A CBR-Based Approach for Ship Collision Avoidance

Liu, Y.; Yang, Chunsheng; Du, X.

Publisher’s version / Version de l'éditeur:

Proceedings of the Twenty First International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2008), June 18-20, 2008, Wroclaw, Poland, 2008

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

*A CBR-Based Approach for Ship Collision Avoidance **

Liu, Y., Yang, C., Du, X.
June 2008

* published in the Proceedings of The Twenty First International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2008). June 18-20, 2008. Wroclaw, Poland. NRC 49918.

A CBR-Based Approach for Ship Collision Avoidance

Yuhong Liu[1], Chunsheng Yang[2], and Xuanmin Du[3]
[1] Merchant Marine College of Shanghai Maritime University, Shanghai 200135, China
[2] Institute for Information Technology, National Research Council Canada
[3] Shanghai Marine Electronic Equipment Research Institute Shanghai 201108, China

Abstract. In this paper, we propose a novel CBR-based approach for ship collision avoidance. After the introduction of the CBR-based decision-making support, we present two abstraction principles, selecting view points and describing granularity, to create collision avoidance cases from real-time navigation data. Several issues related case creation and CBR-based decision-making support are discussed in details, including case presentation, case retrieval and case learning. Some experimental results show the usefulness and applicability of CBR-based approach for ship collision avoidance.

Keywords: case-based reasoning; ship collision avoidance; case retrieval; case learning.

1 Introduction

Ship collision avoidance plays an important role in navigation safety. Many works [1,2,3,4,5,6] tried to apply rule-based reasoning or model-based reasoning techniques to solve this challenging problem. However, it is difficult to apply these research results to practical navigation system, since the exciting techniques cannot fully simulate the human ship-handling behavior and experience, which is the most important factor in ship-handling for collision avoidance. For instance, it is captains' navigation experiences rather than theories that be available or usable for collision avoidance in many complicated encounter situations.

Case-based reasoning (CBR) is one of the reasoning paradigms and is a feasible and efficient way to the problems which are difficult to be solved in traditional methods such as model-based reasoning. A CBR-based system solves a new problem by retrieving a similar one from a case base. We believe that CBR is a solid solution for the collision avoidance problem [7, 8, 9, 10, 11, 12], because cases can well document human collision avoidance experiences. In this work, we propose a novel CBR-based approach for collision avoidance, focusing on case creation and CBR-based decision-making support for collision avoidance. In our previous work [13,14], we have developed a tool for evaluating ship ship-handling procedures [13, 14] for collision avoidance in ship-handling simulator. The tool is able to evaluate the ship-handling results by conducting a series of mathematical and logical analyses. Some good ship-handling results can be used for case creation. In this work, we integrate this tool into a CBR-based collision avoidance framework as a means for learning cases from ship-handling simulations.

In this paper, a framework of the CBR-based decision support for ship collision avoidance is first introduced in Section 2. Then, the primary principles for extracting information of cases from ship-handing data are discussed in Section 3. In Section 4, the case representation method with combination of object-oriented and frame characteristics, and a hierarchical indexing tree are discussed in detail. The nearest neighbor case retrieval method and the case learning method are presented in Section 5 and 6, respectively. The experimental results are presented in Section 7 to demonstrate the feasibility and applicability of the proposed approach. Finally, the conclusions and future work are discussed in Section 8.

2 CBR-Based Framework for Collision Avoidance

In this section, we present the proposed CBR-based framework for ship collision avoidance. The framework shown as Fig.1 consists of three main process: problem generation, case retrieval and update, and case learning. The following is the brief description of each process.

