

NRC Publications Archive Archives des publications du CNRC

Designing Multiagent-based Education Systems for Navigation Training

Yang, Chunsheng; Lin, H.; Lin, O.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version.
/ La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Proceedings of The 5th IEEE International Conference on Cognitive Informatics (ICCI2006), 2006

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
<https://nrc-publications.canada.ca/eng/view/object/?id=aaf899d1-ad50-48f9-a290-f75a0229049d>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=aaf899d1-ad50-48f9-a290-f75a0229049d>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



National Research
Council Canada

Conseil national
de recherches Canada

Institute for
Information Technology

Institut de technologie
de l'information

NRC - CNRC

Designing Multiagent-based Education Systems for Navigation Training *

Yang, C., Lin, H., and Lin, O.
July 2006

* published in the Proceedings of The 5th IEEE International Conference on Cognitive Informatics (ICCI2006). Beijing, China. July 17-19, 2006.
NRC 48555.

Copyright 2006 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Designing Multiagent-Based Education Systems for Navigation Training

Chunsheng Yang¹, Hong Lin² and Fuhua Oscar Lin³

¹ Institute for Information Technology, National Research Council Canada

Chunsheng.Yang@nrc.gc.ca

² Dept of Computer & Mathematical Science, University of Houston-Downtown

ling@uhd.edu

³ School of Computing and Information Systems, Athabasca University

oscarl@athabascau.ca

Abstract

This paper describes how to design a multiagent-based education system for navigation training. The paper addresses several important issues in designing multiagent e-education systems, including the infrastructure for multiagent systems, the multiagent system specification using a chemical reaction metaphor, and the knowledge-based decision-making support for intelligent agents. The presented approach for designing multiagent-based systems is not only useful for education systems but also for other multiagent-based applications such as multi-robotic systems.

Keywords: multiagent systems, intelligent agent, decision-making support, education systems.

1. Introduction

In the past decades, there have been many achievements in the use of ship-handling simulators to train students¹ [5][6]. In our previous work [2], we proposed to apply collision avoidance expert systems to an existing ship-handling simulator as an intelligent tutor. The objective was to enhance navigation training systems and change current training routines from trainer-instruction-based procedures to intelligent-tutor-based procedures. An intelligent tutor provides decision-making support for collision avoidance by recommending an effective action for collision avoidance with detailed explanations. The students can acquire knowledge and skills from the recommended action and its explanation. Throughout the designed training courses, novices can gain basic navigation knowledge and master navigation skills without using high-cost navigation training environments, such as on-board training in real navigation environments.

This Research is Supported Through Scholars Academy of UHD from U.S. Army Research Office Award #W911NF-04-1-0024

However, the proposed intelligent tutoring system is a single-agent-based system. It can only provide training courses to a single student at a specified time and at a designated location where a ship-handling simulator is available. There are existing location and timing limitations, and the system does not allow other students to share training courses. These limitations restrict the wide usage of training systems from students. As an effective navigation training system, it should be able to provide much richer information or intuitive solutions to students and allow many of them to access training courses from different locations at any time.

Recently, with the rapid development and maturation of multiagent systems technology, it has become possible to develop the desired navigation training systems. Therefore, we propose to develop multiagent-based education systems for navigation training. In this paper, we describe how to design such a system by incorporating multiagent and knowledge-based reasoning techniques into education systems. We address several important issues, including the infrastructure for multiagent systems, the multiagent system specification using a chemical reaction metaphor (CRM), and the knowledge-based decision-making support for intelligent agents.

This paper has 5 additional sections: Section 2 presents the infrastructure for multiagent-based education systems; Section 3 specifies agent behaviors using CRM; Section 4 discusses the design for each agent in the education systems in detail; Section 5 discusses the open issues; and the final section concludes the paper.

2. Infrastructure

To allow students to access the training systems from different locations and platforms at any time, we propose to use distributed learning environments [7] as the infrastructure for multiagent-based education systems.

