Journal of Navigation and Port Research International Edition
Vol.32, No.7 pp. 529~535, 2008 (ISSN-1598-5725)

A Study on Development of Expert System for Collision Avoidance and Navigation(I): Basic Design

† *Tae-Gwoen Jeong* · *Chao Chen**

† *Division of Navigation System Engineering, Korea Maritime University, Pusan 606-791, Republic. of Korea*
**Division of Ship Operation Systems Engineering, Korea Maritime University, Pusan 606-791, Republic. of Korea*

Abstract : *As a method to reduce collision accidents of ships at sea, this paper suggests an expert system for collision avoidance and navigation (hereafter "ESCAN"). The ESCAN is designed and developed by using the theory and technology of expert system and based on the information provided by AIS and RADAR/ARPA system. In this paper the ESCAN is composed of four(4) components; Facts/Data Base in charge of preserving data from navigational equipment, Knowledge Base storing production rules of the ESCAN, Inference Engine deciding which rules are satisfied by facts or objects, User System Interface for communication between users and ESCAN. The ESCAN has the function of real-time analysis and judgment of various encountering situations between own ship and targets, and is to provide navigators with appropriate plans of collision avoidance and additional advice and recommendation. This paper, as a basic study, is to introduce the basic design and function of ESCAN.*

Key words : *Expert system, Navigation expertise, Knowledge base, Inference engine, Collision avoidance, Pattern match algorithm*

1. Introduction

Along with large-sized and high-speed trends in ships, serious collision accidents between ships happened again and again. According to the recent report of Korean Maritime Safety Tribunal (KMST), there were 1724 accidents from 2002 to 2006 in Korean coastal waters. The collision accidents are 1082 cases and occupy 62.76% of all. And in the 1082 collisions, the ones caused by human error are 880 cases and occupy 81.33%(KMST, 2007). From these figures we can consider that if human error can be dealt with well, the majority of these collision accidents could have been avoided.

As a method to reduce these accidents, we proposed an Expert System for Collision Avoidance and Navigation (ESCAN). According to information of other ships from AIS and RADAR/ARPA, encountering situations between own ship and other ships can be analysed and assessed in real time by ESCAN. Moreover, ESCAN can provide reasonable decision in collision avoidance by using the knowledge of related fields.

We, thus, believe that the ESCAN can give navigators useful help for dealing with any collision situation and provide good evidence of it. As a basic study, the basic design and construction concept of ESCAN which is implemented by CLIPS(Giarratano, 1998) and Visual C++, are described in this paper.

2. Design of ESCAN

2.1 Integrated Structure of ESCAN

As a data-driven program, ESCAN should receive data from other navigational equipment including AIS(Automatic Identification System) receiver as shown in Fig.1

In ESCAN, some data, such as position information of a target from AIS receiver, only need to be extracted from character strings from equipment. On the other hand, other information need to be calculated, for example, approaching time of a target. All these data are important because they are used for premises and conditions in drawing inferences.

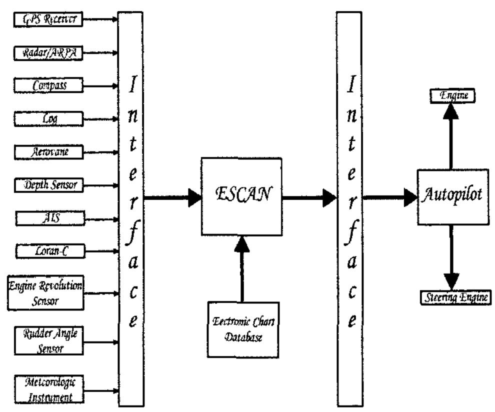

Fig. 1 Connection Diagram of ESCAN with Navigational Equipment and Autopilot System

† Corresponding Author : Tae-Gwoen Jeong, tgjeong@hhu.ac.kr 051)410-4246
* cc20202@163.com 051)410-4856

As shown in Fig. 2, the structure of ESCAN consists of four parts such as Facts/Data Base, Knowledge Base, Inference Engine and User System Interface.

