapping the variables. Observe that the formalization utilizes terms like "is approaching" or "keep course" that are very close to the natural language used in Colregs. At this point it is important to note that these terms are logic concepts which need a clear grounding in spatio-temporal knowledge about the world. Assuming a reasonable interpretation of these terms, the formalization can easily be checked against the official Colregs by any domain expert, e.g., trained helmsman or naval expert. Let us now look into the technical details of how these concepts are grounded in the logic and how logic reasoning can be performed.

3.1 *A spatio-temporal logic for formalizing navigation behaviors*

We give a brief introduction of the modal logic underlying our formalization. Since the key focus of this paper is not discussing the logic itself but to demonstrate its applications as well as the domain dependent language established on top of it, we only introduce the logic informally.

For our approach we developed a so-called multi-modal logic. Like any modal logic, this logic is a

generalization of propositional logic which is equipped with the concept of different *states*, also called *worlds*. Truth of a formula is evaluated with respect to a specific state. For example, the logic primitive "`sailsSet`" may be true in one state, but false in another. All possible states constitute the so-called universe and individual states are connected by specific relations called modals. Typically, a universe is assumed to be given and to be finite (Blackburn et al., 2006, Chapter 1). A universe and a set of modals together with the information about which state of the universe makes which logical primitives true form a *model* of a modal logic.

A prominent example for a modal is time: one state may represent the circumstances at a time point ti and the connected state talks about the next moment in time ti+1. As navigation regulations are grounded in time and space we employ two modals (thus we have a multi-modal logic): one modal captures the course of time and another one captures possible spatial changes. The spatial modal will allow us to talk about possible changes of the states and, e.g., to express the possibility of collision as a logic primitive. Technically speaking, we adopt the relation of conceptual neighborhood defined in qualitative spatial reasoning; two states are conceptually neighbored if one state can be continously changed to another (Dylla, 2009). The model for our logic is thus a set of such states along with their temporal ordering, spatial structure, and valuation of all logic primitives. Essentially, our logic is a spatially enhanced generalization of the well-established Linear Temporal Logic (Pnueli, 1977).

For convenience, we write, e.g., `sailsSet(X)`, to denote the logic primitive holding the truth value that corresponds to whether vessel X has sails set or not. Returning to our previous example (Rule 12 i), it can be written in logic notation as follows:

$$\begin{aligned}
\exists X,Y \in Vessels: \big[& \big(Sailboat(X) \wedge Sailboat(Y) && (1a)\\
& \wedge Approaching(X,Y) \wedge Approaching(X,Y) && (1b)\\
& \wedge \langle cn \rangle \big(Collsion(X,Y)\big) && (1c)\\
& \wedge WindOn(X, port) \wedge WindOn(Y, starboard)\big) && (1d)\\
& \rightarrow \big(GiveWay(X) \wedge KeepCourse(Y)\big)\big] && (1e)
\end{aligned}$$

Note that our logic is already close to the modeling language; so we meet the demand of easy translation from domain language to logic. In this example ⟨cn⟩ in line 1c stands for a conceptual change which can lead into a state where X and Y collide. The instructed behavior (line 1e in the formula) is written as implication of the preconditions 1a–1d. Also note that some spatial relations such as `Approaching` used above are in fact independent formulas themselves as we will explain in the following. As a regulation is applicable to all vessels, the simple logic form "context → instructed behavior" needs to be stated explicitly for all logic primitives representing vessels. This is achieved by combining sub-formulas for any choice of X and Y by the logic conjunction "or". By building a modeling language atop this logic layer we can ensure that all regulation formalizations adhere to this pattern of logic formulas.

3.2 *A domain language for navigation regulations*

In this section we explain how our domain language is build atop the spatio-temporal logic outlined above. We describe how the key notions of context and instructed behavior are expressed and how spatial and temporal knowledge can be represented.