¹ Students in the paper could be navigators, seafarers, or trainees.

As shown in Figure 1, the navigation training systems provide a distributed training environment for students from different locations and at any times to learn how to handle ships in navigation environments. Obviously, this is

This multiagent-based distributed infrastructure provides a fundamental learning environment, which consists of the following main agents:

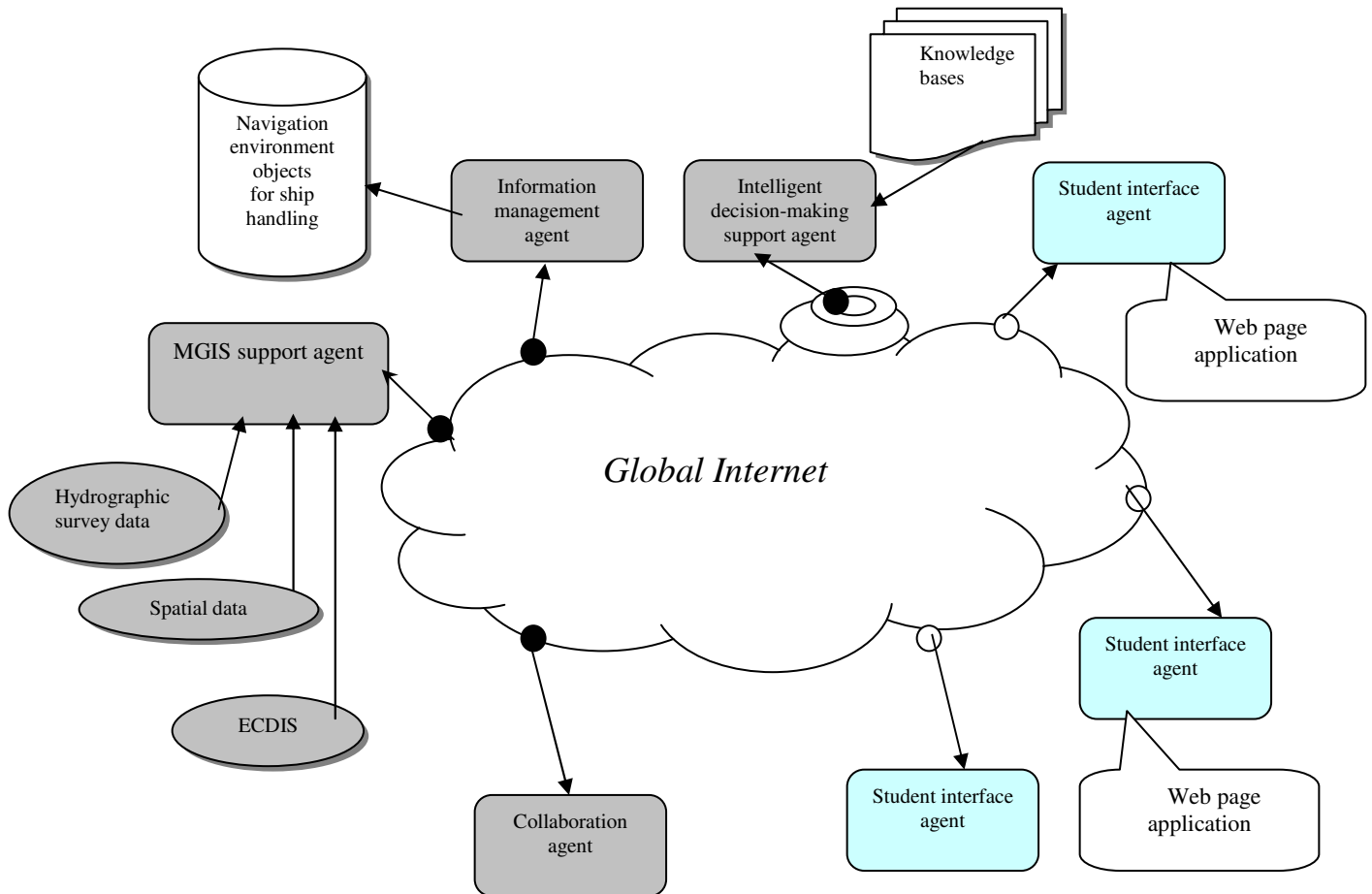


Figure 1, the infrastructure for multiagent-based education systems

a multiagent-based distributed learning environment, which provides intelligent decision-making support as well as a multitude of training course objects for students to choose and experience under various navigation environments.