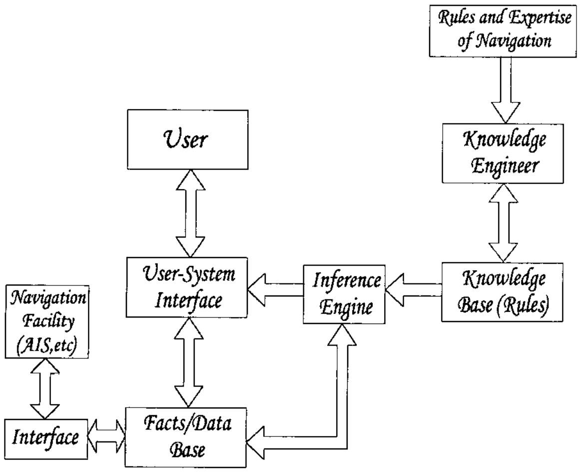

Fig. 2 Structure of ESCAN

Facts/Data Base (hereafter "FB") is in charge of preserving data received from other navigational equipment. Before the data are preserved into FB, it should be transformed into facts according to the format of CLIPS. It is a global database of facts used by the rules. Facts of ESCAN include several types such as fact of own ship, fact of target, fact of visibility and so on.

Knowledge Base (hereafter "KB") is to store production rules of ESCAN. Building KB of ESCAN is to use a proper method to represent navigation expertise. Ability of an expert system lies upon quantity and quality of the knowledge in the KB of it.

Inference Engine (hereafter "IE") is a mechanism which controls operation of ESCAN. IE draws inferences by deciding which rules are satisfied by facts, prioritizes the satisfied rules, and executes the rule with the highest priority.

User System Interface (hereafter "USI") is for communication between user and ESCAN. USI can convert the commands or data input by users into the format of CLIPS, and clearly display results of inferences. User means ship pilots, shipmasters, and other navigators who use ESCAN for navigation.

Navigational expertise and rules are collected, abstracted, induced and processed into the CLIPS-format production rules, which are stored in KB. Data from navigational equipment or input by users are converted into facts which are preserved in FB. IE can judge which rules are corresponding to the existing facts and can also decide which rules will be executed at an appropriate time.

2.2 Facts/Data Base

In ESCAN, facts are data for inference. Other ships and own ship are the most important types of facts. And the facts of the two types contain navigational information on current encountering situations between targets and own ship.

In CLIPS, the definition template of facts called 'deftemplate' is used to define these facts. Using deftemplate, related fields of facts can be defined and restricted. An example of the fact of target is as follows:

(Target (target-name "Berry")(range 2.0)(bearing 3.0)
(course 182.0)(speed 5.0)(DCPA 0.05485)
(TA 0.28582)(CR 3.4913))

Here 'target-name' is the name or symbol of a target, 'range' distance from the target, 'course' COG(Course Over Ground) of the target,'speed' SOG Speed Over Ground) of the targe. These four items can all be acquired from AIS signal. 'bearing' is true bearing of the target, 'DCPA' distance to CPA of the target, 'TA' approaching time of the target to own ship, and 'CR' value of collision risk. The latter four items should be calculated for acquiring. In ESCAN, Equation (1) is used to calculate value of the collision risk(Jeong et al, 2007; Jeong, 2003).

$$CR = \frac{p \cdot \operatorname{sech}(a \cdot dcpa)}{ta} + r \cdot \Phi(\theta, \alpha) \qquad (1)$$

In ESCAN, facts can be added and modified by using data from AIS receiver, RADAR/ARPA or users.

2.3 Knowledge Base

The navigational knowledge used in ESCAN is based on COLREGS and other navigation expertise materials including Jeong(1997)'s study. In ESCAN, the sources of knowledge are as follows:

(1) Elementary knowledge of collision avoidance acquired from COLREGS..
(2) Heuristic knowledge acquired from navigation expertise.
(3) Explanations and comments of COLREGS made by experienced shipmasters, navigators and other experts.
(4) Explanations of collision cases made by navigation experts and maritime tribunals.
(5) Experience of good seamanship.
(6) Sophisticated avoiding actions of typical multi-target encountering scenarios in every material.
(7) Statistic and analyse results of collision cases and avoiding behaviors of navigators.