A. Problem Generation

The problem generation process creates a problem description for collision avoidance from real-time navigation data. In this work, the ship-handling operation is simulated using a simulator. The navigation data from the simulator can be treated as real-time ship-handling data. These data include static information (such as ship type, ship

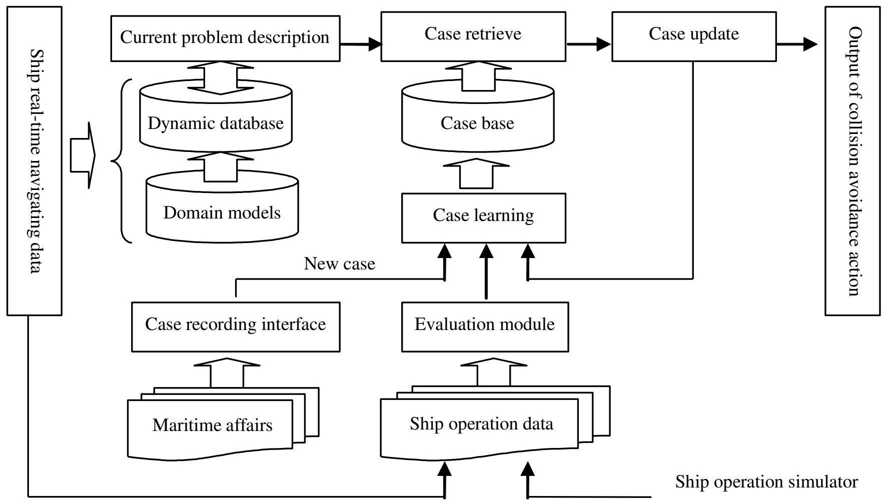


Fig. 1 CBR-based decision support framework for ship collision avoidance

length and sea gauge), dynamic information (such as course, speed and position), and navigation information (such as the relative course and speed, azimuth, distance, DCPA, TCPA, encounter situation and collision risk). Navigation information is derived from the real-time navigation data by using domain models which are stored in the domain model base in the form of procedures. All data are stored in the dynamic database in a given data format and can be used to form the current problem description as soon as the decision support model is activated.

B. Case Retrieval and Update

The cases are stored in a case base with a given presentation format and an index structure. Once a collision problem is defined from the problem generation process, a case retrieval algorithm is used to retrieve similar cases from the case base. The case with maximal similarity is selected as the proposed solution for the current collision problem. A pre-operation process is adopted to determine if the proposed solution is applicable for the current problem. If the pre-operation result is satisfying, no update is needed. Otherwise the proposed solution is updated for solving the current problem.

C. Case Learning

Case learning is an important process in the framework. The main task is to create cases from real-time ship-handling simulations. For real-time ship-handling data, we apply the developed evaluation tool to create cases by adding some remarks on the operation data, processing playback, and analyzing data.

3 Case Abstraction Principles

Ship collision avoidance is a dynamic process having a close relationship with the sea, the ship, the human, and the environment, and involving much information and changes during a period. A ship-handling procedure records ship operations over a long period. It contains all information related to ship navigation. When we create cases from a given resource such as ship-handling operation, we have to extract useful and necessary information. Therefore, we proposed two abstraction principles, selecting view points and describing granularity, in order to simplify and describe the collision avoidance procedures. Selecting view points transforms a dynamic process into a static scene. To describe this principle, we give the following definitions

Definition 1, **Encounter Scene (ES)**: a well-defined data structure. It is used to record the environment information (EI), the basic information (BI) of each ship, the relative information (RI) between own ship and each target ship and the proposed actions (PA) at a given time point. That is:

$$ES = < EI,\ BI,\ RI,\ PA > \tag{1}$$

Definition 2, **View Point (VP)**: during the ship collision avoidance, we label one of the encountering ships as the own ship (OS) and the others as target ships (TS). And then we select a time point T and record the encounter scene (ES) at this moment. VP is denoted as:

$$VP = < OS,\ TS,\ T,\ ES > \tag{2}$$

Definition 3, **Describing Granularity (DG)**: the precision of case description, which is decided by the case attribute number (AN) and attribute value type (VT). That is:

$$DG = <AN,\ VT > \tag{3}$$

Where: VT may be a precise value (PV), a fuzzy value (FV) or a default value (DV). When DV= {NULL}, the value of this attribute has nothing to do with case matching.