A multiagent-based system for navigation training consists of the client portion and the server portion. On the client side it has an html/JSP user interface. On the server side there are Java servlets and a multiagent platform implemented using JADE (Java Agent Development Framework) [8]. JADE is a framework for developing multiagent systems according to FIPA (The Foundation for Intelligent Physical Agents) standards.

- Information management agent;
- Intelligent decision-making support agent;
- MGIS support agent;
- Collaboration agent;
- Student interface agent.

We will discuss these agents in detail in a later section. With these agents, the learning environments support the interactions among agents that may be geographically dispersed on the Internet. This distributed learning environment provides an on-line training environment that enables students to acquire knowledge and improve navigation skills at any time and from anywhere, simply by using a Java-enabled browser. The following is a scenario

showing how students use the training systems. When a student logs on an education system through web-based applications, a student interface agent looks at requirements and profile, then chooses a training course object. After the system downloads the assigned training course object, it can run the course object, which will communicate with a collaboration agent. The collaboration agent coordinates with other agents. For example, it will ask an information agent to provide navigation environment data to an intelligent decision-making support agent and an MGIS² agent. Once the MGIS agent receives the navigation environment data it provides geographical information to the student's learning object and displays an ECDIS (Electronic Chart Display Information Systems) chart on his/her local desktop. An intelligent decision-making agent provides the recommended action for safe navigation with detailed explanations. Therefore, the students could acquire basic navigation knowledge and master navigation skills through the well-designed training course objects.

3. Specifying agent behaviors

In this section, we specify the agents in the education systems using CRM [1].

The Chemical Reaction Model describes a computation as a process of iterative applications of transition rules (so called "reaction" rules) to data in multisets (representing "chemical solution") causing the multisets to be updated until no more changes are possible, at which point the resulting multisets represent the results of the program. The Gamma language was developed as a formal language to specify programs in CRM. Existing work [1] has proven that CRM is very suitable for specifying the architectural properties of multi-agent systems. The Gamma language can specify the system behaviors without involving any implementation or platform details. Therefore, it provides us a good starting point for the design of a multi-agent system.

In the following narration, we assume that the readers have basic knowledge about CRM and the Gamma language. Interested readers may refer to [9, 10] for detailed description about the Gamma language.

In the architectural specification of the navigation training system, we avoid involving specific user requests and deduction rules or knowledge used to form the response to the user requests. Instead, we focus on the

interaction among agents in the system. The following is the list of agents in the system:

MGIS: MGIS support agent

IMA: Information management agent

IDMS: Intelligent decision-making support agent

SIA: Student interface agent

CA: Collaboration agent

IMA manages the base information needed to support decision-making. The base information includes the information about the global environment. Information managed by IMA is considered static because it is a collection of knowledge and historic facts. MGIS manages specific environment knowledge and performs inferences to acquire information needed for decision-making. It may get information from IMA to support its operations. In addition, MGIS has a rendering component that displays the geographic environment in graphical mode. IDMS receives requests from the SIAs and makes recommendations for the students. IDMS interacts with MGIS to obtain necessary information for decision-making. SIA provides the user interface through which the students send requests to IDMS and obtain responses from IDMS. CA is an agent that manages the interactions among other agents. It defines communication protocols and handles the communication channels. In our system, we describe CA as a manager of a pool of messages transmitted among agents.