The set of primitive symbols used by the logic is divided, identifying the subset of discernible primitives. Discernible primitives can directly be observed by others (like `sailsSet(X)`, for example) whereas other primitives may not. We employ this distinction such that it can be checked whether a navigation behavior can be recognized by observation: specifications of navigation behaviors allow for recognition if they only involve discernible primitives. The context comprises a set of navigation behaviours. In order to decide whether a context formalization matches a given situation we require the context to only involve discernible behaviours. Moreover, formalization of contexts is restricted to only talk about now, the past, and things possible in future. This can easily be accomplished by restricting the set of modal operators allowed in the formalization. Thus, we inhibit the use of universal-qualified expressions in this part of the formalization. With respect to instructed behavior there is only one requirement: it must not refer to past actions. This is also achieved by disallowing the respective modal operators in the formula. All in all we obtain that all parts of a regulation are logic formulas, each class with a specifically restricted syntax.

$$\exists X,Y : [context] \rightarrow [instructed\ behavior]$$

In summary, our system translates all rules into the pattern as shown in the previous section. The key feature of our approach is its seamless integration with qualitative spatial logics that allows us to define a rich repertoire of spatial relations.

3.3 *Spatio-temporal primitives*

In formalizing Colregs it is essential to formalize the manifold spatio-temporal concepts referenced in the regulations. The key building block of the spatial formalization is a set of qualitative spatial relations that capture directional information as presented in (Wolter et al., 2011). This modeling is a sector-based model presented in (Moratz, 2006) (see Figure 2) which allows us to derive most important spatial concepts. Essentially, the model allows directional sectors to be defined that are aligned with respect to position and orientation of an observer. While the number of sectors can be chosen arbitrarily to accommodate for any desired resolution, we restrict the presentation here due to space constraints to showing only the eight-sector variant. In the example shown in Figure 2 (A), the position of B is in sector 0 with respect to A and vice versa—A and B are thus oriented to one another. Figure 2 (B) shows how the model can be used to describe the wind. The vessel depicted has the wind of port side as the wind comes from sector 2. Analogously, the same model serves to state which is the right side to pass by a buoy, see Figure 2 (C). Here, the white area reresents a waterway.

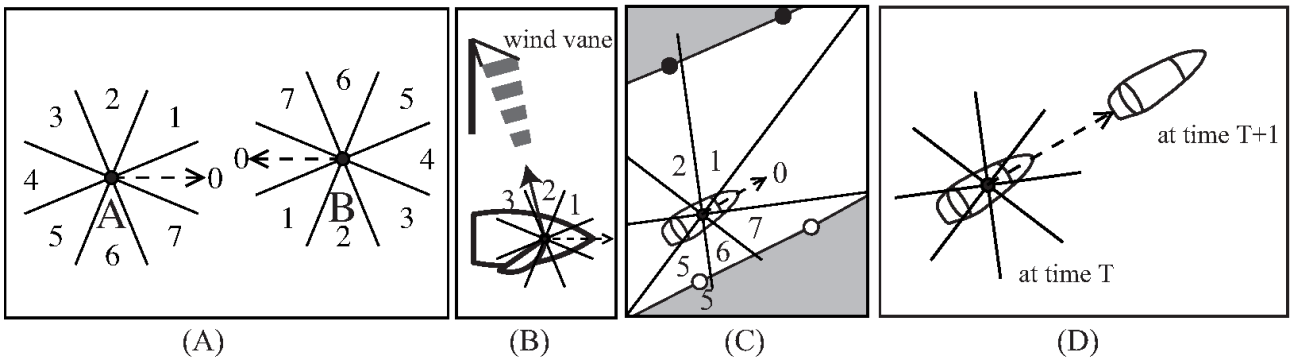

Figure 2. Illustration formalizing the spatial concepts underlying Colregs.