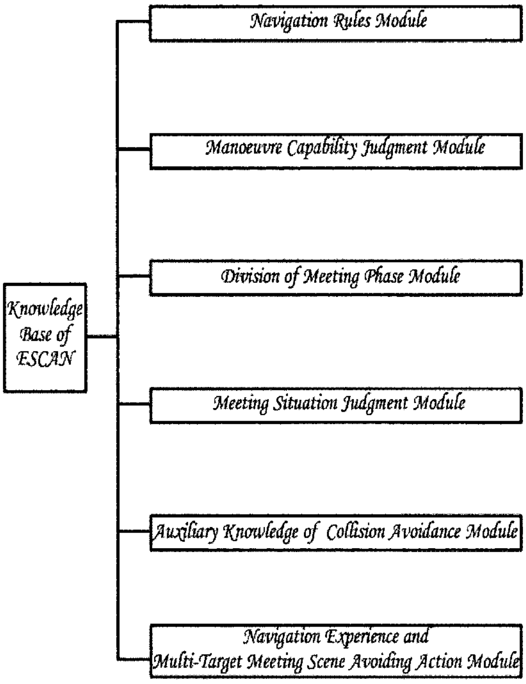

Fig. 3 Module Structure of Knowledge Base

In ESCAN, KB is designed into module structure. As shown in Fig.3, KB of ESCAN is divided into six modules, and every module contains several related aspects.

Collision avoidance is very complicated. So during the process of decision-making of collision avoidance, in addition to principles of navigation (Navigation Rules Module), manoeuvre capability of own ship and manoeuvre capabilities of targets (Manoeuvre Capability Judgment Module), encountering phase(Division of Encountering Phase Module), encountering situation(Encountering Situation Judgment Module) and timing to take avoiding action(Auxiliary Knowledge of Collision Avoidance Module) also need to be considered.

Module structure of KB is favorable not only for improving inference efficiency, but also for organizing and managing the whole system of ESCAN. The whole inference process is controled by IE. When knowledge of one module is needed, IE selects the module containing related rules for inference.

In ESCAN, production rule is used to represent knowledge of collision avoidance. A production rule is composed of an antecedent and a consequent. And a production rule can be described as follows:

IF [P] THEN [Q] or P → Q

Here, P is the antecedent of a rule and is a set of conditions which must be satisfied for the rule to be applicable. In CLIPS, the conditions of a rule are satisfied based on the existence or non-existence of specified facts in the fact-list. And Q is the consequent of a rule and is the set of actions to be executed when the rule is applicable. The actions of applicable rules are executed when IE is instructed to begin execution of applicable rules.

If manoeuvre capability is not considered, production rules of single-target collision avoidance can be simply described as follows:

(1) Nine kinds of predicates are defined to describe the antecedent of a production rule as follows:

<P> = {P1, P2, P3, P4, P5, P6, P7, P8, P9}

Here, P1 is the encountering situation, P2 the bearing of a target, P3 the aspect of a target, P4 the distance between own ship and a target, P5 the manoeuvring state of a ship, P6 the navigation sea area, P7 the traffic situation near own ship, P8 the speed ratio of speed of a target to speed of own ship, and P9 the visibility.

And two kinds of actions are defined to describe the consequent of a production rule as follows:

<Q> = {Q1, Q2}

Here, Q1 is a set of basic avoiding actions such as course alteration to port side. Q2 is a set of heuristic avoiding actions. For example, own ship is not able to alter her course to starboard because other ships or obstacles are on that side of own ship.