A request-response cycle includes the following steps: A request is generated by one of the SIAs, and the request is transmitted to IDMS. IDMS finds a set of deduction rules that will be used to form a solution and a set of necessary factors that are needed by the deduction rules. To obtain the factors, a set of requests are sent to MGIS. MGIS finds another set of rules that will be used to obtain the requested factors. It also finds a set of necessary factors that are needed by IDMS and a set of requests to obtain the factors and send them to IMA. IMA then responds with the requested factors, enabling MGIS to find the factors requested by IDMS and in turn enable IDMS to form the solution. Note that all the communications in the above process are coordinated by CA.

The Gamma specification of the system follows:

$$\begin{aligned}
 \text{NT S D E K} = & \\
 & [CA, \text{MGIS} = [P, \text{Knl} = D, \\
 & \quad \text{Req} = \emptyset, \\
 & \quad \text{Fac} = \emptyset, \\
 & \quad \text{Sol} = \emptyset \\
 &], \\
 & \text{IMA} = [Q, \text{Env} = E, \\
 & \quad \text{Req} = \emptyset, \\
 & \quad \text{Sol} = \emptyset \\
 &],
 \end{aligned}$$

² MGIS stands for Marine Geographical Information Systems. It provides the graphical information underwater for navigational environments. Such information is useful for navigators to make decision for safe navigation.

$$\begin{aligned}
 & \text{IDMS} = [\text{R}, \text{Knl} = \text{K}, \\
 & \quad \text{Reg} = \emptyset, \\
 & \quad \text{Fac} = \emptyset, \\
 & \quad \text{Sol} = \emptyset \\
 & \quad], \\
 & \text{SIA} = [\text{T}, \text{Stu} = \text{S}, \\
 & \quad \text{Req} = \emptyset, \\
 & \quad \text{Sol} = \emptyset \\
 & \quad] \\
 &] \text{ where} \\
 \text{CA} &= \text{Up1} + \text{Up2} + \text{Up3} + \\
 & \quad \text{Dn1} + \text{Dn2} + \text{Dn3} \text{ where} \\
 \text{Up1} &= \text{s:SIA}, \text{i:IDMS} \rightarrow \\
 & \quad \text{s}[\text{r:Req} = \emptyset]:\text{SIA}, \\
 & \quad \text{i}[\text{r:Req} = \text{r+s.r}]:\text{IDMS} \leftarrow \\
 & \quad \text{s.r} \neq \emptyset \\
 \text{Up2} &= \text{i:IDMS}, \text{m:MGIS} \rightarrow \\
 & \quad \text{i}[\text{r:Req} = \emptyset]:\text{IDMS}, \\
 & \quad \text{m}[\text{r:req} = \text{r} + \text{i.r}]:\text{MGIS} \leftarrow \\
 & \quad \text{i.r} \neq \emptyset \\
 \text{Up3} &= \text{m:MGIS}, \text{i:IMA} \rightarrow \\
 & \quad \text{m}[\text{r:req} = \emptyset]:\text{MGIS}, \\
 & \quad \text{i}[\text{r:req} = \text{r} + \text{m.r}]:\text{IMA} \leftarrow \\
 & \quad \text{m.r} \neq \emptyset \\
 \text{Dn1} &= \text{i:IMA}, \text{m:MGIS} \rightarrow \\
 & \quad \text{i}[\text{s:Sol} = \emptyset]:\text{IMA}, \\
 & \quad \text{m}[\text{f:Fac} = \text{f} + \text{i.s}]:\text{MGIS} \leftarrow \\
 & \quad \text{i.s} \neq \emptyset \\
 \text{Dn2} &= \text{m:MGIS}, \text{i:IDMS} \rightarrow \\
 & \quad \text{m}[\text{s:Sol} = \emptyset]:\text{MGIS}, \\
 & \quad \text{i}[\text{f:Fac} = \text{f} + \text{m.s}]:\text{IDMS} \leftarrow \\
 & \quad \text{m.s} \neq \emptyset \\
 \text{Dn3} &= \text{i:IDMS}, \text{s:SIA} \rightarrow \\
 & \quad \text{i}[\text{s:Sol} = \emptyset]:\text{IDMS}, \\
 & \quad \text{s}[\text{s:Sol} = \text{s} + \text{i.s}]:\text{SIA} \leftarrow \\
 & \quad \text{i.s} \neq \emptyset \\
 \text{P} &= \text{Pre} + \text{Post} \text{ where} \\
 \text{Pre} &= \text{r:Req}, \text{d:Dat} \rightarrow \\
 & \quad \text{f:Fac}, \text{d:Dat} \leftarrow \\
 & \quad \text{f} = \text{Pred}(\text{r}, \text{d}) \\
 \text{Post} &= \text{f:Fac}, \text{d:Dat} \rightarrow \\
 & \quad \text{s:Sol}, \text{d:Dat} \leftarrow \\
 & \quad \text{s} = \text{d}(\text{f}) \\
 \text{Q} &= \text{r:Req}, \text{e:Env} \rightarrow \\
 & \quad \text{s:Sol}, \text{e:Env} \leftarrow \\
 & \quad \text{s} = \text{e}(\text{r}) \\
 \text{R} &= \text{Pre} + \text{Post} \text{ where} \\
 \text{Pre} &= \text{r:Req}, \text{k:Knl} \rightarrow \\
 & \quad \text{f:Fac}, \text{k:Knl} \leftarrow \\
 & \quad \text{f} = \text{Pred}(\text{r}, \text{k}) \\
 \text{Post} &= \text{f:Fac}, \text{k:Knl} \rightarrow \\
 & \quad \text{s:Sol}, \text{k:Knl} \leftarrow \\
 & \quad \text{s} = \text{k}(\text{f})
 \end{aligned}$$