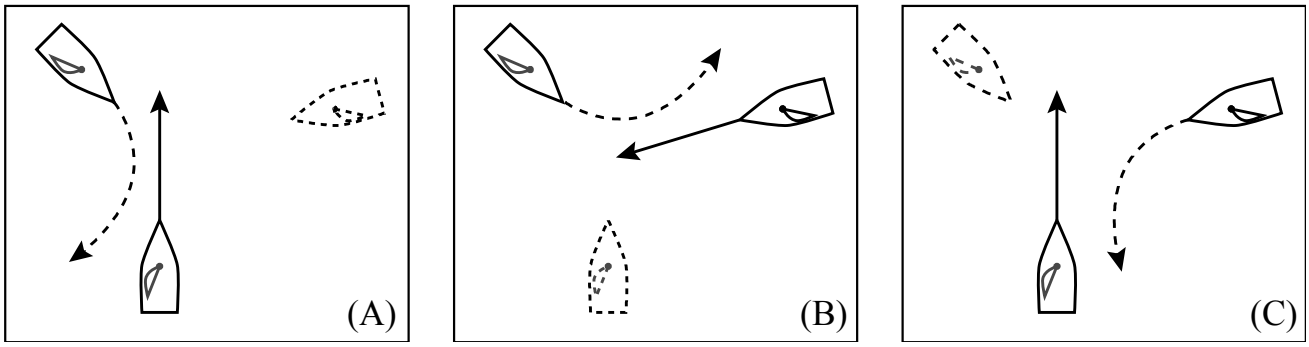

Figure 3. Depiction of collsion avoidance patterns (i.e., interpretations of the rules) for pair of vessels as found in text books on navigation. A dashed line indicates that the vessel has to give way while a solid line means that the vessel should keep course. In the depicted configuration, these collision avoidance patterns are contradicting one another.

Exhausting the expressivity of a temporal logic we can also exploit these spatial relations to define dynamic navigation behavior. For example, the term "head-on course" can be defined by saying that at one time point two vessels are oriented towards one another (see above), while in the next time point they are still oriented the same way but that both have advanced towards one another. A's position at time point t_{n+1} is ahead of where A was at time point t_n, i.e., A at t_{n+1} is within sector 0 as seen from A at time point t_n —see Figure 2 (D) for illustration. It is the modal operators of a temporal logic that grants us the expressivity to relate A's position between different points in time.

3.4 *Model checking with spatio-temporal logics*

Generally speaking, given a model M and a state w in M and a formula ϕ, the task of model checking in modal logic is to determine whether w along with M satisfies the formula ϕ. Specifically in the context of our spatio-temporal logic, model checking is the task of searching for a sequence of spatio-temporal transitions starting with the input state w of vessels which makes regualtion ϕ true with respect to the model M of the spatio-temporal logic described in Section 3.1. By means of the combination of model checking with methods from qualitative reasoning we are able to reason about whether given input states are critical with respect to safety.

The important feature of modal logics is that model checking can be realized efficiently. In our system we utilize the state of the art model checker PRISM (Kwiatkowska et al., 2011) which requires us to provide a set of states to check. PRISM either returns that all states satisfy the given formula or it provides us with a counter-example that falsifies the formula.

In order to generate all possible states in our spatio-temporal logic, qualitative spatial reasoning is required. For example, consider the statement "`WindOn`(X, port) $\wedge$ `WindOn`(X, starboard)" which is of course not satisfiable. However, from the perspective of a pure modal logic model checker the formula is just the same as "a $\wedge$ b" and thus there is no reason why a and b should not hold at the same time. This is where spatio-temporal reasoning is required to rule out configurations which are spatially or temporally not possible. To this end, we combine our spatio-temporal reasoning system SparQ (Wolter & Wallgrün, 2010) to check all candidates of states for their spatial and temporal consistency.

In the following section we show how various practical problems can be supported with the two reasoning tasks on the logic level only: model checking of formulas in our spatio-temporal logic (PRISM) and consistency checking of qualitative spatial configurations (SparQ).

4 REASONING FOR SAFE NAVIGATION

We now demonstrate how a formalization of navigation regulations serves three major applications in bridge systems: recognition of regulation compliant/violating behavior, regulation compliant planning, and verification of regulation specifications.

4.1 *Identifying regulation compliancy and regulation violations*

Observing the navigation behavior around one's own position, a natural question to ask is: do all other vessels comply with the regulations or is some vessel violating a regulation? In the domain of safe sea navigation a system assisting in the detection of regulation violating behavior of others can be of great importance to a bridge crew. By alerting the bridge crew of such violations appropriate preparations can be made. Assuming that observations of the surrounding navigation behavior are available (e.g., extracted from AIS data or radar), we apply model checking of the Colregs formalization to the observations as follows: From the formalization of Colregs we have the corresponding logic formulas $\phi = \phi_1, \phi_2, ..., \phi_m$ which we combine by logic conjunction "and": $\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_m$. Doing so we obtain a single formula ϕ that represents the complete body of navigation regulations. Observed behavior constitutes the set of states, each snapshot of time defines its own valuation of logic primitives. For example, at time point t_n we have that `Approaching`(X, Y), while at the next time point t_{n+1} we already have `TurningAwayFrom`(X, Y) and it holds that `TurningAwayFrom`(X, Y)$\rightarrow\neg$`Approaching`(X, Y). If and only if the set of states obtained from observations provides a model for the logic formula ϕ, the observed behavior is compliant with Colregs. If model checking fails, a counter-example is generated. This counter-example falsifies ϕ and identifies which vessels/actions did not comply with the regulations.