Knowledge of single-target collision avoidance can be represented by above-mentioned predicates and actions. For example, head-on situation can be described by 2 heuristic rules. And they are *'If the distance between own ship and a target is less than 6 miles. And in the course of time, the target's average bearing* $\theta \in [0°,6°]$*, its aspect* $\alpha \in [-6°,0°]$*, then own ship forms head-on situation with the target.'* and *'If conditions permit, own ship should alter her course towards starboard when she forms head-on situation with a target'.* And then the two rules can be described by two production rules of P and Q respectively as follows:

IF (P2 = < Bearing_Sector_1 > AND
P3 = < Aspect_Sector_1 > AND
P4 = < Distance_Level_6 >)
THEN (P1= < Head_On_DeadAhead_Starboard > AND
P5 = < Give_Way_State >)

IF (P1= < Head_On_DeadAhead_Starboard > AND
P5 = < Give_Way_State >
P6 = < Wide_Sea_Area > AND
P7= < Traffic_Excellent > AND
P8 = < K_Level_2 > AND
P9 = < Visibility_Good >)
THEN (Q1 = < Turn_Starboard > AND
Q2 = <None>)

Obviously, practical decision-making of collision avoidance is far more complicated than the example, especially when own ship encounters multiple targets. In ESCAN, some rules propose possible avoiding actions. some other rules veto, limit and modify these proposals so as to give more reasonable decision. So final decision of collision avoidance is determined by rules in the different six modules all together.

2.4 Inference Engine

Inference is the act or process of deriving logical conclusions from premises known or assumed to be true. In ESCAN, inference is implemented by inference engine. And IE can infer the most appropriate plan of collision avoidance by using the rules in KB and facts in FB.

1) Inference Process of ESCAN

In ESCAN, a new inference process of collision avoidance is introduced. As shown in Fig. 4, the inference process (hereafter "IP") can be explained as follows:

In this paper, the part of ESCAN programmed in VC++ is called 'Outer S/W' and that in CLIPS is called 'Inner S/W'.

Step 1: Outer S/W will check interface circuit for acquiring data from AIS receiver or RADAR/ARPA and judge whether a target exists or not. If no target exists, then own ship keeps ordinary navigation. Otherwise, Outer S/W informs Inner S/W and then IP enters Step 2.

Step 2: Inner S/W judges the encountering situation between own ship and the target, then IP enters Step 3.

Step 3: Inner S/W judges whether the collision risk between the target and own ship exists or not according to the result of Step 2. If collision risk does not exist, then own ship keeps ordinary navigation. Otherwise, IP enters Step 4.

Step 4: Inner S/W judges whether the current encountering situation is a multi-target encountering situation or not. If it is not, then IP enters Step 5. Otherwise, IP enters Step 6.

Step 5: Inner S/W judges whether own ship is a give-way ship or not. If own ship is, IP enters Step 7. Otherwise, own ship keeps her course and speed. Then Inner S/W checks whether the action of the give-way ship is proper or not. If it is proper, own ship continues keeping her course and speed. Otherwise, IP also enters Step 7.

Step 6: Inner S/W chooses the most dangerous ship as the primary avoiding target of own ship. Then IP enters Step 7.