$$\begin{aligned}
 \text{T} &= \text{s:Stu} \rightarrow \text{s:Stu}, \text{r:Req} \leftarrow \\
 & \quad \text{Send}(\text{s}, \text{r})
 \end{aligned}$$

NT is the main configuration of the navigation training system. S, D, E, and K denotes the set of students, the database maintained by the MGIS agent, the environment information base maintained by IMA, and the knowledge and inference rule base maintained by IDMS agent. In NT configuration, each agent is represented by a configuration, viz., MGIS, IMA, IDMS, and SIA. CA is represented as a set of higher level rules that operate on other configurations. The Up1, Up2, and Up3 rules are used to transfer requests from SIA to IDMS, from IDMS to MGIS, and from MGIS to IMA, respectively. Dn1, Dn2, and Dn3 rules transfer solutions from IMA to MGIS, from MGIS to IDMS, and from IDMS to SIA, respectively. The Pre rules in MGIS and IDMS are used to form outgoing requests based on received requests; and the Post rules are used to send solutions to the requesting agent.

Other notes: If c is a configuration and $e1$ an environment variable in c , $c[e1 = e2]$ denotes c with $e1$ replaced by $e2$. If d is a deduction rule and f a fact, $s = d(f)$ means that s is the solution found out by applying d to f . If d is a deduction rule and r a request, $f = \text{Pred}(r, d)$ means that f is a factor needed to apply d to r .

4. Detail design of agents

After we specify the agent behavior using CRM, we can start to design each agent in the navigation training systems. In this section, we discuss the detailed design for five agents: the information management agent, the student interface agent, the collaboration agent, the MGIS support agent and the intelligent decision-making support agent.

4.1 Information management agent (IMA)

This agent provides all necessary information for navigation training. The main information includes navigation environments: water depth, surface congestion, weather, ship maneuverability, and ship management. The information is stored in training course objects. When a student selects a specified training object, the information management agent will provide all necessary data through the collaboration agent. At the same time it will also send the training course object to the MGIS support agent and the intelligent decision-making support agent, which uses the information for decision making support.