We demonstrated feasibility of this method in a previous study using a different domain, showing that the approach works even on noisy and incomplete sensor data (Kreutzmann et al., 2012). A small scale warehouse was simulated in a laboratory and a robot observed the warehouse, gathering partial observations. The observations were matched against a set of logistic movement patterns—which can be considered as a form of navigation regulations. Based on these partial observations, a matching between real-world observations and abstract model could be established, i.e., compliancy with some logistic rules was verified.

4.2 *Regulation compliant planning*

In complex situations such as crowed waterways like the English Channel, planning routes that are regulation compliant but avoid detours can be a demanding task. An autopilot system could factor in routes of other vessels and inhibit the planning procedure to output routes that are known or likely to violate regulations. This enables the vessel to avoid unnecessary evasive maneuvers and thereby save fuel. Kolendo et al. (2011) have demonstrated that randomized planners are appropriate for computing collision-free routes. In (Wolter et al., 2011) we demonstrated how this approach can be advanced to ensure regulation compliant planning. We extend the state-of-the-art paradigm of randomized roadmap planners to acknowledge navigation regulations as side constraint in planning. This also augments existing work on randomized planning by a verification component.

In essence, the approach is similar to that of recognizing regulation-violating navigation behavior discussed above. While planning takes place, all partial plans considered by the planner are checked for their regulation-compliancy using the very same technique as explained before. Whenever a partial plan is identified to violate some regulation, it is discarded and the planner has to search for an alternative route.

4.3 *Verification of regulation specifications*

When developing safety-critical software all development steps should be verifiable. Moreover, software developers need to be supported to identify problems with their software. To this end, we are currently developing a tool to support the verification of navigation software based on the formalization of navigation regulations. Following the concept of Proof-Carrying Code (Necula, 1997), developers can declare complex assertions in their software. For example, the command to increase engine speed may be guarded by an assertion that no obstacle is in front of the vessel.

Such a tool is particularly valuable when working with Colregs-compliant navigation. Regulations like the Colregs have mostly been developed with respect to defining right of way for just two vessels at a time. Thus, in complex situation a multitude of regulations may be applicable at the same time. Regulations might even contradict themselves or endanger a collision if strictly followed. We can support a knowledge engineer by providing a set of useful verification methods. Given a set of regulations, our tool provides the necessary means to check that regulations are non-overlapping, i.e., there are no situations in which different regulations are contradicting one another. See Figure 4 for a screenshot of the tool checking regulations for problems. The tool also allows us to check whether all (critical) situations are covered by some rule, i.e., whether a set of rules is complete. One important feature of the tool is that it does not only identify conflicts of regulations, but it can also generate an exemplary configuration that triggers the conflict. Here we use spatial reasoning to generate a prototypical pictogram that depicts the conflict (see Figure 4, top right). The position of the boom indicates direction of the local wind. Conflicting regulations are particularly present if an autonomous navigation systems implements its own regulations for special maneuvers, which might interfere with Colregs. Consider the commonly recommended evading actions in the form of collision avoidance patterns as presented in text books for navigation (Dreyer, 2012, e.g.) as shown in Figure 3. In the situation depicted in Figure 3, the vessel on the right has the conflict of keeping course and give way at the same time. Further the top left vessel should turn port and starboard at the same time. While each collision avoidance pattern on its own is sensible, taking into account the specific requirements such as wind direction, they fail to combine in some situations, e.g., multi-ship encounters. Our tool is able to detect this inconsistency solely by reasoning about the rule definitions as shown in Figure 4.