For a given VP, each type of information in ES is defined as an attribute in a case. The describing granularity is to determine the number and value type of the case attribute. DG impacts directly on storage space, retrieval time and applicability of a CBR-based system. In general, the bigger the DG, the smaller the storage spaced and the quicker the retrieving time, but the applicability may not be satisfied. As a result, we have to choose a trade-off DG value when we build a case base for a CBR-based application.

4 Case Representation and Organization

According to case abstraction principles above, we can represent the case by combining the characteristics of object-oriented methods and frame knowledge representation methods. Five kinds of attribute classes and their relationships are defined and shown in Fig. 2.

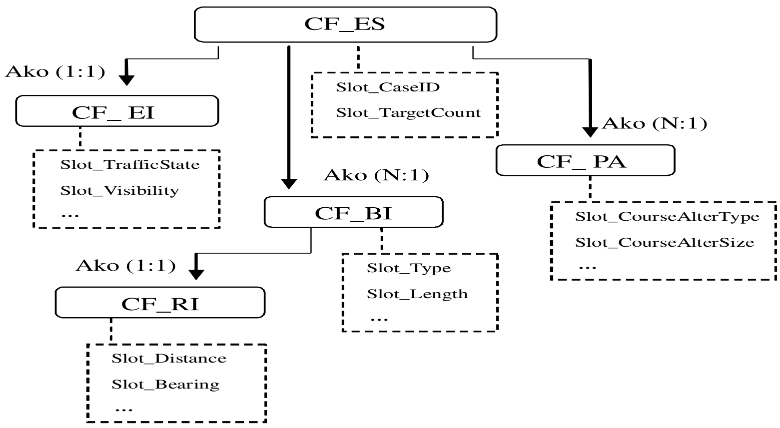


Fig. 2 Case representation: classes and attributes

CF_ES is the class of ES with four sub-classes, CF_EI, CF_BI, CF_RI and CF_PA, which corresponds to the EI, BI, RI and PA defined in equation (2). Each class has some attributes in a frame representation format. Ako (N: M) is the abbreviation for a-kind-of and is used to describe the relationship between two classes. That means M objects of the super-class will have some relationships with N objects of the sub-class.

The organization of a case base has an important impact on case retrieving efficiency. We create a hierarchical indexing tree for organizing the case base. The indexing tree is ordered with attributes in terms of their importance. These attributes are the number of encounter ships (Slot_TargetCount), visibility at sea (Slot_Visibility), two ship encounter situation (Slot_Situation) and water area condition (Slot_TrafficState). In a top-down way, the indexing tree shown in Fig. 3 is organized by decreasing importance and each node of the indexing tree corresponds to one of the attribute values. Since each case belongs to one of the nodes in the tree, such index facilitates case retrieval.

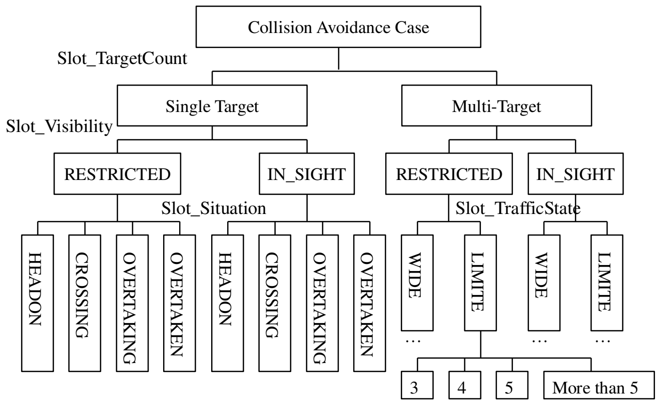


Fig. 3 The hierarchical indexing tree of a case base for collision avoidance

5 Case Retrieval

Given a description of a collision problem, we use a retrieval algorithm to retrieve the most similar cases from the case base that is indexed with a hierarchical indexing tree above. Our case retrieval algorithm is a Nearest-neighbor-based approach, which assesses the similarity between the stored cases and the problem description by a weighted sum of attributes. We describe the algorithm as follows:

If $V_{ck}^{i} \in [0,1]$ is defined as the similarity between a current problem *c* and a stored case *k* in the i^{th} attribute, x_c^i and x_k^i are the value for the ith attribute in *c* and *k* respectively. With different value types, PV, FV or DV, there are four kinds of functions to calculate attribute similarity.