4.2 Student interface agent (SIA)

The interface agent provides interactive operations to students. A student logs on to an interface agent and selects a necessary training course object, then the interface agent submits the requirements to a collaboration agent, which communicates with the information management agent to get all requested data for the interface agents. Once the interface agent downloads all information, it displays instructions on ship handling based on the reasoning results of a decision-making support agent, and explains the information about decision-making and navigational environments to the student. Such explanation information also contains marine geographical information provided by a MGIS support agent.

4.3 Collaboration agent (CA)

The collaboration agent coordinates communications, balances workloads for each information management agent, and monitors agents' performance. It also manages training records for each student. In particular, the collaboration agent defines communication protocols and handles the communication channels. Therefore, we design it as a manager of a pool of the messages transmitted among the agents in education systems.

4.4 MGIS support agent (MGIS)

The MGIS support agent is a decision-making support component that provides rich navigation information based on spatial analysis. ECDIS is a real-time GIS on board vessels, specifically designed for navigation safety. MGIS for a restricted water area such as a harbor can be established by using, for example, a Swath-sounding Sonar System to transmit digitized hydrographic survey data to update the sounding data of scanned paper charts or existing ECDIS. The SDSS (Spatial Decision Support System) was developed for navigation safety decisions based on ECDIS. The role of ECDIS in MGIS for navigation safety and decision support is described as follows:

- **Maritime spatial database**

The database used in ECDIS is S57 ENC (Electronic Navigational Chart), which is adopted internationally. This is more than just a replica of paper charts. The information content and intelligence coded in the S57 ENC, especially the international support provided by the Hydrographic Society, has made it the best choice for a maritime spatial database. It is in a vector data model with chain-node topology. The object classes and attributes defined in the S57 standard cover essentially all hydrographic features, including time-varying objects such as tide and currents. The object

classes for the encoding meteorological information are designed by the International Maritime Organization (IMO) and the International Hydrographic Office (IHO).

- **Onboard GIS functionality**

GIS (Geographic Information System) is a decision-making support system providing information based on spatial analysis. ECDIS is a real-time GIS onboard the vessel, specifically designed for navigation safety. Basic functions include route planning, route checking, and route monitoring (anti-grounding, deviation alarm, danger alarm, intrusion alarm, etc.). The information coded in the ENC can be queried and analyzed by cursor pointing, by user defined areas, or ship domain both from the vessel's current position and predicted position.

- **Central display and information system-user interface to mariners**

Presentation (symbolization, color, and content) of the data is in conformance with international standards (IHO S52 IEC61174 and IMO resolution A.817). ECDIS is already considered a central display and information system in the navigation training system. Figure 2 shows an example of an ECDIS based on the IHO standard. From the ECDIS graphic display, students can see the geographical environments under water and understand the recommended action from intelligent decision-making support agent.

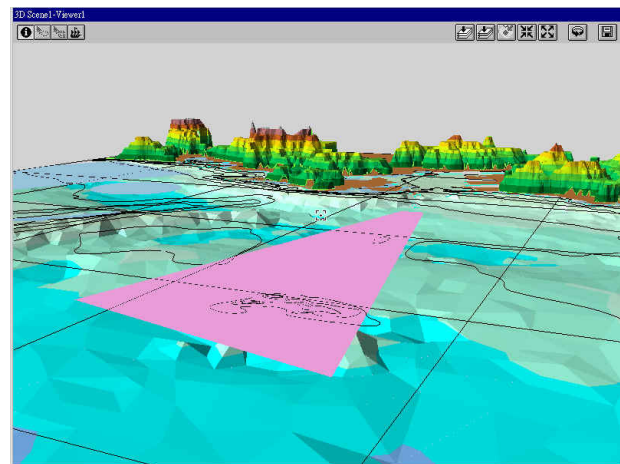


Figure 2, an example of ECDIS in MGIS agent

4.5 Intelligent decision-making support agent (IDMS)

This agent is the core of navigation training systems. It is designed as a knowledge-based reasoning system [2][3].