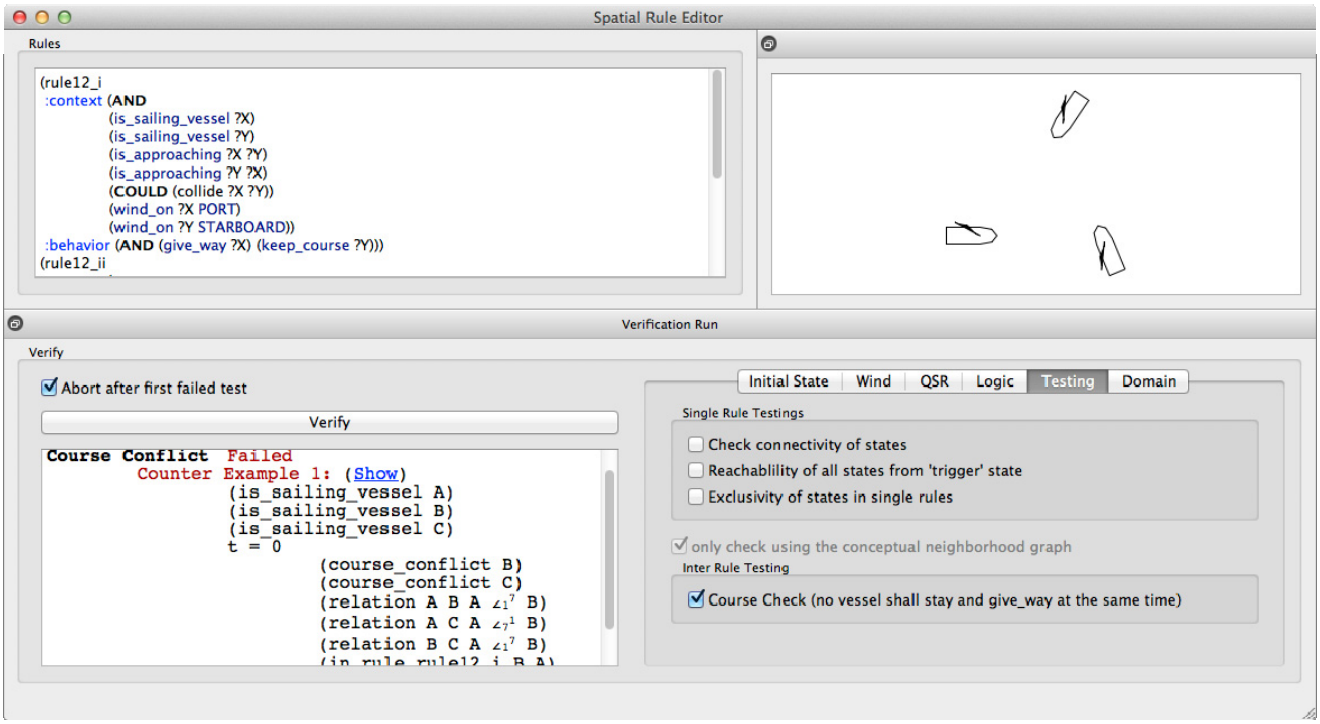

Figure 4. Screen shot of the reasoning tool that automatically detects a rule conflict and generates a visualization of the conflicting situation.

5 SUMMARY AND OUTLOOK

Adhering to navigation regulations is an important factor in safe navigation. In order to allow bridge systems to incorporate regulations such as Colregs in a verifiably correct manner, a formalization of navigation regulations is required. In this paper we show that a spatio-temporal logic provides a solid basis for formalizing Colregs. We propose an easy to understand domain language built atop a spatio-temporal logic. Logic reasoning enables automated tools to check formalizations for correctness. This enables software developers for bridge systems to verify their software, meeting high standards of safe software design. Our approach is applicable to a wide range of navigation regulations and can even help to develop new ones (see, for example Kemp, 2009). Thereby, we can provide an answer to a longstanding problem that "there is no possibility of testing new proposals [for Colregs] before they are introduced." (Kemp, 2007). Moreover, a formalization of Colregs in our spatio-temporal logic can also be utilized in navigation systems directly: autopilots can determine routes that are known to comply with Colregs or chart displays can identify violations of Colregs and signal appropriate warnings. We develop a software tool based on the concept of proof-carrying code that enables software developers to verify their software with respect to a formalization of Colregs. In future work, we aim to apply this tool to real navigation software in order to improve safety in navigation technology.

ACKNOWLEDGEMENTS

This work is carried out in context of the Transregional Collaborative Research Center Spatial Cognition, project R3-[Q-Shape]. Financial support by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged.

REFERENCES

Aiello, M., Pratt-Hartmann, I. & van Benthem, J. (ed.) 2007. *Handbook of Spatial Logics*. Berlin: Springer.