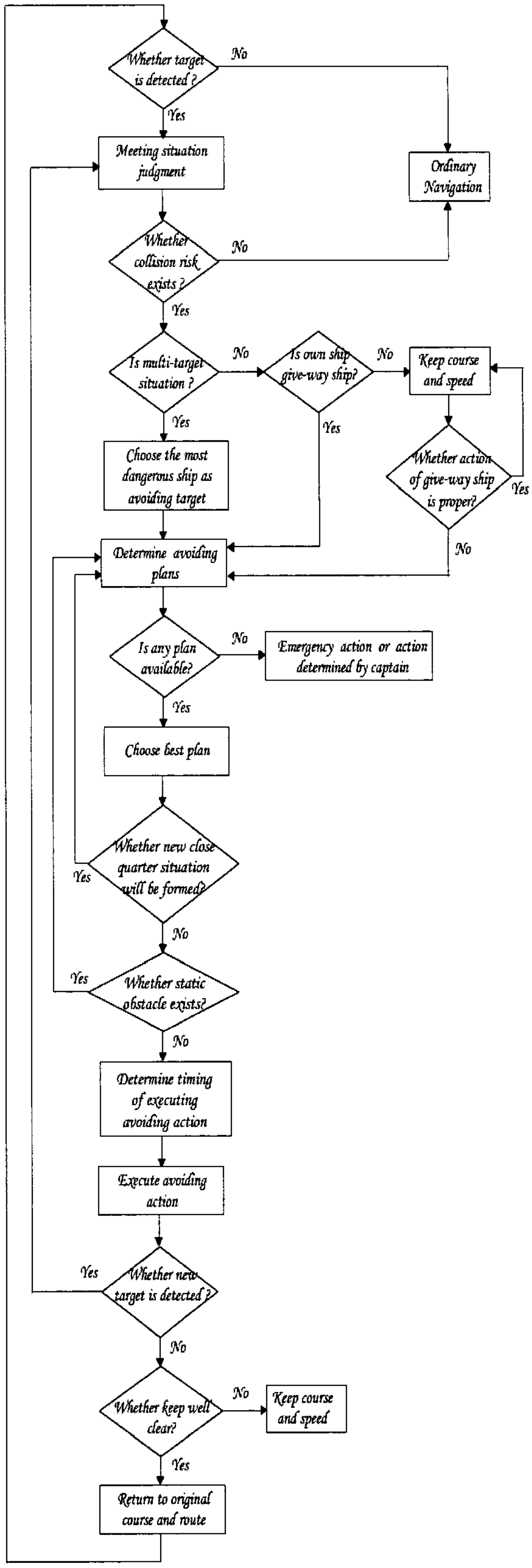

Fig. 4 Inference Flow Chart of ESCAN

Step 7: Inner S/W makes avoiding action plans and informs Outer S/W to display these plans on USI. IP enters Step 8.

Step 8: Inner S/W judges whether there is an action plan available or not. If no plan is available, Inner S/W informs Outer S/W to warn users. Own ship takes emergency action or the action decided by captain. Otherwise, IP enters Step 9.

Step 9: Inner S/W chooses a most appropriate plan and informs Outer S/W to display the plan on USI. Then IP enters Step 10.

Step 10: Inner S/W judges whether new close quarter situation will be formed or not. If it will be formed, then IP goes back to Step 7. Otherwise IP enters Step 11.

Step 11: Outer S/W detects whether any static obstacle is on the path of action plan. If a static obstacle exists Outer S/W informs Inner S/W, then IP goes back to Step 7. Otherwise, IP enters Step 12.

Step 12: Inner S/W decides the timing to take the avoiding action and inform Outer S/W to display relevant information on USI. Then IP enters Step 13.

Step 13: Execute avoiding action. Inner S/W gives orders of avoiding action to autopilot or informs Outer S/W to display the orders on USI so that they can be manually implemented by users. Then IP enters Step 14.

Step 14: Outer S/W detects whether new target exists or not. If new target exists, Outer S/W informs Inner S/W and IP goes back to Step 2. Otherwise, IP enters Step 15.

Step 15: Inner S/W judges whether own ship has kept well clear off the target. If own ship hasn't, she continues keeping her course and speed. Otherwise, IP enters Step 16.

Step 16: Return to original course and route of own ship. Inner S/W gives relevant orders to autopilot or informs Outer S/W to display the orders so that they can be implemented manually by users. Then IP goes back to Step 1 to continue next inference cycle.

In ESCAN, Inner S/W is in charge of inference and Outer S/W is in charge of receiving data from navigational equipment and users, displaying results of inferences of Inner S/W, and also providing several convenient functions, for example a function which uses graphic display technology can simulate avoiding actions provided by Inner S/W.

2) Inference Method

During the inference process, deduction inference is used. Two types of deduction inference are used in ESCAN. And they are forward chaining and backward chaining inference.

Forward chaining inference is used during the whole process to keep the inference process can be carried out smoothly.