1) Similarity between two precise values: for two precise values x_c^i and x_k^i, the equal or unequal judgement method is not adopted. Instead, a more flexible similarity function is introduced.

$$V_{ck}^{i} = 1.0 - \frac{\left|x_c^i - x_k^i\right|}{\lambda_i} \tag{4}$$

where: λ_i is a threshold for the i^{th} attribute.

2) Similarity between a precise value and a fuzzy value: for two values x_c^i and x_k^i, if one is a precise value *x*, and the other is a fuzzy value with a fuzzy set U, then the fuzzy membership function $\mu_U(x)$ is selected as the similarity function.

$$V_{ck}^{i} = \mu_U(x) \tag{5}$$

3) Similarity between two fuzzy values: for two fuzzy values x_c^i and x_k^i with two fuzzy set *A* and *B*, their approximate relation matrix can be calculated through equation (6) and (7).

$$V_{ck}^i = \rho \cdot \left[A \circ B + \left(1 - A \otimes B\right)\right] \tag{6}$$

where:

$$\begin{cases} A \circ B = \underset{x \in X}{\vee} \left(A(x) \wedge B(x)\right) \\ A \otimes B = \underset{x \in X}{\wedge} \left(A(x) \vee B(x)\right) \\ \rho = \frac{1}{2}\left(1 - \left|CG_A - CG_B\right|\right) \end{cases} \tag{7}$$

CG_A and CG_B are the centre of gravity of *A* and *B*.

4) Similarity between default value and non-default value: for two values x_c^i and x_k^i, if one is the default value NULL, and the other is a non-default value, then similarity function will be the default value because default value NULL can be any value in the algorithm:

$$V_{ck}^i = 1.0 \tag{8}$$

Finally, S_{ck}, the similarity between a current problem *c* and a stored case *k*, can be obtained by equation (9)

$$S_{ck} = \frac{\sum_{i=1}^{m} \omega_i \cdot V_{ck}^i}{\sum_{i=1}^{m} \omega_i} \tag{9}$$

where: $\omega_i \in [0,1]$ is the weight of the i^{th} attribute and *m* is a total of attributes in a given case.

6 Case Learning

Since ship navigation data can be collected from the ship-handling simulator, we could create or learn some cases for collision avoidance by analyzing these data. To this end, we applied the developed evaluation tool in our previous work [13][14] as a method for learning cases. By evaluating the ship-handing results, our case learning procedure shown in Fig. 4 consists mainly of three processes:

Automatic case creating: For the operation of collision avoidance with excellent results, a new case is created. To obtain the attribute values for the new case from the initial data, several steps are involved, including evaluating ship action time, action type and action size; extracting ship basic information (BI) and collision avoidance action (PA); calculating relative information (RI) between two ships; and determining the view point (VP). Using the obtained values, a new case is automatically created and added to the case base.

Manual case revising: In learning cases, not all of the attribute values can always be obtained automatically from the initial data. Some values such as the environment information (EI) and ship static data are not involved in the initial data. In such a situation, a manual case revision is necessary. Therefore, we developed a user interface to help input some information manually when necessary.