Banas, P. & Breitsprecher, M. 2011. Knowledge base in the interpretation process of the collision regulations at sea. *TransNav—International Journal on Marine Navigation and Safety of Sea Transportation*, 5(3):359–364, Gdynia: Poland.

Blackburn, P., van Benthem, J. & Wolter, F. (ed.) 2006. *Handbook of Modal Logic*, New York: Elsevier.

Cohn, A. & Renz, J. 2007. Qualitative Spatial Representation and Reasoning. In van Harmelen, F., Lifschitz, V. & Porter, B. (ed.), *Handbook of Knowledge Representation*, 551–596, New York: Elsevier.

Dreyer, R. 2012. *Sportküstenschifferschein & Sportbootführerschein See* (in German). Bielefeld: Delius Klasing.

Dylla, F. 2009. Qualitative Spatial Reasoning for Navigating Agents, In Gottfried, B. & Aghajan, H. (ed.), *Behaviour Montoring and Interpretation—Ambient Assisted Living*. Amsterdam: IOS Press.

Kemp, J.F. 2007. The Colregs and the Princess Alice. *TransNav—International Journal on Marine Navigation and Safety of Sea Transportation*, 1(1):57–61

Kemp, J.F. 2009. Behaviour Patterns in Crossing Situations. *TransNav—International Journal on Marine Navigation and Safety of Sea Transportation*, 3(1):75–79

Kreutzmann, A., Colonius, I., Wolter, D., Dylla, F., Frommberger, L. & Freksa, C. 2012. Temporal logic for process specification and recognition. *Intelligent Service Robotics*. 1–14.

Kolendo P., Smierzchalski R., Jaworski B. 2011. Experimental Research on Evolutionary Path Planning Algorithm with Fitness Function Scaling for Collision Scenarios. *TransNav—International Journal on Marine Navigation and Safety of Sea Transportation*, 5(4):489–495

Kwiatkowska, M., Norman, G. & Parker, D. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In Gopalakrishnan, G. and Qadeer, S. (ed.), *Proceedings of 23rd International Conference on Computer Aided Verification*, Berlin: Springer.

Mohamed-Seghir, M. (2012). The branch-and-bound method and genetic algorithm in avoidance of ships collisions in fuzzy environment. *Polish Maritime Research*, 19(S1):45–49.

Moratz, R. (2006). Representing relative direction as a binary relation of oriented points. In Brewka, G., Coradeschi, S., Perini, A. & Traverso, P. (ed.), *European Conference on AI 2006*. Amsterdam: IOS Press.

Necula, G. C. (1997). Proof-carrying code. In P. Lee, F. Hengelein & N.D. Jones (ed.), *Proceedings of the 24th ACM SIGPLAN—SIGACT Symposium on Principles of Programming Languages, POPL '97*. New York: ACM.

Pietrzykowski, Z. & Uriasz, J. 2010. Knowledge representation in a ship's navigational decision support system. TransNav—International Journal on Marine Navigation and Safety of Sea Transportation, 4(3):359–364. Gdynia: Poland.

Pnueli, A. 1977. The temporal logic of programs. In Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), 46–57. Los Alamitos: IEEE Computer Society.

Smierzchalski, R. & Michalewicz, Z. (2000). Modeling of ship trajectory in collision situations by an evolutionary algorithm. In Trans. Evol. Comp, 4(3):227–241.

Szłapczynski, R. 2010. Evolutionary Sets of Cooperating Tra-jectories in Multi-Ship Encounter Situations—Use Cases. TransNav—International Journal on Marine Navigation and Safety of Sea Transportation, 4(2):191–196, Gdynia: Poland

Wolter, D., Kreutzmann, A. & Dylla, F. (2011). Rule-Compliant Navigation With Qualitative Spatial Reasoning, In Schlaefer, A., Blaurock, O. (ed.), *Robotic Sailing—Proceedings of the 4th International Robotic Sailing Conference*, 141–155, Berlin: Springer.

Wolter, D. & Wallgrün, J. (2012). Qualitative Spatial Reasoning for Applications: New Challenges and the SparQ Toolbox. In Hazarika, S. (ed.), *Qualitative Spatio-Temporal Representationand Reasoning—Trends and Future Directions*, 336–362. Hershey: IGI Global.