If possible avoiding plans are two or more, they should be evaluated to find out which one is the most appropriate and effective one by using backward chaining inference. Forward inference derives logical conclusions from data. However, backward inference derives evidences from conclusions. It sets off from these possible plans to evidences which support them. Information of each plan such as DCPA, need to be calculated assuming the plan is chosen. By comparing the information and considering environment conditions and other related factors, a most appropriate plan of avoiding action can be chosen.

CLIPS are designed for forward chaining and it can directly implement forward chaining inference. However, backward chaining can be emulated using forward chaining CLIPS rules.

3) Pattern-Matching Algorithm

The individual condition of production rule is called a conditional element or a pattern. The process of matching facts to pattern is called pattern-matching. IE is the mechanism which automatically matches patterns against the current facts and determines which rules are applicable. In an expert system, efficiency of the algorithm of pattern-matching concerns efficiency of the whole system. In order to reach a satisfied pattern-matching efficiency, rule-based language CLIPS uses a very efficient algorithm for matching facts against the patterns in rules to determine which rules have had their conditions satisfied. This algorithm is called the Rete Pattern-Matching Algorithm (Giarratano et al, 2005).

One ordinary method of pattern-matching is to have the IE check each rule to direct the search for facts after each cycle of execution provide a simple and straightforward technique for solving this problem. But the primary disadvantage of such an approach is that it can be very slow. And each cycle of execution may see only a small percentage of facts either added or removed and so only a small percentage of rules are typically affected by the changes in the fact list. Thus, having the rules drive the search for needed facts required a lot of unnecessary computation, since most of the rules are likely to find the same facts in the current cycle as were found in the last cycle. The inefficiency of this method is shown in Fig.5. The grey area represents the changes that have been made to the fact list. Not only facts added by Outer S/W, but facts added or removed by executed rules can cause the changes.

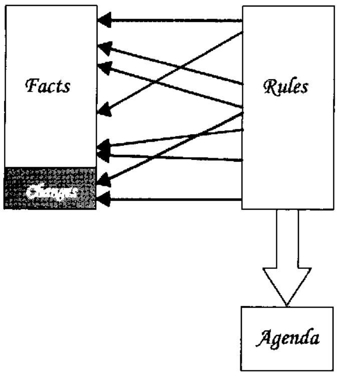

Fig. 5 Rules Search for Facts in Ordinary Algorithm

Unnecessary redundant recomputation could be avoided by remembering what has already been matched from cycle to cycle and then computing only the changes necessary for the newly added or newly removed facts, as shown in Fig.6. The rules remain static and the facts change, so the facts should find the rules, and not the other way around.

The Rete Pattern-Matching Algorithm is designed to save the state of the matching process from cycle to cycle and recomputing the changes in this state only for the changes that occur in the fact list. That is, if a set of patterns finds two of three required facts in one cycle, it is not necessary for a check to be made in the next cycle for the two facts that have already been found and only the third fact is of interest. The state of the matching process is updated only as facts are added and removed. Moreover, the algorithm also improves the efficiency of rule-based systems by taking advantage of structural similarity in the rules.

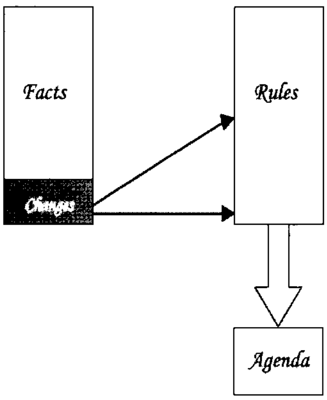

Fig. 6 Facts Searching for Rules in Rete Algorithm

2.5 User System Interface

User system interface is the mechanism for implementing the communication between users and ESCAN. USI of ESCAN is developed by Visual C++.

3 Validation of ESCAN

In this part, effect of ESCAN is validated by an example. It operates in a real time environment. At a time, it detects a single target which its distance from own ship is 7 miles, its true bearing is 003°, its course(COG) is 182° and its speed(SOG) is 5 knots. Meanwhile, course and speed of own ship is 000° and 2 knots respectively.