There are many ways to design or construct knowledge-based systems [4]. Our knowledge-based system was developed in terms of experiences obtained from many pilots, using rule-based reasoning techniques. It aims to assist students in their decision-making process to avoid collisions. It is a rule-based knowledge system designed using a hierarchical architecture and modularized knowledge structures. The top layer in the system is the inference control. The second layer is the main knowledge bases, which consist of the classification of target ships, the prediction of target ships' actions, the identification of the method of collision avoidance, and the establishment of course-line way-points. The third layer contains knowledge modules of every knowledge base. The fourth layer includes preliminary knowledge modules such as traffic regulation, and identification of target ships.

We incorporate the developed knowledge-based systems into the intelligent agent as an intelligent decision-making support agent, which makes decisions based on the selected navigation training course object and knowledge bases. The result informs the interface agent while intelligent support information shows students how to handle the ship in different ship-handling scenarios, with clear explanations. The main decision-making support information or function includes target ship classification, the encounter situation on the scheduled course line, the risk distribution around the scheduled course line, and the recommended action for collision avoidance. The following are their brief descriptions:

(1) Target ship classification

Each target ship is classified as a dangerous ship, a restricting ship, or an indifferent ship in terms of its risk. A dangerous ship is one that can potentially cause a collision if both ships maintain their current speed and course; a restricting ship is one that prevents own ship from performing any action for collision avoidance; and an indifferent ship offers no threat, whatever alteration of course own ship makes.

(2) Encounter situation on the scheduled course line

This support information indicates the encounter type between a dangerous or a restricting ship and own ship, such as overtaking, and so on. Using such supporting information, the student may easily classify the encounter relation between two ships by taking one look at the CRT.

(3) Risk distribution around the scheduled course line

This support information aims at helping the student to analyze the risk situation around a scheduled course

line. In real navigation, the ship has a destination and scheduled plan. The captain hopes to keep the ship following the scheduled plan and not deviate very far from it. In Figure 3, students can clearly see the scheduled course line and the potential collision avoidance routes. The dangerous area shows that students have to avoid that area by choosing a safe collision avoidance course line.

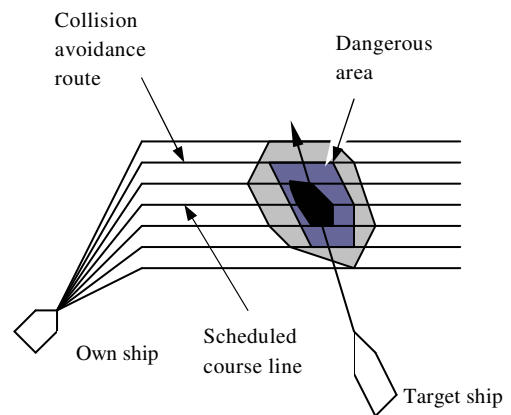


Figure 3, risk distribution around planned route

(4) Recommendation of collision avoidance action

Using the above support information is useful for students to master and analyze the risk situation around the scheduled course line. However, the final action for collision avoidance needs to be decided by the student or the instructor. We expect that training systems should be able to recommend a proper maneuvering action for collision avoidance in order to

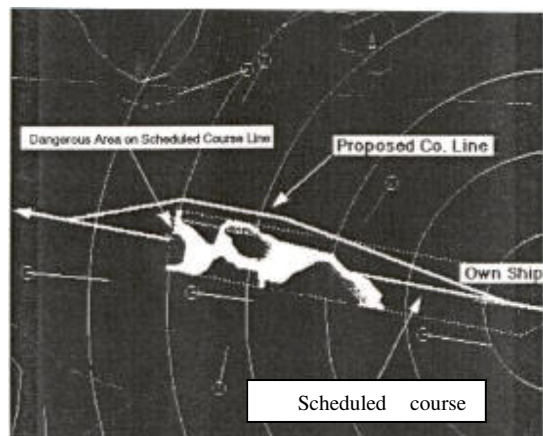


Figure 4, an example of recommended course line.