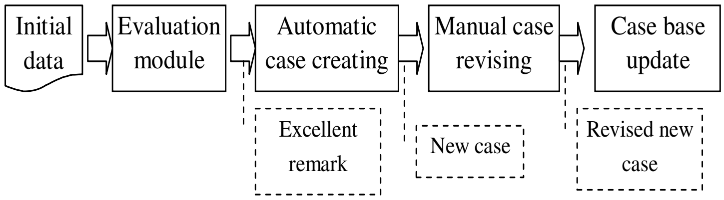


Fig. 4., The processes for learning cases from navigation data

Case base update: When adding a new case to a case base, we have to check the integrity and consistency. At the same time, the indexing tree needs to be updated automatically. The following is the description of the case base updating method. If we define $C=\{c_1,c_2,\cdots,c_M\}$ as a case set where M is the total number of cases, define $A=\{a_1,a_2,\cdots,a_N\}$ and $B=\{b_1,b_2,\cdots,b_L\}$ as a set of case condition attributes and a set of case conclusion attributes, respectively, where *N* and *L* are the number of attributes. For a case $c_i \in C \quad (i=1,2,\cdots,M)$, $x_i^{a_j}$ and $x_i^{b_k}$ are the attribute values for attribute $a_j \in A \quad (j=1,2,\cdots,N)$ and $b_k \in B \quad (k=1,2,\cdots,L)$. We also define $S_{ck}^A=\frac{1}{N}\sum_{i=1}^{N}V_{ck}^i$ and $S_{ck}^B=\frac{1}{L}\sum_{i=1}^{L}V_{ck}^i$ as the condition similarity and conclusion similarity between case *c* and *k*, where V_{ck}^i can be obtained through equation (4), (5) or (6).

Definition 4, **Identical Case**: for two cases $c_1 \in C$ and $c_2 \in C$, if there are $S_{12}^A \geq \lambda_A$ and $S_{12}^B \geq \lambda_B$, where λ_A and λ_B are two thresholds within [0,1.0] predefined based on domain knowledge, then c_1 and c_2 are regarded as identical cases.

Definition 5, **Incompatible Case**: for two cases $c_1 \in C$ and $c_2 \in C$, if $S_{12}^A \geq \lambda_A$, but $\exists b_k \in B \quad (k=1,2,\cdots,L)$ with $x_1^{b_k}$ and $x_2^{b_k}$ having incompatible values, then c_1 and c_2 are regarded as incompatible case.

Definition 6, **Inclusive Case**: for two identical cases $c_1 \in C$ and $c_2 \in C$, if $\exists a_j \in A$ with $x_1^{a_j}=NULL$ while $x_2^{a_j} \neq NULL$, then c_1 and c_2 are regarded as an inclusive case, called that c_1 includes c_2 or c_2 is included in c_1.

Definition 7, **Case Base Consistency**: if neither an identical case nor an incompatible case exists in the case base, then the case base has consistency.

Definition 8, **Case Base integrity**: if no inclusive case exists in the case base, then the case base has integrality.

Based on these definitions of the integrity and consistency, we have 4 possible operations for updating a case base: (a) if there exists an identical case against the new case, then the new case will not to be added to the case base; (b) if there exists incompatible case against the new case, then one of them will be kept in the case base; (c) if there exists a case which includes the new case, then the new case will not to be added; (d) if there exists a case which is included in the new case, then the new case will be added and the other one should be deleted

7 Experiments and Results

We implemented a prototyping system for the proposed CBR-based approach in a VC^{++} platform. We conducted some experiments for validating the applicability of the approach. For validating the approach, we focused on case learning ability and decision-making support ability by retrieving the similar cases from the case base. We present some results in this section.

Fig 5 illustrates a case learning procedure from navigation data which is collected from a two-ship encounter simulation. Fig. 5(*a*) displays the collision avoidance procedures between two ships. Each ship is treated as an intelligent agent with