When distance from the target is less than 6 miles, IE infers that own ship is involved in collision risk and forms head-on situation with the target. So ESCAN provides a plan of avoiding action for dealing with the situation as shown in Fig. 7.

Just as the words shown in circle area of Fig.7, own ship and the target are in head-on situation, the target locates a little bit on starboard side of own ship and its course points to starboard of own ship. ESCAN suggests own ship should alter her course 40° towards starboard and use rudder angle at least 10° during the action. Suggestions of flash signals and timing to return to original course are also provided.

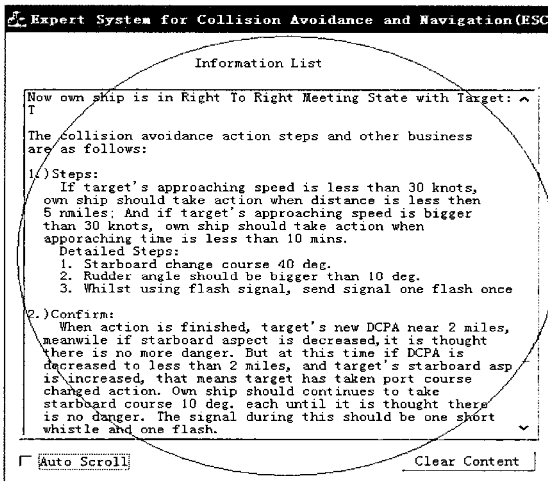

Fig. 7 Suggestion of ESCAN

The related rule of head-on situation in COLREG 1972 is as follows:

'When two power-driven vessels are meeting on reciprocal or nearly reciprocal course so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other'.

According to this rule, when a target and own ship form head-on situation, the two ships should alter course towards starboard. And the suggestion of ESCAN fits the rule quite well. So the inference of ESCAN is reasonable and effective.

Some other validations of different encountering situations are also performed to validate the effect of ESCAN, and the results are satisfactory.

4. Conclusion

Collision avoidance is a difficult topic in the field of navigation of ships at sea. As a method to reduce collisions, ESCAN is introduced in this paper. Some special features of ESCAN are as follows:

(1) Knowledge base of ESCAN is composed of six(6) modules.

(2) A new inference process of collision avoidance as shown in Fig. 4 is used in ESCAN.

(3) Because CLIPS adopts Rete Pattern-Matching Algorithm, response speed of ESCAN is increased greatly.

(4) According to the results of validations, the suggestions provided by ESCAN follow the rules of COLREGS and the advice given by navigation experts well.

(5) Navigators can get more helpful information from ESCAN, and thereby they can know better on traffic situations and can make more resonable decisions of collision avoidance when dangerous situations happen.

(6) It is easy to upgrade ESCAN when rules are required to be upgraded in the future. Only rules in KB should be rewritten rather than the whole system.

In the next study, upgrading rules for dealing with complicated multi-target encountering situations and developing more convenient functions in ESCAN are all needed to be kept researching and studying in the future.

References

[1] Giarratano, J. C.(1998), "CLIPS User's Guide Version 6.24," pp.1~152.

[2] Giarratano J. C. and Riley, G. D. (2005), "Expert Systems -Principles and Programming-," 4th ed., Thomson, pp.477~480.

[3] Jeong, T. G.(1997), "Navigational AI Judgement Rules' Development and Collision Avoidance System's Research," Samsung Heavy Industries, pp.1~86.

[4] Jeong, T. G.(2003), "A New Approach to the Evaluation of Collision Risk using sech Function", Journal of Korean Navigation and Port Research, Vol.27, No.2 pp. 103~110.

[5] Jeong, T. G. and Chen, C.(2007), "A Modification of the Approach to the Evaluation of Collision Risk Using Sech Function", Journal of Korean Navigation and Port Research, Vol.31, No.2 pp. 121~126.

[6] KMST(2007), "Table 22 Present Maritime Accidents Reason Statistic According to Different Kinds of Accidents (2002-2006)", Korean Maritime Safety Tribunal.

Received 22 Autust 2008
Revised 17 September 2008
Accepted 20 September 2008