instruct students on how to determine an effective action in case of potential collision. Therefore, the designed intelligent agent is able to recommend a proper action for collision avoidance to students. Figure 4 shows an example of the recommended collision avoidance route and the risk distribution around the scheduled course line. The risk distribution information helps to explain the reason for a recommended collision avoidance action. With such support students only need to follow the recommended actions for collision avoidance. Through the well-designed training course objects, students can experience different collision situations and learn how to avoid them during real navigation in the future.

5. Discussion

Designing a multiagent-based education system involves many challenges, including agent coordination, agent negotiations, agent communications, system infrastructure, intelligence of agents, multiagent system specification and implementation. In this paper we mainly focus on the infrastructure, the multiagent system specifications, and the intelligence of agents. We also propose to use CRM to specify agent behaviors in designing the multiagent-based distributed education systems. We found that CRM is a good approach for designing the multiagent-based systems. In the Chemical Reaction Metaphor, we can express the architectural properties of a multiagent system succinctly and precisely. However, how to transform CRM-based specifications into the implementation of agents is still an open issue. We expect that it will be possible to automatically transfer CRM-based specifications into the designated implementations of agents in the future.

6. Conclusions

We discussed how to design multiagent-based education systems for navigation training using multiagent systems and knowledge-based reasoning techniques. We addressed several important issues, including CRM-based specifications of agent behaviors, system infrastructure, and the knowledge-based decision-making support for intelligent agents. These techniques and approaches are useful not only for designing multiagent-based education systems but also for other multiagent-based systems such as multi-robotics systems. Even through we have implemented some parts of the multiagent systems, we still need to fully implement the designed multiagent systems for navigation training in our future work.

Acknowledgements

The authors would like to thank Professor Pi-kuei Guo and Dr. Sheng-long Kao of the National Taiwan Ocean University for providing us ECID figures and the design requirements for MGIS support agent. The special thank goes to Bob Orchard for his proofreading.

References

1. H. Lin, and C.S. Yang, "Specifying Distributed Multi-agent Systems in Chemical Reaction Metaphor", *Journal of Applied intelligence*, Vol.24, No.2, April, 2006, pp. 155-168
2. C. Yang, et al. "Applying Collision Avoidance Expert System to Navigation Training Systems as an Intelligent Tutor", *Proceedings of IEA/AIE 14th International Conference*, Budapest, Hungary, June, 2001, 941-947
3. C. Yang, An Expert System for Collision Avoidance and Its Application, Ph.D. Thesis, the National Hiroshima University, Japan, 1995
4. M. Hogeboom, F. Lin, L. Esmahi and C. Yang, "Constructing Knowledge Bases for E-Learning Using Protege 2000 and Web Services", in *Proceedings of The International Conference on Advanced Information Networking and Applications (AINA 2005)*, Taiwan, March 28 - 30, 2005
5. K. Kose, "On an Intelligent Harbor Maneuvering Simulator and Its Application", *MARSIM & ICSM 90*, 1990
6. The business profile, *The Center for Marine Simulation*, edited by Marine Institute of Memorial University of Newfoundland, 1999
7. F. O. Lin (Ed), Designing Distributed Learning Environments with Intelligent Software Agents, *Information Science Publishing*, 2004
8. F. Bellifemine, A. Poggi, and G. Rimassa, "Developing multi-agent systems with JADE", In the *Seventh International Workshop on Agent Theories, Architectures and Languages (ATAL-2000)*, Boston, MA., 2000 (Available at:<http://jade.cselt.it>)
9. J. P. Banatre and D. Le Metayer, "The Gamma model and its discipline of programming", *Science of Computer Programming*, 15, 55-77, 1990
10. J. P. Banatre and D. Le Metayer, "Programming by multiset transformation", *CACM*, 36(1), 98-111, 1993