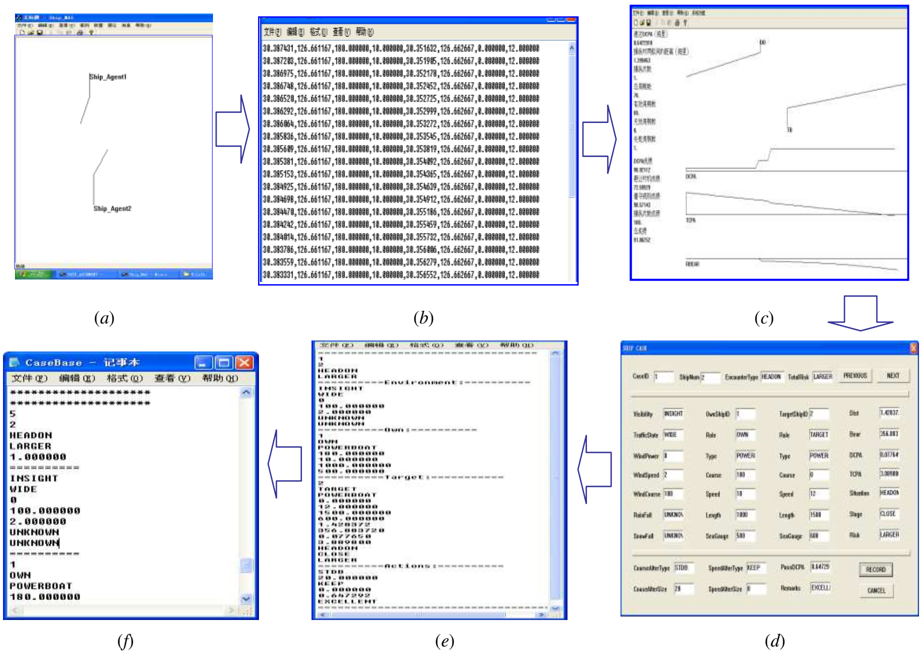
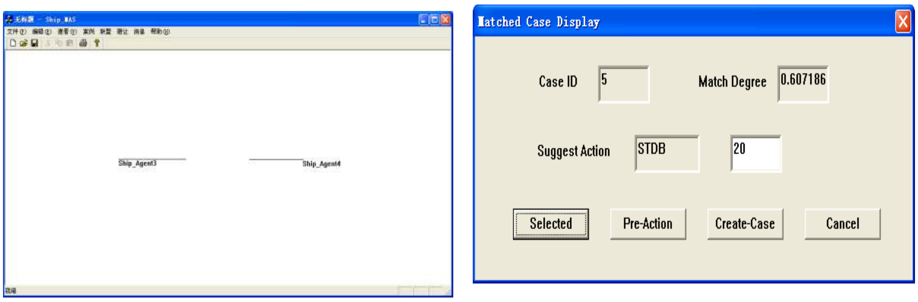


Fig. 5 An example of case learning



(*a*) Fig. 6 An example of case retrieving (*b*)

calculating, reasoning, learning and communicating abilities Ship_Agent1 has the initial course 180º and speed 10 kn. Ship_Agent2 has the initial course 0º and speed 12 kn. The two ships are in a head on situation with certain collision risk. In order to avoid the collision risk, each ship takes an action: turn right (denoted as STDB in the case conclusion attribute). Ship_Agent1changes course to 200º, and Ship_Agent2 changes course to 30°. Fig. 5(*b*) shows the basic navigation data (latitude, longitude, course and speed) of these two ships, recorded in a file every 5 minutes. Using the recorded data, the evaluation tool can analyze their trajectories and obtain the necessary information for creating cases. Fig. 5(*c*) shows these evaluating results, which include the ship navigation trajectories, the variation curves of DCPA, TCPA and azimuth. After selecting Ship_Agent1 as an own ship and Ship_Agent2 as a target ship, the evaluation tool gives the remarks on the ship navigation in a form of grade. In this case, the total grade is 91.06252. Using the obtained information, the case learning algorithm creates a case for that collision avoidance procedure. Fig. 5(d) shows a case that has 27 attributes: 5 attributes for the case conclusion and 22 attributes for the case condition. Using the interface, some attribute values can also be input, revised and recorded if necessary. Fig. 5(*e*) shows that the revised new case is stored in a temporal file. Fig. 5(*f*) demonstrates that the new case has been added into the case base and assigned to an ID number 5 in the indexing tree.

Fig. 6 is a case retrieval example. In this experiment, the system retrieved a similar case from the case base for a new collision problem. Fig. 6(*a*) is a new encounter situation, in which two ships are head on to each other. They are noted as Ship_Agent3 and Ship_Agent4. Ship_Agent3 has course 90º and speed 12 kn. Ship_Agent4 has course 270º and speed 10kn. In order to make decisions on collision avoidance, Ship_Agent3 is selected as an own ship and the current problem description is formed. Fig. 6(*b*) shows that a case is retrieved from a case base bin terms of the computed similarity. In this experiment, the case with ID 5 is retrieved since it has the maximal similarity 0.607186. Therefore, its solution (turn right 20º) is used as a proposed solution for the current problem.

8 Conclusions and Future Work

In this paper, we presented a CBR-based approach for ship collision avoidance. We discussed several important issues in the development of CBR-based collision avoidance systems, including case representation, case base indexing, case retrieval, and case learning. Using the developed prototyping system, we conducted some experiments for validating the usefulness and applicability of the proposed approach. The experiments demonstrated that the system can provide an effective way for learning cases from the real-time ship-handling data automatically. The created case can be retrieved as a solution for a similar collision problem.

Although the experiment results showed the usefulness and applicability, we still have to conduct large-scale exclusive experiments for different navigation environments and more complicated encounter simulations in order to evaluate the approach effectively. Some work is on going; we will report the results in other paper. From the viewpoint of case learning ability, we have to develop a technique for learning cases from maritime affairs records which were/are collected in ship navigation over many years. These records reflect either instructive and successful case or edifying and failing cases. They are valuable resources to generate cases for CBR-based collision avoidance systems. Therefore, learning cases from maritime affaire records will be our future work.

Acknowledgments

This research is supported by HuoYingdong educational fund grant (No.0254-05-FN580001) and Shanghai academic project (No.T0603).

References

1. Y. Sato and H. Ishii, "Study of collision-avoidance system for ships," *Control Engineering Practice*, vol. 6, pp. 1141-1149, 1998.
2. Y. Liu and W. Yang, "The structure Design of an Intelligent Decision Support System for Navigation Collision Avoidance," *IEEE The Third International Conference on Machine Learning and Cybernetics*, vol. 1, pp 302-307, August 2004.
3. Y. Liu, "A design and Study on Intelligence Collision Prevention Expert System for Navigation," *Ph.D. thesis, Harbin Engineering University*, *china*, July 1999.
4. C. Yang, "An Expert System for Collision Avoidance and Its Application," *Ph.D. thesis, Hiroshima University, Japan,* September 1995.
5. C. Hwang, "The Integrated Design of Fuzzy Collision Avoidance and H_∞- Autopilots on ships," *The Journal of navigation*, vol. 55, no.1, pp.117-136, 2002.
6. Y. Liu, X. Du and S. Yang, "The Design of a Fuzzy-Neural Network for Ship Collision Avoidance," *ICMLC 2005*, *Lecture Notes in Computer Science,* vol. 3930, pp. 804-812, July 2006.
7. I. Watson and F. Marir, "Case-Base Reasoning: A Review", Knowledge Engineering Review, vol. 9, no. 4, 1994.
8. C. Yang, B. Orchard, B. Farley and M. Zaluski, "Automated Case Base Creation and Management," *Proceedings of International Conference on Industrial & Engineering. Applications of Artificial Intelligence & Expert System (IEA/AIE-2003)*, June 2003.
9. M. Therani, J. Zhao and B. Marshall, "A case-based reasoning framework for workflow model management," *Data & Knowledge Engineering,* vol. 50, pp. 87-115, 2004.
10. P. Avesani, S. Ferrari and A. Susi, "Case-Based Ranking for Decision Support Systems," ICCBR 2003, LNAI 2689, PP. 35-49, 2003.
11. D. McSherry, "Similarity and Compromise," ICCBR 2003, LNAI 2689, pp. 291-305, 2003.
12. J. Nordlund and H. Schafer, "Case-Based Reasoning in a support System," Master's thesis, UMEA University, Sweden, April 2006.
13. Yuhong Liu, Shenping Hu, "An Evaluation System for Single-Target Ship Collision Avoidance Based on Data Fusion," NAVIGATION OF CHINA, vol. 65, no. 4, pp. 40-45, Dec. 2005.
14. Yuhong Liu, Hongxia Liu, "Case Learning Based on Evaluation System for Vessel Collision Avoidance," IEEE The Fifth International Conference on Machine Learning and Cybernetics, vol. 4, pp. 2064-2069, August 2006.